

1) Why is testing required?

- To point out the defects and errors that were made during the developmental phases.
- To make sure of the customer's reliability and their satisfaction in the application.
- To ensure the quality of the product.
- Delivering high quality product or software application which requires lower maintenance cost and hence results into more accurate, consistent and reliable results.
- Effective performance of software application or product.

2) What types of applications do we test?

- web applications
- desktop/windows applications
- Mobile applications
- Web services (SOAP/REST)
- ETL jobs, database validations
- Back end/batch programs/windows services

3) What is SDLC and different phases in SDLC?

- Software development life cycle (SDLC) is a process to develop the application.
 - o Requirement gathering and analysis
 - o Design
 - o Implementation or coding
 - o Testing
 - o Maintenance

Requirement gathering and analysis: Senior team members analyze the requirements/input given by customers/business users. They will check whether the requirement is feasible or not (can be done or not). They also identify the risks associated with project.

Define/Design: Business Analyst define more details about requirements (which are in BRD) in the form of SRS (software requirement specification) or Use Case diagram. As part of design, Senior Developers write High Level Design Document (HLD), Developers write Low Level Design Document (LLD) and Seniors Tester writes the Test Planning document.

Implementation/Development: Developers write the code for the requirements and Testers write test cases as per SRS.

Testing: Execute the test cases what we prepared in previous stage

Deployment: Release the tested code to production

Maintenance: Support team monitoring the system that is running in production

4) What is waterfall in SDLC?

- The concept is similar to as in a real waterfall. Once the water starts to flow down the edge of the cliff it cannot turn back up, similarly, once a developmental phase is complete while developing the software there is no turning back and the development proceeds to the next phase!
- Following are the phases involved in the waterfall model:
 - o System and software requirements
 - o Analysis: resulting in models, business rules.
 - o Design: software architecture
 - o Coding: Development, proving and integration of software
 - o Testing: systematic and thorough inspection and debugging the defects
 - o Operations: installation, migration, support and maintenance of complete systems.

5) What is the process in agile model?

- Agile model is a combination of iterative and incremental process models with focus on adaptability of the process and customer satisfaction by rapid delivery of the working software product.
- Agile methods break the software into small incremental builds which are provided in iterations. Every iteration involves teams working simultaneously on various areas like planning, requirement analysis, design, coding, unit testing and acceptance testing.

6) What is scrum methodology?

- It is an agile model where development of software is done through incremental and iterative process.
- Many frameworks can be used to implement agile, but scrum has a unique flavor because of the commitment to short iterations of work.
- With scrum, the product is built in a series of fixed-length iterations called sprints that give teams a framework for shipping software.
- Scrum involves four processes divided in each sprint:
 - o Sprint planning- which involves what has to be completed within the next sprint.
 - o Daily scrum meeting- a 15 min mini meeting for the team to sync in.
 - o Sprint demo- a sharing meeting where the team shows what they have shipped in that sprint.
 - o Sprint retrospective- a review of what did and didn't go well with actions to take which will make the next sprint better.

7) What is daily standup meeting and what we discuss?

- The daily standup meeting is nothing but the daily 15 min scrum meeting which involves three things for everyone on the team to explain:
 - o What they did yesterday.
 - o What they are going to finish by tomorrow.
 - o And the blocks that occur every day while achieving their respective tasks.

8) What is user story/feature/sprint backlog items and tasks in user story?

- In Scrum, work is typically expressed in the Product Backlog as user stories. A team may write its user stories in a number of ways as long as they are written from the perspective of the end user. Team members are encouraged to think of their work from the perspective of who will use it.
- Sprint backlog items consist of:
 - Product backlog Items negotiated between the team and the product owner during the sprint planning meeting.
 - Product Backlog Items include functional and nonfunctional customer requirements as well as technical team generated requirements.

9) What is sprint planning and spring retro?

- The Sprint planning meeting is a negotiation between the team and the product owner about what the team will do during the next sprint.
- The sprint retrospective meeting is held at the end of every sprint after the sprint review meeting. The team and Scrum Master meet to discuss what went well and what to improve in the next sprint. The product owner does not attend this meeting.
- The sprint retrospective should be time-boxed to three hours.

10) What is burndown chart and velocity?

- A sprint burndown chart depicts the total task hours remaining per day. This shows you where your team stands regarding completing the tasks that comprise the product backlog items that achieve the goals of the sprint. The X-axis represents days in the sprint, while the Y-axis is effort remaining.
- In Scrum, velocity is how much product backlog effort a team can handle in one sprint. This can be estimated by viewing previous sprints, assuming the team composition and sprint duration are kept constant. It can also be established on a sprint-by-sprint basis, using commitment-based planning.

