

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

**Ans.** A software bug is an error, flaw, failure or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways. A bug is the consequence/outcome of a coding fault

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

**Ans.** 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 defect is logged and posted for the first time. It's state is given as new.
- **Assigned:** After the tester has posted the bug, the lead of the tester approves that the bug is genuine and he assigns the bug to corresponding developer and the developer team. It's state given as 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.

#### 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:** After the tester has posted the bug, the lead of the tester approves that the bug is genuine and he assigns the bug to corresponding developer and the developer team. It's state given as 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?**

**Ans.** It is assigned to Developer

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

**Ans.** JIRA is a tool developed by Australian Company Atlassian. It is used for bug tracking, issue tracking, and project management. The basic use of this tool is to track issue and bugs related to your software and Mobile apps. It is also used for project management.

**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?**

**Ans. 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). It enables one to decide if a piece of software is ready to be released.

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

**Ans. 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?**

**Ans.** A good bug report contains the information needed to reproduce and fix problems. It is an efficient form of communication for both bug reporter and bug receiver

Parameters of bug:

- **Defect Id**- Id provided by testing team for the bug reported.
- **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 Successfully 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.**

**Ans. 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.

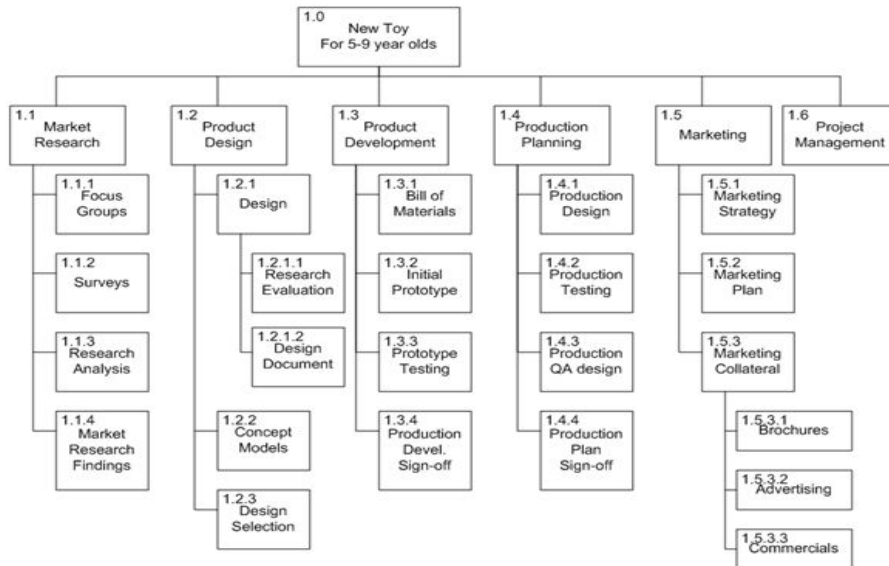
The **second step** of the bug management process is dedicated to the formulation of bug reports. These are the documents that include all necessary information about certain bugs. Usually, they contain data on the type of bug, and the possible ways of its correction.

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?**

**Ans. Test Estimation** is a management activity which approximates **how long** a Task would take to complete. Estimating effort for the test is one of the **major** and **important** tasks in Test Management. **Work Breakdown Structure (WBS)**, in Project Management and Systems Engineering, is a deliverable-oriented decomposition of a project into smaller components. WBS is a key project deliverable that organizes the team's work into manageable sections. WBS element may be a product, data, service, or any combination thereof. WBS also provides the necessary framework for detailed cost estimation and control along with providing guidance for schedule development and control.



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

**Ans. Test 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  
Make a **decision** on the software release.

## 13. What are the test management tools?

**Ans.** Test management tools are used to store information on how testing is to be done, plan testing activities and report the status of quality assurance activities. The tools have different approaches to testing and thus have different sets of features. Generally they are used to maintain and plan manual testing, run or gather execution data from automated tests, manage multiple environments and to enter information about found defects. Test management tools offer the prospect of streamlining the testing process and allow quick access to data analysis, collaborative tools and easy communication across multiple project teams. Many test management tools incorporate requirements management capabilities to streamline test case design from the requirements. Tracking of defects and project tasks are done within one application to further simplify the testing

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

**Ans.** 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?**

**Ans.** 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?**

**Ans.** 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

**Ans.** 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

**Ans.** 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.)**

**Ans. 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**

**Ans. 1. Low severity low priority**

**21. Find bugs and report the same on JIRA for below-mentioned modules in the website: <http://www.rushplace.com/>: Testwebsite1**

- 1. My Account**
- 2. Add to basket**
- 3. Search**
- 4. Homepage**

**22 . Find bugs and report the same on JIRA for below-mentioned modules in the website: <http://newtours.demoaut.com/mercurywelcome.php> : Testwebsite2**

- 1. Register Here**
- 2. Top header navigation options**
- 3. UI bugs for the complete website.**

**23. Write Test Cases for Amazon login, Sign up and Forgot password on TestLink.**

**24. Write Test Cases for placing an order in Myntra on TestLink.**

**25. Write Test Cases for Search functionality on TestLink.**