Vidit Gupta
Group - 3
Vidit_assignment_March_6

1. **What is bug In Software Testing?**

   In Software testing, when the expected and actual behavior is not matching, an incident needs to be raised. An incident may be a Bug. It is a programmer's fault where a programmer intended to implement a certain behavior, but the code fails to correctly conform to this behavior because of incorrect implementation in coding. It is also known as Defect.

2. **Differentiate Error, Defect, And Failure?**

Mistake in coding is Error.

Error found by tester is Defect.

Defect accepted by Developers/Developer Team is Bug.

Product does not meet the stakeholders requirement is a Failure.

3. **What Are The Different Types Of Status Of Defects?**

- **New:** When a new defect is logged and posted for the first time. It is assigned a status as NEW.
- **Assigned:** Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team
- **Open**: The developer starts analyzing and works on the defect fix
- **Fixed**: When a developer makes a necessary code change and verifies the change, he or she can make bug status as "Fixed."
- **Pending retest**: Once the defect is fixed the developer gives a particular code for retesting the code to the tester. Since the software testing remains pending from the testers end, the status assigned is "pending retest."
- **Retest**: Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to "Re-test."
- **Verified**: The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is "verified."
- **Reopen**: If the bug persists even after the developer has fixed the bug, the tester changes the status to "reopened". Once again the bug goes through the life cycle.
- **Closed**: If the bug is no longer exists then tester assigns the status "Closed."
- **Duplicate**: If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to "duplicate."

- **Rejected**: If the developer feels the defect is not a genuine defect then it changes the defect to "rejected."
- **Deferred**: If the present bug is not of a prime priority and if it is expected to get fixed in the next release, then status "Deferred" is assigned to such bugs
- **Not a bug**:If it does not affect the functionality of the application then the status assigned to a bug is "Not a bug".

4. Explain About Defect/Bug Life Cycle?

New: When a defect is logged and posted for the first time. It's state is given as new.

Assigned: When the tester assigns the bug to developer its status changes to "assigned"

Open: At this state the developer has started analyzing and working on the defect fix.

Fixed: When developer makes necessary code changes and verifies the changes then he/she can make bug status as 'Fixed' and the bug is passed to testing team.

Pending retest: After fixing the defect the developer has given that particular code for retesting to the tester. Here the testing is pending on the testers end. Hence its status is pending retest.

Retest: At this stage the tester do the retesting of the changed code which developer has given to him to check whether the defect got fixed or not.

Verified: The tester tests the bug again after it got fixed by the developer. If the bug is not present in the software, he approves that the bug is fixed and changes the status to "verified".

Reopen: If the bug still exists even after the bug is fixed by the developer, the tester changes the status to "reopened". The bug goes through the life cycle once again.

Closed: Once the bug is fixed, it is tested by the tester. If the tester feels that the bug no longer exists in the software, he changes the status of the bug to "closed". This state means that the bug is fixed, tested and approved.

Duplicate: If the bug is repeated twice or the two bugs mention the same concept of the bug, then one bug status is changed to "duplicate".

Rejected: If the developer feels that the bug is not genuine, he rejects the bug. Then the state of the bug is changed to "rejected".

Deferred: The bug, changed to deferred state means the bug is expected to be fixed in next releases. The reasons for changing the bug to this state have many factors. Some of them are priority of the bug may be low, lack of time for the release or the bug may not have major effect on the software.

Not a bug: The state given as “Not a bug” if there is no change in the functionality of the application. For an example: If customer asks for some change in the look and feel of the application like change of colour of some text then it is not a bug but just some change in the look of the application.

5. **A bug is identified by the tester it is assigned to whom?**

It is assigned to developer concerned with the bug.

6. **Why is JIRA used? Explain step by step how an issue is created in JIRA.**

JIRA is a project management tool used for issues and bugs tracking system. It is widely used as an issue-tracking tool for all types of testing.

Steps :

Click **Create** at the top of the screen to open the **Create Issue** dialog box.

Select the relevant **Project** and **Issue Type** in the **Create Issue** dialog box.

Type a **Summary** for the issue and complete any appropriate fields — at least the required ones that are marked by an asterisk.

If you want to access fields that are not shown in this dialog box, or you want to hide existing fields:

Click the **Configure Fields** button at the top right of the screen.

Click **Custom** and select the fields you want to show or hide by selecting or clearing the relevant check boxes respectively, or click **All** to show all fields.When you next create an issue, these selected fields will be displayed.

Optional: To create a series of similar issues – with the same **Project** and **Issue Type** – select the **Create another** checkbox at the bottom of the dialog. Depending on your configuration and the values you may have specified when creating previous issues, some of the fields in the new Create Issue dialog box may be pre-populated. Make sure you check they're all correct before creating the next issue. When you are satisfied with the content of your issue, click the **Create** button.

7. **What is Defect Density?**

   Defect Density is the number of defects confirmed in software/module during a specific period of operation or development divided by the size of the software/module.(Defect Density = Defect count/ size of the release

8. **What is the difference between defect density and defect triage?**

   Defect Density is the number of defects confirmed in software/module during a specific period of operation or development divided by the size of the software/module.(Defect Density = Defect count/ size of the release

   Defect triage is a process where each bug is prioritized based on its severity, frequency, risk, etc

9. **Explain Bug reporting and parameters of bug?**

A **bugs report** or defects **report** is a list of **bugs** found out by **testers** while **testing** a **software** product in **testing** phase under a **testing** environment. The **test** environment is created at development site similar to the actual environment in which the **software** is supposed to work or run in live situation at customer site.