11) What is product backlog item and sprint backlog items?

- Product Backlog Item is a unit of work small enough to be completed by a team in one Sprint iteration. Backlog items are decomposed into one or more tasks.
- Sprint Backlog Item defines the work for a sprint, represented by the set of tasks that must be completed to realize the sprint's goals, and selected set of product backlog items.

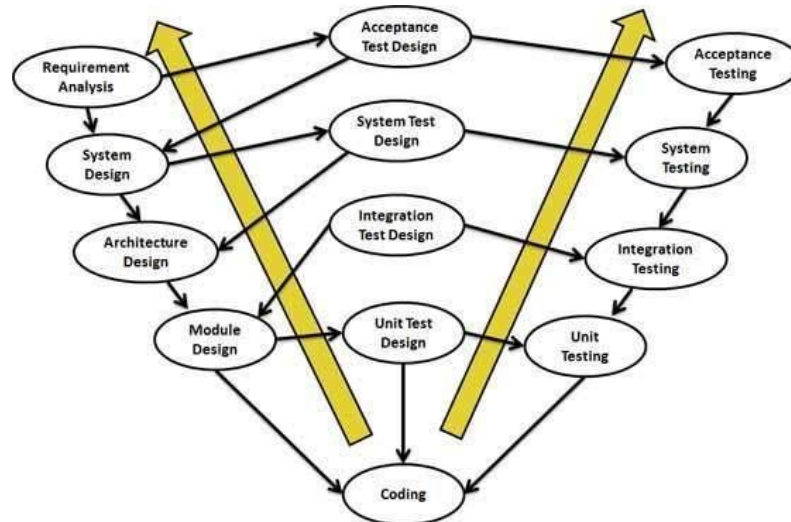
12) What is user acceptance criteria test cases?

- Acceptance Criteria is defined as "Conditions that a software product must satisfy to be accepted by the user, customer or the stakeholder. These are a set of statements, each with a clear pass/fail result, that specify both functional and nonfunctional requirements applicable at the current stage of project integration.

- o Functional Criteria: Identify specific user tasks, functions or business processes that must be in place. A functional criterion might be “A user is able to access a list of available reports.” A non-functional criterion might be “Edit buttons and Workflow buttons comply with the Site Button Design.”
- o Non-functional Criteria: Identify specific non-functional conditions the implementation must meet, such as design elements. A non-functional criterion might be “Edit buttons and Workflow buttons comply with the Site Button Design.”
- o Performance Criteria: If specific performance is critical to the acceptance of a user story, it should be included.

13) What is v model?

- The V - model is SDLC model where execution of processes happens in a sequential manner in V-shape. It is also known as Verification and Validation model. It is an extension of the waterfall model and is based on association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle there is a directly associated testing phase. This is a highly disciplined model and next phase starts only after completion of the previous phase.



- Verification Phase includes business requirement analysis, system design, architectural design, and module design.
- In the coding phase the coding is done based on the coding guidelines and standards. The code goes through numerous code reviews and is optimized for best performance before the final build is checked.
- Validation testing involves Unit testing which eliminates bugs at an early stage, Integration testing which is associated with the architectural design phase, System testing which is directly associated with the system design phase and Acceptance testing which involves testing the product in the user environment.

14) What is STLC?

- Software Testing Life Cycle is the testing process which is executed in a systematic and planned manner. In STLC process, different activities are carried out to improve the quality of the product.
- Following are the steps involved:
 - o Requirement analysis
 - o Test planning
 - o Test case development
 - o Environment setup
 - o Test execution
 - o Test cycle closure

15) What is defect?

- A defect is an error or a bug, in the application which is created. A programmer while designing and building the software can make mistakes or error. These mistakes or errors mean that there are flaws in the software. These are called defects.

16) How to arise a defect and what we specify while logging defect?

- When a tester finds a bug or defect it's required to convey the same to the developers. Thus they report bugs with the detail steps and are called as Bug Reports, issue report, problem report, etc.
- This Defect report or Bug report consists of the following information:
 - o Defect ID – Every bug or defect has it's unique identification number
 - o Defect Description – This includes the abstract of the issue.
 - o Product Version – This includes the product version of the application in which the defect is found.
 - o Detail Steps – This includes the detailed steps of the issue with the screenshots attached so that developers can recreate it.
 - o Date Raised – This includes the Date when the bug is reported
 - o Reported By – This includes the details of the tester who reported the bug like Name and ID
 - o Status – This field includes the Status of the defect like New, Assigned, Open, Retest, Verification, Closed, Failed, Deferred, etc.
 - o Fixed by – This field includes the details of the developer who fixed it like Name and ID
 - o Date Closed – This includes the Date when the bug is closed
 - o Severity – Based on the severity (Critical, Major or Minor) it tells us about impact of the defect or bug in the software application
 - o Priority – Based on the Priority set (High/Medium/Low) the order of fixing the defect can be made.

