What is Exploratory Testing?

- This testing is mainly performed when the tester has no access of requirement documentation.

1. **What is traceability matrix?**

- It is agarph of requirement vs component tht one should able to trace back from every system component to the orignl requirement.

2. **What is Boundary value testing?**

- Boundary value analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges.

3. **What is Equivalence partitioning testing?**

- Equivalence partitioning is the process of defining the optimum number of tests by:
- Reviewing documents such as the Functional Design Specification and Detailed Design Specification, and identifying each input condition within a function,
- Selecting input data that is representative of all other data that would likely invoke the same process for that particular condition

4. **What is Integration testing?**

- Integration testing tests integration or interfaces between components, interactions to different parts of the system such as an operating system, file system and hardware or interfaces between systems

5. **What determines the level of risk?**

- Risk analysis should be used to determine what to test in each component and just as importantly what not to test

6. **What is Alpha testing?**

- Alpha Testing is definitely performed and carried out at the developing organizations location with the involvement of developers
- Alpha Testing is not open to the market and public

7. **What is beta testing?**

- It is always performed by the customers at their own site.
- Beta Testing (field testing) is performed and carried out by users or you can say people at their own locations and site using customer data.

8. What is component testing?

- Unit Testing is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.

9. What is functional system testing?

- Functional System Testing : A requirement that specifies a function that a system or system component must perform

10. What is Non-Functional Testing?

- Non-Functional Testing: Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability and portability

11. What is GUI Testing?

- Graphical User Interface (GUI) testing is the process of testing the system's GUI of the System under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc.

12. What is Ad hoc testing?

- This testing is primarily performed if the knowledge of tester in the system under test is very high.

13. What is load testing?

- Its a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system's response time degrades or fails

14. What is stress Testing?

- Stress testing is used to test the stability & reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions.

15. What is white box testing and list the types of white box testing?

- Testing based on an analysis of the internal structure of the component or system.
- Types of White Box Testing.
    - Statement Coverage

- Decision coverage
- Condition coverage

16. What is black box testing? What are the different black box testing techniques?

- Testing either functional or non-functional, without reference to the internal structure of the component or system.
- Types of Black Box Testing Techniques.
  - Equivalence partitioning
  - Boundary value analysis
  - Decision tables
  - State transition testing
  - Use-case Testing
  - Other Black Box Testing
    - Syntax or Pattern Testing

17. Mention what bigbang testing is?

- In Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.

18. What is the purpose of exit criteria?

- Exit criteria is used to determine when testing at any stage is complete The set of generic and specific conditions, agreed upon with the stakeholders, for permitting a process to be officially completed

19. When should "Regression Testing" be performed?

- Testing of a previously tested program following modification to ensure that defects have not been introduced or uncovered in unchanged areas of the software, as a result of the changes made. It is performed when the software or its environment is changed.

20. What is 7 key principles? Explain in detail?

- **7 key Principles**
  - **Testing shows presence of defects**
  - **Exhaustive testing is impossible**
  - **Early testing**
  - **Defect clustering**
  - **The pesticide paradox**

- **Testing is context depedent**
- **Absence of error fallacy**

- **Testing Shows Presence of Defects**
  - Testing can show that defects are present, but cannot prove that there are no defects.
  - Testing reduces the probability of undiscovered defects remaining in the software but, even if no defects are found, it is not a proof of correctness.
- **Exhaustive testing is impossible**
  - Testing everything including all combinations of inputs and preconditions is not possible. So, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts.
  - For example: In an application in one screen there are 15 input fields, each having 5 possible values, then to test all the valid combinations you would need 30 517 578 12^5 (51^5) tests.
  - This is very unlikely that the project timescales would allow for this number of tests.
- **Early testing**
  - Testing activities should start as early as possible in the software or system development life cycle, and should be focused on defined objectives.
  - Testing activities should start as early as possible in the development life cycle
  - These activities should be focused on defined objectives – outlined in the Test Strategy
  - Remember from our Definition of Testing, that Testing doesn't start once the code has been written.
- **Defect clustering**
  - A small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures.
  - Defects are not evenly spread in a system
  - They are clustered
  - In other words, most defects found during testing are usually confined to a small number of modules
  - Similarly most operational failures of a system are usually confined to a small number of modules
  - An important consideration in test prioritisation
- **The pesticide paradox**
  - If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects.

