

Testing Concepts 2 Assignment
Quality Engineering Competency
Email Id: vasudev.mudgal@tothenew.com

Exercise

1. What is bug In Software Testing?

SOL : In software testing, An error, flaw, failure or fault in a computer program or system that causes it to produce an incorrect or unexpected result. A bug is the consequence/outcome of a coding fault i.e wrong code, missing code or extra code.

2. Differentiate Error, Defect, And Failure?

SOL:

Error	Defect	Failure
The incorrect human action that produces incorrect result is called an error or mistake.	Deviation between the expected behavior to actual behavior identified while testing is called defect.	Deviation from expected behavior to actual behavior by end user in operations is failure.
Can be raised by Developer, Automation Test Engineer	Test Engineer	Lots of defect leads to failure of the software.



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

SOL : During the life cycle of a defect, it can have the following statuses:

(i). OPEN / NEW: Defect is identified via a review or a test and reported. This is the beginning of a defect's journey.

(ii).ASSIGNED: The defect is assigned to a person who is tasked with its analysis or resolution.

(iii).DUPLICATE: If the defect is invalid because it's a duplicate of another one already reported, it is marked as a duplicate.

(iv).DROPPED / REJECTED: If the defect is invalid because of various other reasons like false positives, it is dropped / rejected.

(v).DEFERRED: If the defect is valid but it's decided to be fixed in a future release instead of the current release, it is deferred. When the time comes, the defect is assigned again.

(vi).FIXED / RESOLVED / COMPLETED: After the defect is 'fixed', its status is changed so that its verification can begin. If the defect can't be fixed, it could be because of any of the following reasons:

(vii).Need More Information: More information, such as the exact steps to reproduce, is required to analyze and fix the defect.

(viii).Can't Reproduce: The defect cannot be reproduced for reasons such as change of environment or the defect somehow being already fixed.

(ix).Can't Fix: The defect cannot be fixed due to some other reason. Such defect is either assigned to another person (in the hope that the other person will be able to fix it), deferred (in the hope that the delay in fixing won't hurt much), or dropped (when there's no hope for its resolution ever and the defect needs to be considered as a 'known issue')

(x).REASSIGNED: If the 'fixed' defect is, in fact, verified as not being resolved at all or being only partially resolved, it is reassigned. If there's time to fix the reassigned defect in the current release, it is fixed again but if it's decided to wait and fix in a future release, it is deferred.

(xi).CLOSED / VERIFIED: If the defect is verified as being resolved indeed, it is closed. This is the ending.

4. Explain About Defect/Bug Life Cycle?

- (i) New: This is the first state of a defect in the Defect Life Cycle. When any new defect is found, it falls in a 'New' state, and validations & testing are performed on this defect in the later stages of the Defect Life Cycle.
- (ii) Assigned: In this stage, a newly created defect is assigned to the development team to work on the defect. This is assigned by the project lead or the manager of the testing team to a developer.
- (iii) Open: Here, the developer starts the process of analyzing the defect and works on fixing it, if required.

If the developer feels that the defect is not appropriate then it may get transferred to any of the below four states namely Duplicate, Deferred, Rejected, or Not a Bug-based upon a specific reason. We will discuss these four states in a while.

(iv) Fixed: When the developer finishes the task of fixing a defect by making the required changes then he can mark the status of the defect as “Fixed”.

(v) Pending Retest: After fixing the defect, the developer assigns the defect to the tester to retest the defect at their end, and until the tester works on retesting the defect, the state of the defect remains in “Pending Retest”.

(vi) Retest: At this point, the tester starts the task of retesting the defect to verify if the defect is fixed accurately by the developer as per the requirements or not.

(vii) Reopen: If any issue persists in the defect, then it will be assigned to the developer again for testing and the status of the defect gets changed to ‘Reopen’.

(viii) Verified: If the tester does not find any issue in the defect after being assigned to the developer for retesting and he feels that if the defect has been fixed accurately then the status of the defect gets assigned to ‘Verified’.

