1. What is Exploratory Testing?

Ans: Though the current trend in testing is to push for automation, exploratory testing is new way of thinking.

It is more structured than Error guessing.

2. What is traceability matrix?

Ans : It is a graph of requirement vs component that you should able to trace back from every system component to the requirement.

Types: forward

Backward

Bi-directional

3. What is boundary value testing?

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

Boundary value analysis is a method which refines equivalence partitioning.

4. What is Equivalence partitioning testing?

Ans: In equivalence partitioning Aim is to treat groups of inputs as equivalent and to select one representative input to test them all.

EP says that by testing just one value we have tested the partition.

It assumes that: If one value finds a bug, the others probably will too

If one doesn't find a bug, the others probably won't either

5. What is integration testing?

Ans: Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems

Integration Testing is a level of the software testing process where individual units are combined and tested as a group.

6. What determines the level of risk?

Ans: 'A factor that could result in future negative consequences; usually expressed as impact and likelihood'

Risks are of two types: Project Risk

Product Risk

7. What is Alpha testing?

Ans: It is the form of Acceptance Testing.

It is always performed by the developers at the software development site.

8. What is beta testing?

Ans: It is also the form of Acceptance Testing.

It is always performed by the customers at their own site.]

9. What is component testing?

Ans: A minimal software item that can be tested in isolation. It means "A unit is the smallest testable part of software."

Unit Testing is a level of the software testing process where individual units/components of a software/system are tested.

10. What is functional system testing?

Ans: A requirement that specifies a function that a system or system component must perform.

Accuracy: Provision of right or agreed results or effects

Interoperability: Ability to interact with specified systems

Compliance: Adhere to applicable standards, conventions, regulations or laws

Auditability: Ability to provide adequate and accurate audit data

Suitability   : Presence and appropriateness of functions for specified tasks

11. What is Non-Functional Testing?

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

It is the testing of "how" the system works.

12. What is GUI Testing?

Ans: 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.

Approach of GUI Testing   : MANUAL BASED TESTING

RECORD AND REPLAY

MODEL BASED TESTING

13. What is Adhoc testing?

Ans: Adhoc testing is an informal testing type with an aim to break the system. It does not follow any test design techniques to create test cases. Main aim of this testing is to find defects by random checking. adhoc testing can be achieved with the testing technique called Error Guessing.

14. What is load testing?

Ans: It's a performance testing to check system behaviour 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.

Load testing is a kind of performance testing which determines a system's performance under real-life load conditions. This testing helps determine how the application behaves when multiple users access it simultaneously.

15. What is stress Testing?

Ans: Stress testing is used to test the stability & reliability of the system.

It even tests beyond the normal operating point and evaluates how the system works under those extreme conditions.

Stress Testing is done to make sure that the system would not crash under crunch situations.

Stress testing is also known as endurance testing.

Stress testing is to determine the limit, at which the system or software or hardware breaks.

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

Ans: Testing based on an analysis of the internal structure of the component or system.

Structure-based testing technique is also known as 'white-box' or 'glass-box' testing technique because here the testers require knowledge of how the software is implemented, how it works.

White box testing is also called glass testing or open box testing.

- The basic coverage measure is where the 'coverage item' is whatever we have been able to count and see whether a test has exercised or used this item.
- Types of Coverage The different types of coverage are:
  Statement coverage

Decision coverage

Condition coverage

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

Ans : Black-box testing: Testing, either functional or non-functional, without reference to the internal structure of the component or system. Specification-based testing technique is also known as 'black-box'.

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

18. Mention what are the categories of defects?

Ans: here are the types of defects:

- Data quality/ database defects
- Critical functionality defects
- Functionality defects
- Security defects
- Ui defects

19. Mention what big bang testing is?

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

Big Bang testing has the advantage that everything is finished before integration testing starts. Here all component are integrated together at once, and then tested.

20. What is the purpose of exit criteria?

Ans: Exit Criteria defines the items that must be completed before testing can be conclude.

Purpose of exit criteria is: to define when we STOP testing either at the:

End of all testing – i.e. product Go Live

End of phase of testing (e.g. hand over from System Test to UAT)\

21. When should "Regression Testing" be performed?

Ans: It is performed when the software or its environment is changed.

Regression Testing: 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.

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

Ans: General key principle

1. Testing shows presence of Defect: 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.

2. Exhaustive Testing is Impossible!: Testing everything including all combinations of inputs and preconditions is not possible. Exhaustive testing of complex software applications: requires enormous resources,it is too expensive and takes too long.

3. 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

4. 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'

5. The Pesticide Paradox: 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.

Therefore we must learn, create and use new tests based on new techniques to catch new bugs

6. Testing is Context Dependent: Testing is basically context dependent. Testing is done differently in different contexts .Different kinds of sites are tested differently

7. Absence of Errors 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.

23. Explain types of Performance testing.

Ans: Software performance testing is a means of quality assurance (QA). It involves testing software applications to ensure they will perform well under their expected workload.