17) Defect lifecycle

- Defect life cycle, also known as Bug Life cycle is the journey of a defect cycle, which a defect goes through during its lifetime. It varies from organization to organization and

also from project to project as it is governed by the software testing process and also depends upon the tools used.

18) Different types of testing

- There are many types of testing, some of them include:
 - o Manual Testing – which is done manually and
 - o Automation Testing – which is also known as test automation, when the tester writes scripts and uses another software to test a product.
 - o Unit Testing
 - o Regression Testing
 - o Integration Testing
 - o Smoke Testing
 - o Black box, White box, Grey box testing

19) What is unit testing?

- Unit testing is a technique in which individual modules (units) are tested to determine if there are any issues by the developer himself. Unit testing is mainly automated but it can be done manually as well.

20) When do we use regression testing?

- Regression testing is a test to find out whether there are any unwanted defects created after the system has been changed.
- Software regression is any unwanted change that happens after changes are made to the system like including a new feature in a software and the software not working anymore. It may be functional or non-functional.
- Causes for software regression – common code changed incorrectly, incorrect version control, incorrect/incomplete bug fixes and code not updated according to the component changes.
- Regression testing should be done at all the levels – unit testing, integration testing and system testing.

21) What is integration testing?

- In this phase, individual modules are combined and tested in a group. Data transfer between the modules is tested thoroughly.
- Big bang integration – wait for all modules to be developed to begin testing.

22) When do we use integration testing?

- When the unit testing activity is done after each module or unit has been tested by the developers.
- Stubs and drivers come in handy when some part of the module is not yet tested, dummy code used for the module to work with the undeveloped module is called stub.
- The code created to integrate for the module to work is called a driver

23) When do we use smoke testing and sanity testing?

- Smoke testing is used to verify basic or critical existing functionality of an application before starting rigorous testing. Sanity testing is used to verify newly updated functionality or bug fixes. Smoke testing is performed by both Testers and developers whereas Sanity testing is done by the testers.

24) What is UAT?

- UAT is a phase of software development in which the software is tested in the "real world" by the intended audience. UAT can be done by in-house testing in which volunteers or paid test subjects use the software or, more typically for widely-distributed software, by making the test version available for downloading and free trial over the Web. The experiences of the early users are forwarded back to the developers who make final changes before releasing the software commercially.

25) What is alpha and beta testing?

- Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developer's site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.
- Beta testing is the final testing phase where companies release the software for few external user groups outside the company test teams or employees. This initial software version is called as beta version.

26) When do we use white box testing and block box testing?

- Black-box testing is the most commonly used type of testing in traditional organizations that have testers as a separate department, especially when they are not proficient in coding and have difficulties to understand the code. It provides external perspective of the software under test.
- White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) looks inside the software that is being tested and uses that knowledge as part of the testing process. If, for example, exception is thrown under certain conditions, test might want to reproduce those conditions. White-box testing requires internal knowledge of the system and programming skills. It provides internal perspective of the software under test.

27) What we will do if we don't have a time to test all stories?

- When we don't have time to test all the stories we will take the high priority bugs present in the application software and test them first.

28) What will you do if you come across any critical severity issue before release day?

•

29) When do we use automation testing?

- Automation testing is used in the following cases:

- Regression Testing: For re-testing pre-existing application functions that are being carried forward to new versions (usually the majority, unless app is brand new)
- Smoke Testing: For getting a quick high-level assessment on the quality of a build and making go / no-go decision on deeper testing
- Static and Repetitive Tests: For automating testing tasks that are repetitive and relatively unchanging from one test cycle to the next
- Data Driven Testing: For testing application functions where the same functions needs to be validated with lots of different inputs & large data sets (i.e. login, search)
- Load & Performance Testing: No manual alternative exists.

30) What tester will do in each phase of SDLC?

- The Role of a Tester in SDLC is as follows:
 - Requirement stage - PM, Tech. Lead, Test lead will review the document
 - Design Stage - PM or Test lead will review the document
 - Coding Stage - Test lead will prepare Test plan, Tester will prepare test cases and by giving both positive and negative input to the application.
 - Testing Stage - Tester will execute the testcase

31) Difference between load and performance testing?

- Performance testing would how fast the system would perform whereas Load Testing would be defined as testing how much load or volume can the system take to process it.
- Load testing is the process of subjecting a computer, peripheral, server, network or application to a work level approaching the limits of its specifications. Load testing is a part of a more general process known as performance testing.
- Performance testing is testing that is performed from one perspective: to determine how fast some aspect of a system performs under a particular workload. It can also serve to validate and verify other quality attributes of the system, such as scalability, reliability and resource usage.