(ix) Closed: When the defect does not exist any longer, then the tester changes the status of the defect to “Closed”.

(x) Rejected: If the defect is not considered a genuine defect by the developer then it is marked as “Rejected” by the developer.

(xi) Duplicate: If the developer finds the defect as same as any other defect or if the concept of the defect matches any other defect then the status of the defect is changed to ‘Duplicate’ by the developer.

(xii) Deferred: If the developer feels that the defect is not of very important priority and it can get fixed in the next releases or so in such a case, he can change the status of the defect as ‘Deferred’.

(xiii) Not a Bug: If the defect does not have an impact on the functionality of the application, then the status of the defect gets changed to “Not a Bug”.

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

SOL: Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team.

The developer starts analyzing and works on the defect fix. When a developer makes a necessary code change and verifies the change, he or she can make bug status as “Fixed.”

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

SOL: Jira software can be used for the following purposes:

(i). Requirements and Test case management

(ii). In Agile Methodology

(iii). Project Management

(iv). Software Development

(v). Product Management

(vi). Task Management

(vii). Bug Tracking

:- To complete the process of creating an issue, we should follow the pointers given below.

(i). Select the Project where the issue is.

- (ii).Select the type of issue, whether it is a bug/new feature/story, etc.
- (iii).Write a one-line summary to provide the overall idea about the issue.
- (iv).Write the details of the issue in the Description field. Explain the issue, so that stockholders can understand every detail of the issue.
- (v).To create a similar type of issue in the same project and issue type, check the checkbox of “Create another” otherwise keep it as unchecked.
- (vi).After entering all the details, click on the Create button.

7. What is Defect Density?

SOL: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?

SOL: .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. It enables one to decide if a piece of software is ready to be released.

Defect density is counted per thousand lines of code also known as KLOC.

Defect Triage, also known as Bug Triage, is basically used in software Testing.Defect triage is basically process that simply tries to do re-balancing of the process where the test team usually faces problem or issue of less availability of resources that are required. In defect

triage, defect is generally prioritized simply on the basis of its severity, reoccurrence, risk, etc.

9. Explain Bug reporting and parameters of bug?

SOL: A Bug Report in Software Testing is a detailed document about bugs found in the software application. Bug report contains each detail about bugs like description, date when bug was found, name of tester who found it, name of developer who fixed it, etc. Bug report helps to identify similar bugs in future so it can be avoided. If bugs occur (which they certainly do), the person finding the bug should be able to report (document & send) the bug to people in charge of fixing that error or failure. According to Yegor, a bug report “ should explain how exactly the product is broken.

Parameters of Bug are :-

- (i). Defect Id- Id provided by testing team for the bug reported.
- (ii). Priority – business or development team can decide
- (iii). Severity – Testing team can decide
- (iv). Created by – Tester Name
- (v). Created Date – Date of created defect
- (vi). Assigned to – Developer Name
- (vii). Resolved Date – This Date decided by developer
- (viii). Resolved By – Developer Name
- (ix). Status – New, IT Committed, Development, Ready for QA, In-Testing, Testing Successfully Completed.
- (x). Project name – Current module or project Name
- (xi). Product name – Main Product Name

(xii).Detected Build Version – 1.1.1, 1.1.2

(xiii).Release Version (e.g. 1.2.3)

(xiv).Module – Module Name

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

SOL: Defect Management is a systematic process to identify and fix bugs. A defect management cycle contains the following stages:

- (i) Discovery of Defect: In the discovery phase, the project teams have to discover as many defects as possible, before the end customer can discover it. A defect is said to be discovered and change to status accepted when it is acknowledged and accepted by the developers
- (ii) Defect Categorization: Defect categorization help the software developers to prioritize their tasks. That means that this kind of priority helps the developers in fixing those defects first that are highly crucial.i.e Critical, High, Medium, Low.
- (iii) Resolution: Once the defects are accepted and categorized they are fixed by the developer.
- (iv) Verification: After the development team fixed the reported issues, the testing team verifies them.
- (v) Closure: Once a defect has been resolved and verified, the defect is closed.
- (vi) Reporting: Current defect situation is reported.

