**TESTING FUNDAMENTALS**

**1. Equivalence partitioning is:**   
 a. A black box testing technique used only by developers  
 b. A black box testing technique than can only be used during system testing  
 c. A black box testing technique appropriate to all levels of testing  
 d. A white box testing technique appropriate for component testing

**2. Cyclomatic Complexity method comes under which testing method.**   
 a. White box  
 b. Black box  
 c. Green box  
 d. Yellow box

**3. Which of the following is not a characteristic for Testability?**   
 a. Operability  
 b. Observability  
 c. Simplicity  
 d. Robustness

**4. How much testing is enough**   
 a. This question is impossible to answer  
 b. The answer depends on the risks for your industry, contract and special requirements  
 c. The answer depends on the maturity of your developers  
 d. The answer should be standardized for the software development industry

**5. The purpose of requirement phase is**   
 a. To freeze requirements  
 b. To understand user needs  
 c. To define the scope of testing  
 d. All of the above

6. Which one is not tester responsibilities ?  
 a. Assure the process for contracting software is adequate  
 b. Review the adequacy of the contractors test plan  
 c. Perform acceptance testing on the software  
 d. Assure the ongoing operation and maintenance of the contracted software  
 e. None of the above

7. In which step of SDLC project termination could be done ?

1. Design phase
2. System Maintenance phase
3. Feasibility Study phase
4. Coding phase

8. The fundamental objective of system analysis is to:

1. understand computer hardware
2. train managers in mathematical analysis
3. study and understand a complex system and modify it in some way
4. run simulation

9. Order numbers on a stock control system can range between 10000 and 99999 inclusive. Which of the following inputs might be a result of designing tests for only valid equivalence classes and valid boundaries:

1. 1000, 5000, 99999
2. 9999, 50000, 100000
3. 10000, 50000, 99999
4. 10000, 99999
5. 9999, 10000, 50000, 99999, 10000

10. Non-functional system testing includes:

1. testing to see where the system does not function properly
2. testing quality attributes of the system including performance and usability
3. testing a system feature using only the software required for that action
4. testing a system feature using only the software required for that function
5. testing for functions that should not exist

11. Which of the following is NOT a black box technique:

1. Equivalence partitioning
2. State transition testing
3. LCSAJ
4. Syntax testing
5. Boundary value analysis

12. Beta testing is:

1. Performed by customers at their own site
2. Performed by customers at their software developer’s site
3. Performed by an independent test team
4. Useful to test bespoke software
5. Performed as early as possible in the lifecycle

13. The cost of fixing a fault:

1. Is not important
2. Increases as we move the product towards live use
3. Decreases as we move the product towards live use
4. Is more expensive if found in requirements than functional design
5. Can never be determined

14. The difference between re-testing and regression testing is

1. re-testing ensures the original fault has been removed; regression testing looks for unexpected side-effects
2. re-testing looks for unexpected side effects; regression testing is repeating those tests
3. re-testing is done after faults are fixed; regression testing is done earlier
4. re-testing uses different environments, regression testing uses the same environment
5. re-testing is done by developers, regression testing is done by independent testers

15. Which of the following is not a state of a defect in defect life cycle:

A) New  
 B) Open  
 C) Verified  
 D) Deferred  
 E) Critical

16. Agile Software Development is based on

a) Incremental Development  
 b) Iterative Development  
 c) Linear Development  
 d) Waterfall Model  
 e) Both a and b

17. Select the option that suits the Manifesto for Agile Software Development  
a) Individuals and interactions  
b) Working software  
c) Customer collaboration  
d) Responding to change  
e) All of the mentioned

18. Which of the following term describes testing?  
 a) Finding broken code  
 b) Evaluating deliverable to find errors  
 c) A stage of all projects  
 d) None of the mentioned

19. Which of the following is not a software testing generic characteristics?  
a) Different testing techniques are appropriate at different points in time  
b) Testing is conducted by the developer of the software or an independent test group  
c) Testing and debugging are different activities, but debugging must be accommodated in any testing strategy  
d) None of the mentioned

20. What kind of quality cost is incurred when an error is detected in a product prior to shipment?  
 a) Prevention  
 b) Internal Failure  
 c) External Failure  
 d) Appraisal

21. Which of these does not account for software failure ?  
 a) Increasing Demand  
 b) Low expectation  
 c) Increasing Supply  
 d) Less reliable and expensive.