Types of Performance Testing

● Load testing: Load time is normally the initial time it takes an application to start. It is a performance testing to check system behaviour 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.

● Stress testing: Stress Testing is done to make sure that the system would not crash under crunch situations. Stress testing is also known as endurance testing.

● Endurance testing

● Spike testing

● Volume testing

● Scalability testing

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

Ans: Error- A mistake in coding is called error

Defect- error found by tester is called defect

Bug- defect accepted by development team then it is called bug

Failure- build does not meet the requirements then it is failure

25. What is Bug Life Cycle?

Ans: "A computer bug is an error, flaw, mistake, failure, or fault in a computer program that prevents it from working correctly or produces an incorrect result. Bugs arise from mistakes and errors, made by people, in either a program's source code or its design."

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'.
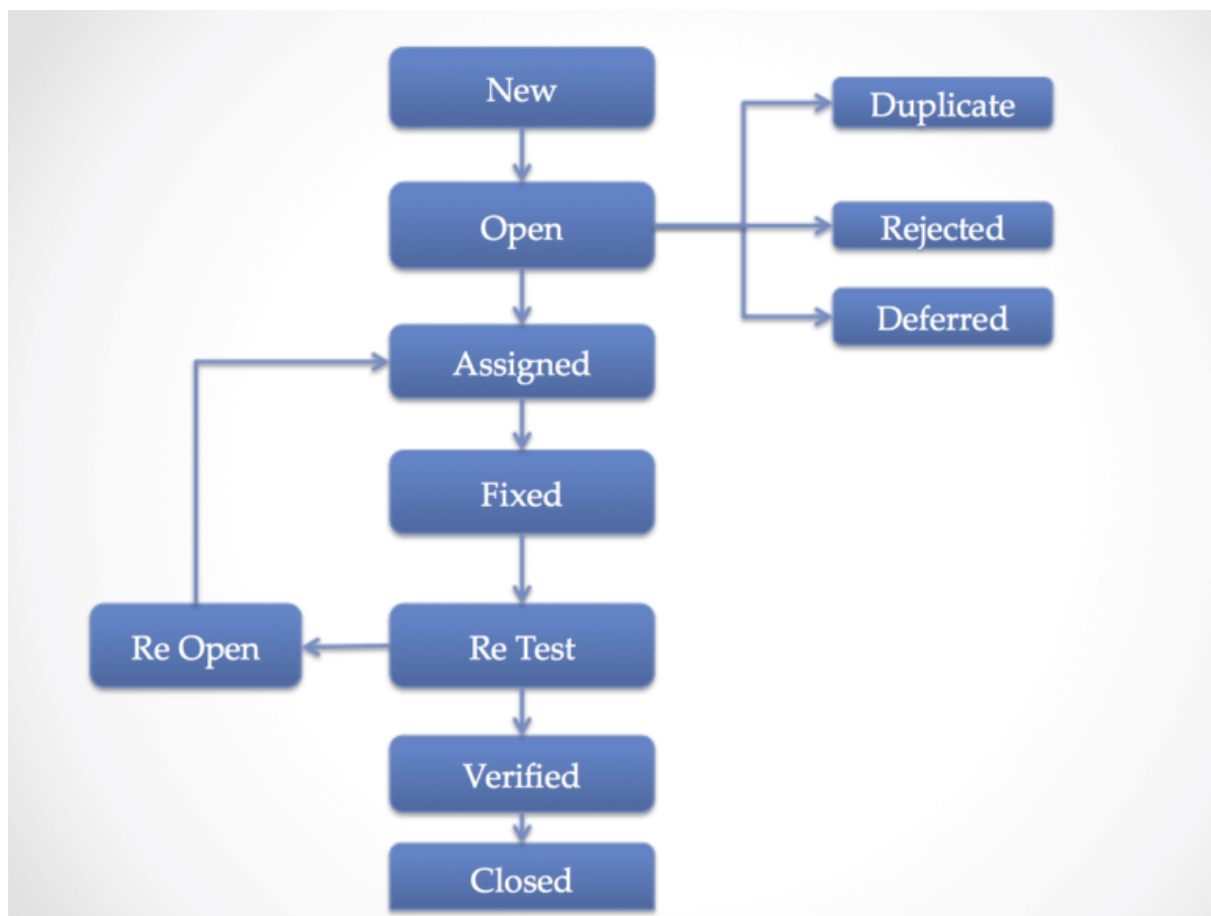
26. What is Priority?

Ans: Priority is Relative and Business-Focused. Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the developer has to fix it at the earliest. The priority status is set based on the customer requirements.

