**Manual Testing:**

**It** is the process of **manually testing** software for defects. It requires a tester to play the role of an end user and use most of all features of the application to ensure correct behavior.

1. **Why testing is required?**

Ans: Testing is required for an effective performance of software application or product.

**2) What types of application we test?**

1. web applications,
2. desktop/windows applications
3. Mobile applications
4. ETL jobs
5. 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

**4)Different phases like:**

* **Requirement Analysis and planning:** 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.

Note: this high-level requirement will be written in BRD (Business Requirement document) by Business Analyst

* **Design:** in the define stage 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)

Seniors Tester write Test Planning document

* **Implementation/Development**: Developers write the code for the requirements, 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

**5) What is waterfall in SDLC?**

Ans: The waterfall model is a sequential design process, used in software development processes, in which progress is flowing steadily downwards through the phases of conception, initiation, analysis, design, construction, testing, production/implementation, and maintenance.

**6) What is the process in agile model?**

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds.

**7) What is scrum methodology?**

Scrum is a management and control process that cuts through complexity to focus on building software that meets business needs. Management and teams can get their hands around the requirements and technologies, never let go, and deliver working software, incrementally and empirically.

Scrum itself is a simple framework for effective team collaboration on complex software projects.

**8) What is daily standup meeting and what we discuss?**

A daily stand-up meeting is a short organizational meeting that is held each day. The meeting, generally limited to between five and fifteen minutes long, is sometimes referred to as a stand-up, a morning roll-call or a daily scrum.

The purpose of the meeting is for each team member to answer the following three questions:

1) What did you do yesterday?

2) What will you do today?

3) Are there any obstacles in your way?

Standing, rather than sitting, reinforces the idea that the meeting is intended to be short and discourages wasted time. The stand-up is not meant to be a place to solve problems, but rather to make the team aware of status. If discussion is needed, a longer meeting with appropriate parties can be arranged.

**9) What is user story/feature/sprint back log items and tasks in user story?**

Epic - Create a web site (T shirt size) (L)

Story - Create the Home page (1 point)

Task - Create a button on the home page that is red (1 hour)

Features are what a system is doing. User stories are just one way amongst others to capture features.

Epic - Is simply a story, but is considered so large that it needs to be broken down into multiple stories.

Story - Is essentially a requirement that is in a low enough detail that it can be estimated.

Task - Developers, testers may break the story down further into tasks to allow them to estimate, develop and test it. (If a story has a lot of tasks, it could be worth classing it as an epic and breaking it down into multiple stories.)

**10) What is sprint planning and spring retro?**

Sprint planning is a collaborative effort involving a ScrumMaster, who facilitates the meeting, a Product Owner, who clarifies the details of the product backlog items and their respective acceptance criteria, and the Entire Agile Team, who define the work and effort necessary to meet their sprint commitment.

Sprint retro: The sprint retrospective is usually the last thing done in a sprint. Many teams will do it immediately after the sprint review. The entire team, including both the ScrumMaster and the product owner should participate.

**11) what is burndown chart and velocity?**

Burndown Charts. The burndown is a chart that shows how quickly you and your team are burning through your customer's user stories.

The rate of progress of a Scrum Team is called "velocity". It expresses the amount of e.g. story points completed per iteration.

**12) What is product backlog item and sprint backlog items?**

The Product Backlog is an ordered list of everything that might be needed in the product and is the single source of requirements.

The Sprint Backlog is a forecast (…) what functionality will be in the next Increment and the work needed to deliver that functionality. The Sprint Backlog makes visible all the work (…) The Sprint Backlog is a plan (…)

**13) What is User Acceptance Criteria test cases?**

User acceptance is a type of testing performed by the Client to certify the system with respect to the requirements that was agreed upon. This testing happens in the final phase of testing before moving the software application to Market or Production environment.

14) **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. V - Model is an extension of the waterfall model and is based on association of a testing phase for each corresponding development stage.

15) **What is STLC?**

Software Testing Life Cycle (STLC) is the testing process which is executed in systematic and planned manner. In STLC process, different activities are carried out to improve the quality of the product.

16) **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.

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

defect lifecycle

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.

**18) Different types of testing:**

Software Testing Types:

Black box testing – Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.

White box testing – This testing is based on knowledge of the internal logic of an application’s code. Also known as Glass box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

Unit testing – Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. may require developing test driver modules or test harnesses.

Incremental integration testing – Bottom up approach for testing i.e continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately. done by programmers or by testers.

Integration testing – Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

Functional testing – This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black-box type testing geared to functional requirements of an application.

System testing – Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.

End-to-end testing – Like system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

Sanity testing – Testing to determine if a new software version is performing well enough to accept it for a major testing effort. If application is crashing for initial use, then system is not stable enough for further testing and build or application is assigned to fix.

Regression testing – Regression testing is nothing but full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.

Acceptance testing -Normally this type of testing is done to verify if system meets the customer specified requirements. User or customer do this testing to determine whether to accept application.

Load testing – It’s 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.

19) **When do we use regression testing?**

Regression Testing is required when there is a

* Change in requirements and code is modified per requirement
* New feature is added to the software
* Defect fixing
* Performance issue fix

20) **When do we use integration testing?**

In Integration Testing, individual software modules are integrated logically and tested as a group.

A typical software project consists of multiple software modules, coded by different programmers. Integration testing focuses on checking data communication amongst these modules. Hence it is also termed as 'I & T' (Integration and Testing), 'String Testing' and sometimes 'Thread Testing'.

* Interfaces of the software modules with the database could be erroneous
* External Hardware interfaces, if any, could be erroneous
* Inadequate exception handling could cause issues.

22**) When do we use smoke testing and sanity testing?**

Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine

For Example, a typical smoke test would be - Verify that the application launches successfully, Check that the GUI is responsive ... etc.