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

SOL: 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. The estimation of resources, time, human skill and cost on a project.

Work Breakdown Structure test estimation technique: In this 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.

Step 1- List all modules of Gym Management App like Customers, Fees, Diet Plans, Exercises.

Step 2- Divide all modules into sub modules like diet plan for vegetarian, for non vegetarian, keto etc for all modules.

Step 3- Divide the sub modules further into functionalities like in this case we can divide further Vegetarian diet for each day of week.

Step 4- Divide functionalities into sub-functionalities not required in this case.

Step 5- Review all the testing requirements to make sure they are added in WBS so that no module is left out.

Step 6- In total the team will have 4 modules and around 15 sub modules and further functionalities.

Step 7- After this we will Estimate the effort for each task and estimate further duration of each task.

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

SOL:Test Report is a document which contains a summary of all test activities and final test results of a testing project. Test report is an assessment of how well the Testing is performed. Based on the test report, stakeholders can evaluate the quality of the tested product and make a decision on the software release.

Rather than create separate test cases, or different sets of steps, it is possible to capture the variables in the testing process as Test Parameters, which can be applied by testers during different executions of the same Test Case. Test Parameters are placeholders for actual values. They enable test case authors to write test cases and steps in an abstract way, thereby enabling testers to execute them with specific variations.

For example, suppose you are testing a logon operation for a web application. Your steps might be as follows:

- (i).Open Browser on your computer
- (ii).Access the URL of server
- (iii).Enter User name and Password in respective fields
- (iv).Click the Login button
- (v).Suppose you want to run this test using different browsers/versions and/or different user accounts. Rather than write separate test cases with similar steps, Test Parameters named Browser, Username and Password can be created and inserted into test steps when writing up a test case.
- (vi).Authors/planners of Test Runs can supply values for the Test Parameters in the Test Cases planned into individual manual Test Runs. Examples of parameter values in the above simple test might be

(vii).Firefox 40, sampleuser, and samplepass in one Test Run, and Chrome, sampleuser, and samplepass in another Test Run. During execution of Test Runs, testers see these values and perform the steps accordingly. Test Records are automatically created showing the step results with Test Parameter values applied.

13. What are the test management tools?

SOL: (i). Zephyr Scale-Zephyr Scale is frequently praised by its users for being easy to use yet incredibly flexible, requiring only minimal effort to get started. Besides tight integration with Jira, Zephyr Scale defining feature from similar tools is that its test entities are not based on Jira Issue types.

This allows for a greater degree of customization and standardization, making Zephyr Scale an especially great fit for testing and QA teams working in regulated industries with tight audit trail requirements.

(ii).PractiTest-PractiTest is a SaaS end-to-end QA management system with some of the most advanced and interesting features. With PractiTest, Testers are able to focus on quality and their actual work rather than side tasks.

Focusing on efficiency and visibility, PractiTest's dashboards and reports set a new standard of data intelligence. Using their unique and customizable filters you can efficiently organize your requirements, create & run tests (Automated, Scripted, and Exploratory), track bugs and generate reports.

(iii).QACoverage-It is a cost-effective, new generation test management solution with innovative vision and workflow to manage Software Test Life Cycles more effectively and efficiently.

It boosts test productivity and provides visibility to better handle and control your established QA process via comprehensive Agile, Requirements Management, Test Design, Test Execution, Defects Management, and Metrics & Reporting modules.

(iv). Testpad-Testpad is an online tool that takes a different approach to manual test management. Instead of managing cases one at a time, you create free-form structured checklists. This lends itself to a wide range of testing styles, including traditional test cases with steps and expected outcomes, but also works well for steering exploratory testing, the manual side of agile testing, syntax highlighting BDD given/when/then, or just ad-hoc where you write a simple list of things to check.