Priority can be of following types:

- Low
- Medium
- High
- Critical

27. What is severity?

Ans: Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software. In other words it defines the impact that a given defect has on the system.

Severity can be of following types:

- Critical
- Major (High)
- Moderate(medium)
- Minor (Low)
- Cosmetic

28. Bug categories are…

Ans: as same as question number one.

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

Ans:

	QA	QC	Testing
Purpose	Setting up adequate processes, introducing the standards of quality to prevent the errors and flaws in the product	Making sure that the product corresponds to the requirements and specs before it is released	Detecting and solving software errors and flaws
Focus	Processes	Product as a whole	Source code and design
What	Prevention	Verification	Detection
Who	The team including the stakeholders	The team	Test Engineers, Developers
When	Throughout the process	Before the release	At the testing stage or along with the development process

30. Difference between Smoke and Sanity?

Ans:

Smoke testing	Sanity testing
Focus on the critical functionalities of an application	Check the new functionalities or bugs fixed
The main purpose is to verify that the application work as expected before going deeper with further testing	The main purpose is to verify that main behavior of newly added features or bug fixed work well as expected
Can be done by developers or testers	Usually performed by testers
Checks the entire system	Only checks particular component of the system
Is part of Acceptance testing (to check that they meet all expectations)	Is a part of Regression testing (to check all existing features are not impacted by the changes)
Wide and shallow approach	Narrow and deep approach

31. Difference between verification and Validation

Ans:

Verification	Validation
1. Verification is a static practice of verifying documents, design, code and program.	1. Validation is a dynamic mechanism of validating and testing the actual product.
2. It does not involve executing the code.	2. It always involves executing the code.
3. It is human based checking of documents and files.	3. It is computer based execution of program.
4. Verification uses methods like inspections, reviews, walkthroughs, and Desk-checking etc.	4. Validation uses methods like black box (functional) testing, gray box testing, and white box (structural) testing etc.
5. **Verification** is to check whether the software conforms to specifications.	5. **Validation** is to check whether software meets the customer expectations and requirements.
6. It can catch errors that validation cannot catch. It is low level exercise.	6. It can catch errors that verification cannot catch. It is High Level Exercise.
7. Target is requirements specification, application and software architecture, high level, complete design, and database design etc.	7. Target is actual product-a unit, a module, a bent of integrated modules, and effective final product.
8. Verification is done by QA team to ensure that the software is as per the specifications in the SRS document.	8. Validation is carried out with the involvement of testing team.
9. It generally comes first-done before validation.	9. It generally follows after **verification**.

32. Difference between Priority and Severity

Ans:

Severity	Priority
Defect Severity is specified as the degree of impact that a defect has on the operation of the product.	Defect Priority has specified the order in which the developer should fix a defect.
Severity means the seriousness of the defect in the product functionality.	Priority means how soon the bug should be fixed.
The test engineer determines the severity level of the defect.	Priority of defects is decided in discussion with the manager/client.
It is driven by functionality.	It is driven by business value.
Severity status is established on the technical aspect of the product.	Priority status is established on customer requirements.

33. Explain the difference between Functional testing and Non Functional testing.

Ans:

Functional Testing	VS	Non-Functional Testing
Test the functionality of the software.		Test the non-functional aspects or readiness of the the software including performance, usability, reliability.
It has to be done before Non-Functional Testing.		It will be done after Functional Testing completes.
It is also called as Behavioral Testing and focuses on the underlying application features.		Focuses on the performance of the application.
It can be done manually, though test cases		It's hard to do it manually. It usually need

34. What is the difference between the STLC (Software Testing Life Cycle) and SDLC

(Software Development Life Cycle)?

Ans:

STLC	SDLC
STLC is related to software testing.	The SDLC is primarily concerned with software development.
STLC fewer people are involved.	SDLC a more people involved in all processes, (number of developers).
STLC ensures that anything we produce meets customer needs and that the products are of high quality.	The SDLC ensures that we are building the correct thing in the correct manner.
STLC is a testing life cycle.	SDLC is a development life cycle.
STLC is concerned with both the development and testing processes, but it is primarily concerned with the testing process.	Software development life cycle assures that we deliver high-quality software which is as per client needs.
In STLC QA team analysis all the requirement from the requirement document and create a system test plan.	In SDLC business analysts gather all project-related requirements from a stockholder and create a development plan.
In the software testing life cycle, the Test Architect or test manager creates a strategy to test a software application.	In software development, the life cycle development team create the high-level and low-level design of project based on clients' requirement.