Parameters :

Bug Id

Priority – business or development team can decide

Severity – Testing team can decide

Created by – Tester Name

Created Date – Date of created defect

Assigned to – Developer Name

Resolved Date – This Date decided by developer

Resolved By – Developer Name

Status – New, IT Committed, Development, Ready for QA, In-Testing, Testing Completed.

Project name – Current module or project Name

Product name – Main Product Name

Release Version (e.g. 1.2.3)

Module – Module Name

Detected Build Version – 1.1.1, 1.1.2

10. **What is defect management? Explain the defect management process.**

Generally, defect management can be defined as a process of detecting bugs and fixing them. It is necessary to say that bugs occur constantly in the process of software development. Hence, every software development project requires a process that helps detect defects and fix them.

The process of defect management usually includes four steps.

- **The first step is the stage of defect detecting.** We have already mentioned that it can be conducted either by the team of developers or by the users. Regardless of the type of testing, its main goal is to detect all bugs in the final product or its part.
- **The second step of the bug management process is dedicated to the formulation of bug reports.** These are the documents that include all necessary information about certain bugs. Usually, they contain data on the type of bug, and the possible ways of its correction.
- **The third step is the stage of bug fixing.** After the bugs are fixed, they should be tested once more to make sure that the software works properly.
- **During the final step the bug list is created.** This is the document that contains information about all bugs that occurred during the project's performance. The team often uses the bug list because similar bugs' occurrence is not rare.

11. **What is Test estimation? Explain Work Breakdown Structure test estimation technique with an example?**

Test Estimation is a management activity which approximates how long a Task would take to complete. In Work Breakdown Structure test estimation technique, a complex project is divided into modules. The modules are divided into sub-modules. Each sub-module is further divided into functionality. It means divide the whole project task into the smallest tasks.

12. What is test reports? What parameters are used in test reports?

Test summary report /Quality Reports is a document which contains summary of test activities and final test results.Test Report is a document which contains A summary of test activities and final test results. An assessment of how well the Testing is performed. Based on the test report, the stakeholders can Evaluate the quality of the tested product. For example, if the test report informs that there are many defects remaining in the product, the stakeholder can delay the release until all the defects are fixed.

13. What are the test management tools?

TestLink: Test-link is most widely used web based open source test management tool.It synchronizes both requirements specification and test specification together. User can create test project and document test cases using this tool. With Test-Link you can create an account for multiple users and assign different user roles.

JIRA: Jira is a proprietary issue tracking product developed by Atlassian which allows bug tracking and agile project management. It **is** used for bug tracking, issue tracking, and project management.

14. What is a test link? How do you write test cases in TestLink?

Test-link is most widely used web based open source test management tool. It synchronizes both requirements specification and test specification together. User can create test project and document test cases using this tool. With Test-Link you can create an account for multiple users and assign different user roles.

To write test cases in TestLink

Creating a Test Project

Click on the tab "create" to create a new project.

Enter all the required fields in the window like a category for a test project, name of the project, prefix, description, etc. After filling all necessary details, click on tab "Create" at the end of the window.

Creating a Test Plan From the home-page, click on Test Plan Management from home-page It will open another page, at the bottom of the page click on a tab "Create".

Fill out all the necessary information like name, description, create from existing test plan, etc. in the open window, and click on "create tab"

Build Creation

Click on Builds/Releases under Test Plan from the home page

Creating Testsuite

Click on test specification option from the home page.

On the right-hand side of the panel, click on the setting**.** It will display a series of test operation.

Click on the "create" tab for the test suite

Fill-up all the details for test-suite and click on save it tab.

Enter the test suite name

Enter the details about your test suite

Click on save button to save the details of test-suite

Creating a Test Case

Click on the test suite folder on the left side of the panel under a folder tree structure.

Click on the setting icon in the right side panel. List of test case operations will be displayed on the right side panel.

New window will open, to create test cases click on create a button in test-case operations.

15. Explain steps how to upload Test case sheet on TestLink?

Step 1 − To import test cases, go to Test Specifications → Test Specification from the dashboard.

Step 2 − Select the nearest test suite folders, where the test cases should be imported.

Step 3 − Click the Actions icon on the right pane. It displays Test Case Operations.

Step 4 − Click the Import icon

Step 5 - Select the file and upload it.

Step 6 - Click the Upload file button.

16. **What is severity and priority in bug/defect?**

Severity is defined as the degree of impact a Defect has on the development or operation of a component application being tested.

Priority is defined as the order in which a defect should be fixed.

17. **While placing an order for clothing website, in order confirmation page there is a logo error. It is a?**
    1. **High priority, high severity**
    2. **Low severity low priority**
    3. **Low severity, high priority (of low severity as it not going to affect the functionality of the website but can be of high priority as you don't want any further shipment to proceed with the wrong logo.)**
    4. **High severity low priority**

3. Low severity, high priority

18. **Website home page failed to load.**
    1. **High priority, high severity (Major functionality failure like log in is not working, crashes in the basic workflow of the software are the best example of High Priority and High Severity)**
    2. **Low severity low priority**
    3. **Low severity, high priority**
    4. **High severity low priority**

1. High priority, high severity

19. **The application works perfectly for 50k sessions but beings to crash after a higher number of sessions.**
    1. **Low severity low priority**
    2. **High priority, high severity**
    3. **Low severity, high priority**
    4. **High severity low priority (This problem needs to be fixed but not immediately.)**

4. High severity low priority

20. **An application (web) is made up of 20 pages. On one of the pages, there is a sentence with a grammatical error.**
    1. **Low severity low priority - This bug may go unnoticed to the eyes of many and won't affect any functionality or the credibility of the company.**
    2. **High priority, high severity**
    3. **Low severity, high priority**
    4. **High severity low priority**

1. Low severity low priority