- To overcome this “pesticide paradox”, the test cases need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.
- Testing identifies bugs, and programmers respond to fix them
- As bugs are eliminated by the programmers, the software improves
- As software improves the effectiveness of previous tests erodes

➢ **Testing is context depedent**

- Testing is basically context dependent. Testing is done differently in different contexts Different kinds of sites are tested differently.
- For example Safety – critical software is tested differently from an e-commerce site.
- While, Testing can be 50% of development costs, in NASA's Apollo program it was 80% testing
- 3 to 10 failures per thousand lines of code (KLOC) typical for commercial software

➢ **Absence of error fallacy**

- If the system built is unusable and does not fulfill the user’s needs and expectations then finding and fixing defects does not help.
- If we build a system and, in doing so, find and fix defects ....
- It doesn’t make it a good system
- Even after defects have been resolved it may still be unusable and/or does not fulfil the users’ needs and expectations

21. Difference between QA v/s QC v/s Tester

S_No	Quality Assurance	Quality Control	Testing
1	Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.	Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements.	Activities which ensure the identification of bugs/error/defects in the Software.
2	Focuses on processes and procedures rather than conducting actual testing on the system.	Focuses on actual testing by executing Software with intend to identify bug/defect through implementation of procedures and process.	Focuses on actual testing.
3	Process oriented activities.	Product oriented activities.	Product oriented activities.
4	Preventive activities.	It is a corrective process.	It is a preventive process.
5	It is a subset of Software Test Life Cycle (STLC).	QC can be considered as the subset of Quality Assurance.	Testing is the subset of Quality Control.

22. Difference between Smoke and Sanity?

Smoke	Sanity
Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine.	Sanity Testing is done to check the new functionality / bugs have been fixed
The objective of this testing is to verify "stability" of the system in order to with more rigorous testing.	The objective of this testing is to verify the "rationality" of the system in order proceed to proceed with more rigorous testing.
The testing is performed by the developers or testers	Sanity testing is usually performed by testers.
Smoke testing is usually documented or scripted.	Sanity testing is usally not documented and unscripted
Smoke testing is a subset of regression testing.	Sanity testing is a subset of acceptance testing
Smoke testing excercises the entire system from end to end	Sanity testing excercises only the particular component of the entire system.
Smoke testing is like general health check up	Sanity testing is like specialized health check up

23. Difference between verification and Validation

Verification	Validation
The process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase.	The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements.
To ensure that the product is being built according to the requirements and design specifications. In other words, to ensure that work products meet their specified requirements.	To ensure that the product actually meets the user's needs, and that the specifications were correct in the first place. In other words, to demonstrate that the product fulfills its intended use when placed in its intended environment.

24. Explain types of Performance testing.

- Load testing
- Stress testing
- Endurance testing
- Spike testing
- Volume testing
- Scalability testing

25. What is Error, Defect, Bug and failure?

- A mistake in coding is called error, error found by tester is called defect, defect accepted by development team then it is called bug, build does not meet the requirements then it is failure.

26. What is Bug Life Cycle?

- The duration or time span between the first time defects is found and the time that it is closed successfully, rejected, postponed or deferred is called as Defect Life Cycle.

27. Explain the difference between Functional testing and Non-Functional testing

FUNCTINAL TESTING	NON-FUNCTIONAL TESTING
Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements.	Non-Functional testing checks the Performance, reliability, scalability and other non-functional aspects of the software system.
Functional testing is executed first	Non-functional testing should be performed after functional testing
Manual testing or automation tools can be used for functional testing	Using tools will be effective for this testing
Business requirements are the inputs to functional testing	Performance parameters like speed , scalability are inputs to non-functional testing.
Functional testing describes what the product does	Non-Functional testing describes how good the product works
Easy to do manual testing	Tough to do manual testing
Types of Functional testing are. • Unit Testing • Smoke Testing • Sanity Testing • Integration Testing • White box testing • Black Box testing • User Acceptance testing • Regression Testing	Types of Non-functional testing are. • Performance Testing • Load Testing • Volume Testing • Stress Testing • Security Testing • Installation Testing • Penetration Testing • Compatibility Testing • Migration Testing

28. What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

SDLC (Software Development Life Cycle)	STLC (Software Testing Life Cycle)
SDLC is mainly related tosoftware development	STLC is mainly related tosoftware testing
Besides development other phases like testing is alsoincluded	It focuses only on testing thesoftware
SDLC involves total six phasesor steps	STLC involves only five phasesor stpes

In SDLC, more number of members (developers) are required for the whole process	In STLC less number of member(testers) are needed
Goal of SDLC is to complete successful development of software	In STLC testing team (test lead ortest architect) makes the plans and designs
In SDLC development team makes the plans and designsbased on the requirements	Goal of STLC is to completesuccessful testing of software
It helps in developing goodquality software	It helps in making the softwaredefects free

29. What is the difference between test scenarios, test cases, and test script?

Test case	**Test script**
A test case is high document with instruction on the specific functionality of the softwareproduct to be tested	Test script a step-by-step instruction to test each softwareproduct's functionality (test case)
A test case is the software development life cycle's 'whatto test' component	Test script is the software development life cycle's 'how totest' component
Test case are written in simpleenglish	Test script are written in programming languages like VB scrips, python, java, etc
A test is a document with instruction on testing the specific functionality of an application	Test script is a program that runsvarious test date on the functionality of an application
Test scenario serve as an outline for writing test cases	Test case serves as an outline forwriting test scripts

30. When to used Usablity Testing?

- Usability Testing identifies usability errors in the system early in development cycle and can save a product from failure.