

**Assignment-1 Summer 2021**

**M. Ahsan Danyal 15258**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | SQA | | **Program** | BS |
| **Faculty** | Naseer Ahmed | | **Maximum Marks** | 05 |
| **Last Date & Time Submission** | 03rd September 2021 | 23:59 |  |  |

# Q.No.1 (Max Marks: 02)

How do you decide when you have tested enough?

1.More than 99% of test coverage and 100% functional coverage is achieved.

2. All test cases are passed.

3. All test document and deliverables are prepared, reviewed and published across.

4. Based on the Testing deadline and Release deadlines

5. When the bug rate falls to down

6.When the alpha or beta testing period ends.

7. All test coverage of code, functionality, module, requirement reaches the specified level.

8.We define Pass Criteria in Test Plan and if Pass Criteria matches with our schedule or current scenario we are in a situation to say that we have tested enough.

9.we match all these with our Pass Criteria then discuss with our concerned manager and can take decisions that we have tested enough.

10. when the company's Exit Criteria have been fulfilled.

# Q.No.2 (Max Marks: 03)

**Tell me some key points to consider while writing a bug report.**

**1. Bug ID**

The bug id should provide a quick description of the bug. For example, homepage.

Assigning an ID to the bug also helps to make identification easier

**2.Module:**

Each Test Module is given a descriptive name to make it easy to identify what the tests in the module cover.

For example: Login, sign up, homepage etc.

**3.Requirement:**

Hence, when a failure is observed, we can see what requirements have been impacted.

This can help in reducing duplicate defect reports in that if we can identify the source requirement, then if another defect is logged with the same requirement number, we may not need to report it again, if the defects are of similar nature.

**4.Testcase Name:**

when we find a bug, we send a bug report, not the test case to the concerned developer.it is used as a reference for the test engineer.

For example: facebook, gmail, youtube, ms word.

**5.Reporter:**

Most bug tracking systems have this information by default, so you could quickly contact the tester who found and described the bug.

**6.Release-Version:**

It provides the release number in which the bug occurs, and also the build version of the application.

**7.Status:**

* The status that shows the current stage of working with the bug.
* Invalid, Duplicate, Deferred, assigned.

**8.Date:**

The date that the defect occurred or was reported is also essential. This is normally useful when you want to search for defects that were identified for a particular release of software or from when the testing phase started.

**9.Assign to:**

This section specifies the information about the developer who will fix the bug.

**10.Severity:**

The bug affects the system, and in turn, how quickly it needs to be fixed.

• Low: Bug won’t result in any noticeable breakdown of the system

• Minor: Results in some unexpected or undesired behavior, but not enough to disrupt system function

• Major: Bug capable of collapsing large parts of the system

• Critical: Bug capable of triggering complete system shutdown

**11.Priority:**

* Low: Bug can be fixed at a later date. Other, more serious bugs take priority
* Medium: Bug can be fixed in the normal course of development and testing.
* High: Bug must be resolved at the earliest as it affects the system adversely and renders it unusable until it is resolved.

**12.Platform:**

A bug can appear in a particular platform and not others. For example, a bug appears when running the website on Firefox, and the other browser bug not appears.

**13.TestData:**

Test Data is data that is used to execute the tests on testware. Test data needs to be precise and exhaustive to uncover the defects.

For example:

Username: Ahsandanyal97@gmail.com

Password: Ajh23$#67

**14.Expected Result:**

The developer needs to know what the requirement is, in order to gauge the extent to which the bug is disrupting the user experience.

Describe the ideal end-user scenario, and try to offer as much detail as possible. Don’t just leave it at the app is crashing, and it shouldn’t.

**15.Actual Result:**

This section will be most helpful to developers. Emphasize distinctly what is going wrong.

Provide additional details so that they can start investigating the issue with all variables in mind. For example:

• Link does not lead to the expected page. It shows a 404 error.

• When clicked, the button does not do anything at all.

• The main image on the homepage is distorted on the iphone.