(v). TestMonitor-TestMonitor is an end-to-end test management tool for every organization. A simple, intuitive approach to testing. Whether you're implementing enterprise software, need QA, building a quality app or just need a helping hand in your test project, TestMonitor has you covered.

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

SOL: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.

Admin user can manage test cases assignment task.

:-Following are the steps to create a Test Case –

Step 1 – Navigate to Test Specification → Test Specification from the dashboard.

Step 2 – Select the test suite from the tree structure on the left side panel on the screen.

Step 3 – Click the Setting type icon known as Actions on this page, present on the right panel of the screen.

Step 4 – Click the + icon of Test Case Operations to create a new test case..

Step 5 – Enter the following fields in the form –

Name of Test Case

Summary

Pre-conditions

Select Status, Importance and Execution type

Enter the estimated execution type in minutes

Keywords – It displays all the available keywords. You can assign them based on the relation.

Step 6 – Click the Create button. Once you click the Create button, you can see the test case in the tree structure on the left side panel as well as the details are available on the right side of the screen.

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

SOL:

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

Step 2 – Select the project and the folder on the left pane where you want to import the test suite.

OR

To import only test cases, 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 the test suite operations.

Step 4 – Click the Import icon

OR

To upload only test cases, click the Import icon present in Test Case Operations.

It opens the Import Test Suite page. By default, the file type is selected as XML.

Step 5 – Fill in the following details –

File – Select the XML file and upload it. Max size of the file should be 400KB.

Consider test case as duplicate if: Select the appropriate option from the list.

Action for duplicate: Select the appropriate option from the list.

Step 6 – Click the Upload file button.

Step 7 – After import, a successful message will be displayed.

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

SOL:Severity is defined as the degree of impact a Defect has on the development or operation of a component application being tested. Higher effect on the system functionality will lead to the assignment of higher severity to the bug.

1. Higher effect on the system functionality will lead to the assignment of higher severity to the bug
2. Defect severity can be categorized into four class (Blocker,Critical, Major, Minor, Trivial)
3. Severity status is based on the technical aspect of the product
4. Severity indicates the seriousness of the defect on the product functionality

Priority is defined as the order in which a defect should be fixed. Higher the priority the sooner the defect should be resolved.

- (i) Priority is defined as the order in which a defect should be fixed.
- (ii). Higher the priority the sooner the defect should be resolved.
- (iii). Defects that leave the software system unusable are given higher priority over defects that cause a small functionality of the software to fail.
- (iv). Defect priority can be categorized as (Low, Medium, High)or P0,P1 and P2.
- (v). Priority status is based on the customer requirements

17. While placing an order for clothing website, in order confirmation page there is a logo error. It is a?

- (i). High priority, high severity
- (ii). Low severity low priority
- (iii). 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.)
- (iv). High severity low priority

SOL: (iii). Low severity, high priority

18. Website home page failed to load.

- (i). 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)
- (ii). Low severity low priority
- (iii). Low severity, high priority
- (iv). High severity low priority

SOL: (i). High priority, high severity

19. The application works perfectly for 50k sessions but begins to crash after a higher number of sessions.

- (i). Low severity low priority
- (ii). High priority, high severity
- (iii). Low severity, high priority
- (iv). High severity low priority (This problem needs to be fixed but not immediately.)

SOL: (iv). 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.

- (i). 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.
- (ii). High priority, high severity
- (iii). Low severity, high priority
- (iv). High severity low priority

SOL: (i). Low severity low priority

21. Find bugs and report the same on JIRA for modules in the website:

<https://www.phptravels.net/login>

Email : user@phptravels.com

Password : demouser

SOL: Given JIRA credentials are not working

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

SOL: SIGN UP

Login

Forgot Password

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

SOL: Given TestLink credentials are not working

24. Write Test Cases for Search functionality on TestLink.

SOL: Given TestLink credentials are not working