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

Ans:

Test Scenario	Test Case	Test Script
Is any functionality that can be tested.	Is a set of actions executed to verify particular features or functionality.	Is a set of instructions to test an app automatically.
Is derived from test artifacts like Business Requirement Specification (BRS) and Software Requirement Specification (SRS).	Is mostly derived from test scenarios.	Is mostly derived from test cases.
Helps test the end-to-end functionality in an Agile way.	Helps in exhaustive testing of an app.	Helps to test specific things repeatedly.
Is more focused on what to test.	Is focused on what to test and how to test.	Is focused on the expected result.
Takes less time and fewer resources to create.	Requires more resources and time.	Requires less time for testing but more resources for scripts creating and updating.
Includes an end-to-end functionality to be tested.	Includes test steps, data, expected results for testing.	Includes different commands to develop a script.
The main task is to check the full functionality of a software application.	The main task is to verify compliance with the applicable standards, guidelines, and customer requirements.	The main task is to verify that nothing is skipped, and the results are true as the desired testing plan.
Allows quickly assessing the testing scope.	Allows detecting errors and defects.	Allows carrying out an automatic execution of test cases.

36. Explain what Test Plan is? What is the information that should be covered?

Ans: A test plan **acts as a blueprint or instruction manual on how and what you will test for a product release or sprint.**

The test plan conveys how the test will be performed. This includes **defining test objectives, test approach, test tools, test environment, test schedules and team responsibilities and composition**.

37. Advantage of Bugzilla.

Ans:

- Open source, free bug tracking tool.
- Automatic Duplicate Bug Detection.
- Search option with advanced features.
- File/Modify Bugs By Email.
- Move Bugs between Installs.
- Multiple Authentication Methods (LDAP, Apache server).
- Time Tracking.
- Automated bug reporting; has an API to interact with system.

38. What are the different Methodologies in Agile Development Model?

Ans: The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of planning, executing, and evaluating.

There are 5 main Agile methodologies: Scrum, Kanban, Extreme Programming (XP), Lean Development e Crystal.

Scrum:
 SCRUM is an agile development method which concentrates particularly on how to manage tasks within a team based development environment.

Basically, Scrum is derived from activity that occurs during rugby match. Scrum believes in empowering the development team and advocates working in small teams (say- 7 to 9 members). It consists of three roles and their responsibilities are explained as follows:

● Scrum Master: Master is responsible for setting up the team, sprint meeting and removes obstacles to progress

● Product owner: The Product Owner creates product backlog, prioritizes the backlog and is responsible for the delivery of the functionality at each iteration

● Scrum Team: Team manages its own work and organizes the work to complete the sprint or cycle

● Sprint: Sprint is a time-boxed period in which the scrum team needs to finish the set amount of work. Each sprint has a specified timeline, i.e., 2 weeks to 1 month. The scrum team agrees with this timeline during the sprint planning meeting.

Kanban:

- Kanban is a very popular framework for development in the agile software
- Development methodology.
- It provides a transparent way of visualizing the tasks and work capacity of a team.
- It mainly uses physical and digital boards to allow the team members to visualize
- the current state of the project they are working on.
- Kanban originated in Toyota in the 1940s.
- Kanban's meaning in Japanese is "billboards."
- The Kanban board has columns and story cards.
- The columns are nothing, but workflow states and cards are nothing but a
- Demonstration of the actual task a team member is performing.

39. Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?

Ans:

Authentication	Authorization
Authentication confirms your identity to grant access to the system.	Authorization determines whether you are authorized to access the resources.
It is the process of validating user credentials to gain user access.	It is the process of verifying whether access is allowed or not.
It determines whether user is what he claims to be.	It determines what user can and cannot access.
Authentication usually requires a username and a password.	Authentication factors required for authorization may vary, depending on the security level.
Authentication is the first step of authorization so always comes first.	Authorization is done after successful authentication.
For example, students of a particular university are required to authenticate themselves before accessing the student link of the university's official website. This is called authentication.	For example, authorization determines exactly what information the students are authorized to access on the university website after successful authentication.

- Insufficient testing for browser compatibility. ...
- Failing to conduct thorough functional testing across mobile. ...
- Failing to conduct thorough functional testing across desktop. ...
- Poor data security. ...
- Failing to provide an intuitive experience.

40. When to used usability testing?

Ans: There are many software applications / websites, which miserably fail,once launched, due to following reasons –

- Where do I click next?
- Which page needs to be navigated?
- Which Icon or Jargon represents what?
- Error messages are not consistent or effectively displayed
- Session time not sufficient

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

41. What is the procedure for GUI Testing?

Ans: Graphical User Interface (GUI) testing is the process of testing the system's GUI of the System under Test.

- WHAT DO YOU CHECK IN GUI TESTING?
- Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.
- Check you can execute the intended functionality of the application using the GUI
- Check Error Messages are displayed correctly
- Check for Clear demarcation of different sections on screen
- Check Font used in application is readable
- Check the alignment of the text is proper
- Check the Colour of the font and warning messages is aesthetically pleasing
- Check that the images have good clarity
- Check that the images are properly aligned
- Check the positioning of GUI elements for different screen resolution.