

Manual Testing Real Time Interview Questions

With answers

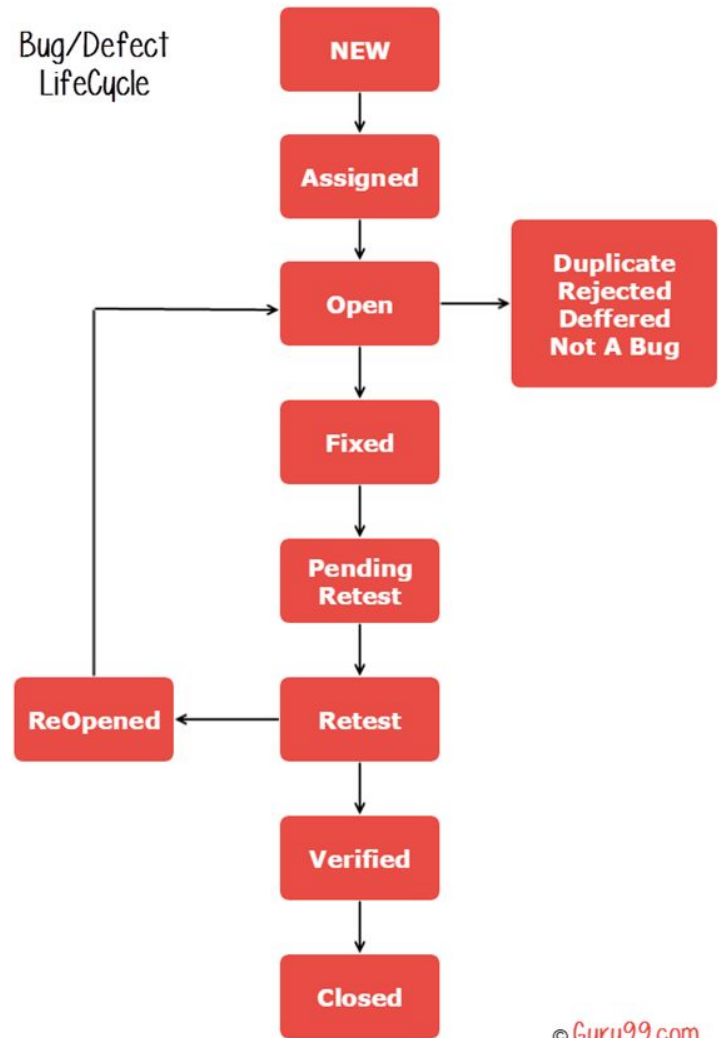
Part - 5

By : Jaikishan Mohanty

41.WHAT IS DEFECT LIFE CYCLE?

Defect Life Cycle or **Bug Life Cycle** is the specific set of states that a Bug goes through from discovery to defect fixation.

Bug Life Cycle phases/status:- The number of states that a defect goes through varies from project to project. Below lifecycle diagram, covers all possible states:



- o **New:** When a new defect is logged and posted for the first time. It is assigned a status as NEW.
- o **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.
- o **Open:** The developer starts analyzing and works on the defect fix.
- o **Fixed:** When a developer makes a necessary code change and verifies the change, he or she can make bug status as "Fixed."
- o **Pending retest:** after fixing the defect the developer gives a particular code for retesting the code to the tester. Here the testing is pending on the testers end, the status assigned is "pending request."
- o **Retest:** Tester does the retesting of the code at, 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 assign to a bug is "Not a bug".

42. How to design test cases? Or tell me different Test Case Design Technique?

We are designing the Test Cases using black box type Test Case Design Technique like

- State Transition Testing
- Boundary Value Analysis
- Equivalence Class Partitioning
- By using Scenario's
- By using Use Case's
- Pair Wise Testing

43. Tell me about State Transition Testing?

Testing the change in state e.g. from on to off, open to close etc. of the application is called as State Transition Testing

44. Tell me about BVA (Boundary Value Analysis)?

BVA is a Test Case Design Technique where test cases are selected at the edges of the equivalence class.

For example if one InputBox accept a number from 1 to 1000 then in that case our test cases would be

- 1) Test cases with test data exactly as the input boundaries of input domain i.e. values 1 and 1000 in our case.
- 2) Test data with values just below the extreme edges of input domains i.e. values 0 and 999.
- 3) Test data with values just above the extreme edges of input domain i.e. values 2 and 1001.
- 4) Test data with values !@#\$%# and sdf24234 .

45. Tell me about Equivalence Class Partitioning?

Equivalence Class partitioning is a Black Box Test Case Design Technique where input data is divided into different equivalence data classes. This method is typically used to reduce the total number test cases to a finite set of testable test cases, still covering maximum requirements.

For example if one InputBox accept a number from 1 to 1000 then Equivalence Class Partitioning in this case is

1. Class For Accepting Valid input : number from 1 to 1000
2. Class For Accepting Invalid Input : number <1 or >1000
3. Class For Accepting Invalid input : entering special characters, alphanumeric value i.e. @#@#@#, blank value etc.

46. Difference between functional specification and requirement specification. Do we need both in order to write the test cases?

Requirement Specification or Business Requirement Document (SRS or BRD) contains all the requirements from the business while FSD is a document which contains a list of functionality for the given Requirement. BRD is commonly used by Business People while FSD is used by the engineering team.

- We are using FSD in order to write down the test cases.

47. IF $a \geq 10$ && $B \leq 5$ Then some line of code Write down the test case for this?

Here we can use Test Case Design Technique to write down the test cases

1. Test the given line of code when the value of $a=10$ and $B=5$ i.e. at the boundary
2. Test the given line of code when the value of $a > 10$ and $b < 5$ i.e within the boundary
3. Test the given line of code when the value of $a=9$ and $b=6$ i.e. just outside the boundary
4. Test the given line of code when the value of $a=\wedge\&\%$ and $b=@\$@\#$ i.e. checking for invalid input

48. WHAT IS EXIT AND ENTRY CRITERIA?

ENTRY→

It describes when to start testing i.e. what we have to test it should be stable enough to test.

Ex:- if we want to test home page, the SRS/BRD/FRD document & the test cases must be ready and it should be stable enough to test.

EXIT→

It describes when to stop testing i.e. once everything mentioned below is fulfilled then s/w release is known as exit criteria:-

- a. Followed before actually releasing the s/w to client. Checking computer testing is done or not.
- b. Documents checking:- test matrix (RTM)/summary reports.

SUSPENSION CRITERIA→ when to stop testing temporarily.

49. WHAT IS BLOCKER?

A blocker is a bug of high priority and high severity. It prevents or blocks testing of some other major portion of the application as well.

50. MONKEY/AD-HOC TESTING?

It is an informal testing performed without a planning or documentation and without having knowledge of the applications/software functionalities.

Monkey testing is a type of testing that is performed randomly without any predefined test cases or test inputs.