32) Different types of non-functional testing?

- Load/Performance testing.
- Compatibility testing.
- Localization testing.
- Security testing.
- Reliability testing.
- Stress testing.
- Usability testing.
- Compliance testing.

33) What is test case?

- A test case is a set of conditions under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do.

34) what is test planning/test strategy document

- The Test Strategy document is a static document meaning that it is not updated too often. It sets the standards for testing processes and activities and other documents such as the Test Plan draws its contents from those standards set in the Test Strategy Document.
- Components of the Test Strategy document:
 - o Scope and Objectives
 - o Business issues
 - o Roles and responsibilities
 - o Communication and status reporting
 - o Test deliverables
 - o Industry standards to follow
 - o Test automation and tools
 - o Testing measurements
 - o Risks and mitigation
 - o Defect reporting and tracking
 - o Change and configuration management
 - o Training plan

35) what is TDD and BDD (cucumber framework)?

- Test Driven Development is also called test-driven design, is a method of software development in which unit testing is repeatedly done on source code. Write your tests watch it fails and then refactor it. The concept is we write these tests to check if the code we wrote works fine. After each test, refactoring is done and then the same or a similar test is performed again. The process is iterated as many times as necessary until each unit is functionally working as expected.
- BDD is similar in many ways to TDD except that the word “test” is replaced with the word “Behaviour”. It’s purpose is to help the the folks devising the system (i.e., the developer) identify appropriate tests to write—that is, tests that reflect the behavior desired by the stakeholders.

36) What is priority and severity in defect?

- Severity of a defect is related to how severe a bug is. Usually the severity is defined in terms of financial loss, damage to environment, company's reputation and loss of life. Priority of a defect is related to how quickly a bug should be fixed and deployed to live servers.

37) How to estimate test cases?

-

38) What is most challenge defect you came across?

-

39) How to deal the production defects?

- Reproduce the issue within the staging environment, create defect in defect tool under the production release version and developers will fix the issue whereas we (QA) test the issue on production version code (staging) and release the fix to production after we verify. We have to create a defect on current **sprint/release** so that developer will add this code to the current sprint/release.

40) What are the Test design techniques?

- Test Design is creating a set of inputs for given software that will provide a set of expected outputs. The idea is to ensure that the system is working good enough and it can be released with as few problems as possible for the average user. Broadly speaking there are two main categories of Test Design Techniques. They are:
 - Static Techniques
 - Dynamic Techniques

41) if we don't have time to test all test cases what we will do?

- When we don't have time to test all the test cases we only the test cases that are of high priority.

42) How we learn the functionality of system?

43) What are the tools to manage defects/stories?

44) Who will assign the work?

- The Team Leader

45) What are the types of test metrics we use normally?

- Process Metrics - Used to improve process efficiency of SDLC.
- Product Metrics - Deals with the quality of the software product
- Project Metrics - Used to measure the efficiency of the project team or any tools being used by the team members.
- Manual Test Metrics:
 - Base metrics - This is the raw data collected by Test Analyst during the test case development and execution (# of test cases executed, # of test cases)
 - Calculated metrics - Calculated metrics is derived from the data gathered in base metrics. Calculated metrics is usually tracked by the test manager for test reporting purpose (% Complete, % Test Coverage).

46) What is traceability matrix?

- A traceability matrix is a document, usually in the form of a table, used to assist in determining the completeness of a relationship by correlating any two baselined documents using a many-to-many relationship comparison. It is often used with high-level requirements (these often consist of marketing requirements) and detailed requirements of the product to the matching parts of high-level design, detailed design, test plan, and test cases.

47) What are typical environments we have in projects?

- Unit test Environment(development side)
- Integration test Environment(development side)
- System test Environment(testing side)
- Performance test Environment(testing side)
- User Acceptance test Environment(testing side/client)

48) What are different defect metrics and measurements we prepare?

49) What is staging environment?

- A stage or staging environment is an environment for testing that exactly resembles the production environment. In other words, it's a complete but independent copy of the production environment, including the database. Staging provides a true basis for QA testing because it precisely reproduces what is in production.

50) What is development environment?

- A Development environment is where you configure, customize, and use source control to build an image of the Waveset application to be promoted to another environment. You also write an upgrade procedure in this environment that you follow in each target environment.

51) What is QA environment?

- A QA environment is where you test your upgrade procedure against data, hardware, and software that closely simulate the Production environment and where you allow intended users to test the resulting Waveset application.

52) What is production environment?

- A Production environment is where the Waveset application is actually available for business use.