Sanity Testing is done to check the new functionality / bugs have been fixed.

The objective of the testing is to verify the "rationality" of the system in order to proceed with more rigorous testing

Alpha testing is a type of acceptance testing; performed to identify all possible issues/bugs before releasing the product to everyday users or public. The focus of this testing is to simulate real users by using blackbox and white box techniques. The aim is to carry out the tasks that a typical user might perform.

Beta Testing of a product is performed by "real users" of the software application in a "real environment" and can be considered as a form of external user acceptance testing.

23) **When do we use white box testing and block box testing?**

White Box Testing is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is also known as clear box testing, open box testing, logic driven testing or path driven testing or structural testing and glass box testing.

Black box testing is a software testing techniques in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of

Internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

23) **When do we use automation testing?**

* To break software more quickly or more effectively in a repeatable manner.
* In order to help me execute complex test cases in a timelier manner.
* To speed up testing.
* To help me set up or tear down test data (in which case my test case may or may not be automated).
* To improve programming skills

24) **What tester will do in each phase of SDLC?**

The Role of a Tester in SDLC

1. Tester prepares the Test cases, Test Scenarios from the SRS
2. Using the script the tester performs different kinds of testing (Regression, Function)
3. Tester Notes the results (pass/Fail)
4. If Result=Fail, then the scenario is raised in the Test director
5. Once it fixed by the developer the tester performs a regression testing

25) **Difference between load and performance testing?**

*Load test:* any test that involves to put a determined load on an application to verify how it behaves (i.e.: response time);

*Performance test:* it is a load test limited by the load defined by the specification of the application - the test is to verify or confirm that the application will work at the planned performance;

Stress test: there are many meanings for stress, but I tried to educate my customers to use the term stress only when you want to load the application beyond the specification definition, to understand the maximum capacity of application before it breaks. It helps for example to foresee any problems in a near future and prepare IT teams to be aware of the application capacity;

And there is Stability test: execute a load test during a long period to understand how stable is an application in a long run. It helps to determine, for example, if an application has memory leaks, which is very difficult to find in a normal load test.

26) **Different types of non-functional testing types?**

Examples of non-functional tests include:

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

27**) What is test case?**

A test case is a document, which has a set of test data, preconditions, expected results, and post conditions, developed for a particular test scenario in order to verify cooperation against a specific requirement.

28**) What is test plan/test strategy document?**

A test plan for software project can be defined as a document that defines the scope, objective, approach and emphasis on a software testing effort.

Ans: Test plan document contains different section like

       Types of testing:

       Exit and Entry criteria:

29) **What is TDD and BDD (cucumber framework)?**

It’s 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. TDD was introduced first by XP. I believe I have explained enough in simple terms.

Behavior-driven development combines the general techniques and principles of TDD with ideas from domain-driven design

DDD-Domain Driven Testing

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 folks devising the system (i.e., the developer) identify appropriate tests to write–that is, tests that reflect the behavior desired by the stakeholders. BDD is usually done in very English-like language helps the Domain experts to understand the implementation rather than exposing the code level tests. Its defined in a GWT format, GIVEN WHEN & THEN.

30) **What is priority and severity in defect?**

In software testing, defect severity can be defined as the degree of impact a defect has on the development or operation of a component application being tested.

Higher effect on the system functionality will lead to the assignment of higher severity to the bug. Quality Assurance engineer usually determines the severity level of defect

31) **Defect Priority states the order in which a defect should be fixed. Higher the priority the sooner the defect should be resolved**.?

Defects that leave the software system unusable are given higher priority over defects that cause a small functionality of the software to fail.

32) **How to estimate test cases?**

* 3-Point Software Testing Estimation Technique.
* Use – Case Point Method:
* Work Breakdown Structure.
* Wideband Delphi technique.
* Function Point/Testing Point Analysis.
* Percentage of development effort method.
* Percentage distribution.
* Best Guess.

33) **What is most challenge defect u came across?**

1. If a defect is found and later is not reproducible then steps to reproduce that defect is a challenge.

2. I was testing a mobile app A, there was multiple features which could be added as a bookmark. While going through one flow, it was working fine.

But if via one flow, the bookmark was added and removed for Y feature and the bookmark section was accessed from second flow, then automatically bookmark was added for Y feature. After spending one full day, it took 17 steps to find the exact issue.

3. There was another app B having a field that had maximum character limit of 13 characters, I entered 12 characters’ alphabets and 13th character as an emoji since emoji takes 2 characters the app crashed. After this was fixed, I was not allowed to enter an emoji then I copied emoji from other place and pasted in tht field again the app crashed.

4. There was one more app C, where s filed had limit of 30 characters. I copied 31 characters from notepad in mobile and pasted it in the field and it accepted.

Basically, I have many such scenarios so I do not know what exactly you are looking for.

34) **How to deal the production defects?**

Ans: normally end user will report this issue.

       we need to talk to them (end users) and reproduce the issue with in staging environment

  Create defect in defect tool under the production release version

  developers will fix the issue

  we (QA) test the issue on production version code (staging) and release the fix to production after we verify

 we must create a defect on current **sprint/release** so that developer will add this code to the current sprint/release

35) **Test design techniques**:

Broadly speaking there are two main categories of Test Design Techniques. They are:

* Static Techniques
* Dynamic Techniques

36) **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.

37) **What is development environment?**

A development environment is a collection of procedures and tools for developing, testing, and debugging an application or program. The development environment normally has three server tiers, called development, staging and production.

38) **What is QA environment?**

Development, Test, QA, and Production Environments. 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 Wave set application.

39) **What is production environment?**

A production environment is where the real-time staging of programs that run an organization are executed, and includes the personnel, processes, data, hardware, and software needed to perform day-to-day operations.