22. Which of the following statements explains portability in non-functional requirements?  
a) It is a degree to which software running on one platform can easily be converted to run on another platform.  
b) It can be enhanced by using languages, OS’ and tools that are universally available and standardized.  
c) The ability of the system to behave consistently in a user-acceptable manner when operating within the environment for which the system was intended.  
d) Both a and b  
e) It refers to the level at which a software system uses scarce computational resources, such as CPU cycles, memory, disk space, buffers and communication channels.

23. Which is one of the most important stakeholder from the following ?  
 a) Entry level personnel  
 b) Middle level stakeholder  
 c) Managers  
 d) Users of the software

24.Which of the following does not apply to agility to a software process?  
a) Uses incremental product delivery strategy  
b) Only essential work products are produced  
c) Eliminate the use of project planning and testing

d) Work on the feedback provided by the customers.

25. The longer a fault exists in software  
a) the more tedious its removal becomes  
b) the more costly it is to detect and correct  
c) the less likely it is to be properly corrected  
d) All of the mentioned

**Subjective type of question??**

1) Why agile model is better than waterfall model ??

Agile model is better than waterfall model because :

|  |  |
| --- | --- |
| **WATERFALL MODEL** | **AGILE MODEL** |
| * Linear approach | * Iterative approach |
| * All requirements are freezed at the first phase of requirement gathering. | * Requirements can get changed according to customer needs at later stages. |
| * Full scope of the product is known in advance. | * Full scope is not known in advance, requirements get modified according to customer needs. |
| * Work product is not available in early stages of development. | * Work product is delivered to the customers at an early stage. |
| * No frequent customer interactions. | * Frequent customer interactions. |

2) Point out the benefits of Early testing ??

Benefits of early testing are:

* Defects detected in early phases of development are easy and less expensive to fix.
* Most of the defects found in testing phase are generated in the early phases.
* Early testing can give early indicators about software fulfillment and quality.
* Early testing is beneficial so that testing is not seen as a bottleneck to release of the product.
* Testers get to know about the software in more details as they are involved at an early stage with the software.

3) Explain the validation scenario of a login screen ?

The scenario are as follows:

* Verify that the login page has space for writing user name and password, and a submit button.
* Check for valid user name and password(the user logs in).
* Check for invalid user name and password(the user is not able to login).
* Verify that a message gets displayed if the user name or password field is left empty.
* Verify if there is a check box labeled “remember password”.
* Verify that the password is in encrypted form when entered.
* Verify that there is a reset button.
* Verify that the password follows a particular set of defined rules.

4) I worked for 1 year as a Developer and next morning I wake up as a Test Engineer. How should I react over this? Explain the pros and cons.

The pros and cons of being a tester are:

pros: -Easy to learn

-get to understand the whole project well

-more fault finding, better the quality of the product being produced.

cons: -repetitive job tasks

-comparative to development, very less new things to learn about.

5) Explain the bug life-cycle and explain the cases of deferring any bugs.

Defect life cycle is a cycle which a defect goes through during its lifetime. It starts when defect is found and ends when a defect is closed, after ensuring it’s not reproduced. Defect life cycle is related to the bug found during testing. Various stages in bug life cycle are:

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 status means the developer accepted the bus, but it is scheduled to rectify in the next build.

* A defect that is confirmed by the team to be a bug but the effort to correct it at this time exceeds the ROI (Return On Investment). This can often be the case when a re-design is scheduled for that area of the application, or if technology barriers exist that make correction of the bug prohibitive.
* When the bug is not causing any serious problem, and it will  
  be fixed in the later version of the build. In such cases  
  the developer will change the status to "Deferred".