*Software Testing Assignment*

Module (Defect Management)

1. What is Priority?

Priority is defined as a parameter that decides the order in which a defect should be fixed. Defects having a higher priority should be fixed first.

* Defects/ bugs that leave the software unstable and unusable are given higher priority over the defects that cause a small functionality of the software to fail.
* It refers to how quickly the defect should be rectified.

Types of Priorities:

Priority in software testing can be divided into 3 categories:

* Low: The defect is irritant but a repair can be done once the more serious defects can be fixed.
* Medium: The defect should be resolved during the normal course of the development but it can wait until a new version is created.
* High: The defect must be resolved as soon as possible as it affects the system severely and cannot be used until it is fixed.

1. What is Severity?

Severity is defined as the extent to which a particular defect can create an impact on the software. Severity is a parameter to denote the implication and the impact of the defect on the functionality of the software.

* A higher effect of the bug on system functionality will lead to a higher severity level.
* A QA engineer determines the severity level of a bug.

Types of Severity:

Severity in software testing can be classified into 4 categories:

* Critical: This severity level implies that the process has been completely shut off and no further action can be taken.
* Major: This is a significant flaw that causes the system to fail. However, certain parts of the system remain functional.
* Medium: This flaw results in unfavourable behaviour but the system remains functioning.
* Low: This type of flaw won’t cause any major breakdown in the system.

1. Bug Categories

Bug: A malfunction in the software/system is an error that may cause components or the system to fail to perform its required functions. In other words, if an error is encountered during the test it can cause malfunction. For example, incorrect data description, statements, input data, design, etc.

* Functional Bugs

Functional bugs are associated with the functionality of a specific software component.

* Logical Bugs

A logical bug disrupts the intended workflow of software and causes it to behave incorrectly. These bugs can result in unexpected software behaviour and even sudden crashes. Logical bugs primarily take place due to poorly written code or misinterpretation of business logic.

* + Assigning a value to the wrong variable
  + Dividing two numbers instead of adding them together resulting in unexpected output
* Workflow Bugs

Workflow bugs are associated with the user journey (navigation) of a software application. Let’s consider an example of a website where a user needs to fill up a form regarding their medical history. After filling the form, the user has three options to choose from:

* + - * + Save
        + Save and Exit
        + Previous Page
* Unit Level Bugs

Unit level bugs are very common, and they are typically easier to fix. Once the initial modules of software components are developed, developers perform unit testing to ensure that the small batches of code are functioning as expected.

* System-Level Integration Bugs

System-level integration bugs primarily pop up when two or more units of code written by different developers fail to interact with each other. These bugs primarily occur due to inconsistencies or incompatibility between two or more components. Such bugs are difficult to track and fix as developers need to examine a larger chunk of code. They are also time-consuming to replicate.

* Out of Bound Bugs

Out of Bound Bugs show up when the system user interacts with the UI in an unintended manner. These bugs occur when an end-user enters a value or a parameter outside the limits of unintended use

* Security Bugs

Security is a major concern for software development. Security Bugs are a major risk for users and should be taken very seriously and resolved. Due to their high severity and vulnerable nature, security bugs are considered among the most sensitive bugs of all types and should handle with criticality and urgency. These bugs might not hinder the operation but can compromise the whole system. These should be checked thoroughly at regular intervals.

1. Advantage of Bugzila

Bugzilla is an open-source bug-tracking system. It was developed by Mozilla Foundation in 1998. It was written in Perl programming language. Basically, it is used as a bug-reporting tool in the market. Companies that use Bugzilla are Frido, My Stack, Adroiti Technologies, CloudByte and many more.

Advantages of Bugzilla

* Deadlines: To fix the bugs, deadlines can be established.
* Types: It reports in a variety of formats and types.
* Request System: You can use the ‘request system’ provided by Bugzilla to ask other users to evaluate codes, provide information and other things.
* Flexible: Bugzilla is quite flexible, so you can modify it to fit your unique process and requirements.
* Bug tracking tool: Bugzilla is extremely good at monitoring and handling bugs and issues.

1. Difference between Priority and Severity



