1. **Why testing is required?**

Ans: Software **testing** is very important because of the following reasons: Software **testing** is really required to point out the defects and errors that were made during the development phases. It's essential since it makes sure of the Customer's reliability and their satisfaction in the application.

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

Ans:

: 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?**

Ans: Software development life cycle (SDLC) is a process to develop the application

**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 requirements will be written in BRD (Business Requirement document) by Business Analyst

Define/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

**what is waterfal in SDLC?**

Ans:

**what is the process in agile model**

**what is scrum methodology**

**what is daily standup meeting and what we discuss**

**what is user story/feature/sprint back log items and tasks in user story**

**what is sprint planning and spring retro**

**what is burndown chart and velocity**

**what is product backlog item and sprint backlog items**

**what is user acceptance criteria test cases**

**what is v model?**

**V model:** it is enhanced version of waterfall model where each level of the development lifecycle is verified before moving to next level. In this testing starts at the very beginning. By testing we mean verification by means of reviews and inspections, static testing. Each level of the development life - cycle has a corresponding test plan. A test plan is developed to prepare for the testing of the products of that phase. Be developing the test plans, we can also define the expected results for testing of the products for that level as well as defining the entry and exit criteria for each level.

**what is STLC?**

**what is defect?**

**how to arise a defect and what we specify while logging defect?**

**defect lifecycle**

**Different types of testing?**

There are two types of testing:

- **Static testing:** Static testing is a technique used in the earlier phase of the development life cycle. The code error detection and execution of program is not concern in this type of testing. Also known as non-execution technique. The Verification of the product is performed in this testing technique like Code Reviews, Inspections, Walkthroughs are mostly done in this stage of testing.

- **Dynamic testing**: Dynamic Testing is concern with the execution of the software. This technique is used to test the dynamic behavior of the code. Most of the bugs are identified using this technique. These are the Validation activities. It uses different methodologies to perform testing like Unit Tests, Integration Tests, System Tests and Acceptance Testing, etc.

**when do we use regression testing?**

**when do we use integration testing?**

**when do we use smoke testing and sanity testing?**

**what is unit testing?**

**Use Case:** A use case is a description of the process which is performed by the end user for a particular task. Use case contains a sequence of step which is performed by the end user to complete a specific task or a step by step process that describe how the application and end user interact with each other. Use case is written by the user point of view.

**Use case Testing:** the use case testing uses this use case to evaluate the application. So that, the tester can examines all the functionalities of the application. Use case testing cover whole application,

**what is UAT?**

User Acceptance Testing (UAT) is performed by the end users on the applications before accepting the application.

**what is alpha and beta testing?**

**Alpha testing**: is performed by the IN-House developers. After alpha testing the software is handed for the Beta testing phase, for additional testing in an environment that is similar to the client environment.

**Beta testing**: is performed by the end user. So that they can make sure that the product is bug free or working as per the requirement. IN-house developers and software QA team perform alpha testing. The public, a few select prospective customers or the general public performs beta testing.

**Gamma Testing**: Gamma Testing is done when the software is ready for release with specified requirements. This testing is done directly by skipping all the in-house testing activities.

**when do we use white box testing and block box testing?**

**what we will do if we don’t have a time to test all stories?**

**what we will do if come across any severity issue before release day?**

**when do we use automation testing?**

**what tester will do in each phase of SDLC?**

**difference between load and performance testing?**

**different types of non-functional testing types?**

**what is test case?**

**what is test plan/test strategy document**

**Ans:** Test plan document contains different section like

Types of testing :

Exit and Entry criteria :

**what is TDD and BDD (cucumber framework)**

**what is priority and severity in defect?**

|  |
| --- |
| Priority: concern with application from the business point of view.  It answers: How quickly we need to fix the bug? Or how soon the bug should get fixed?  Severity: concern with functionality of application.  How much the bug is affecting the functionality of the application?  Ex.  1. High Priority and Low Severity:  If a company logo is not properly displayed on their website.  2. High Priority and High Severity:  Suppose you are doing online shopping and filled payment information, but after submitting the form, you get a message like "Order has been cancelled."  3. Low Priority and High Severity:  If we have a typical scenario in which the application get crashed, but that scenario exists rarely.  4. Low Priority and Low Severity:  There is a mistake like "You have registered success" instead of successfully, success is written. |
|  |

**how to estimate test cases?**

**what is most challenge defect u came across?**

**how to deal the production defects?**

**Ans:** normally end user will report this issue.

we need to talk to them and reproduce the issue with test logins

Create defect in defect tool under the production release version

developers will fix the issue

we (QA) test the issue on production version code and release the fix to proudction 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

**test design review steps**

**if we dont have time to test call test cases what we will do**

**how we learn the functionality of system?**

**what are the tools to manage defects/stories?**

**who will assign the work?**

**types of test metrics we use normally**

**what is traceability matrix?**

**what are typical environments we have in projects**

**what is development environment**

**what is QA environment**

**what is production environment**

**what are different defect metrics and measurements we prepare**

**what are weakness and strong points**

**What is staging environment**

**What is verification and validation?**

**Verification:** process of evaluating work-products of a development phase to determine whether they meet the specified requirements for that phase.

**Validation:** process of evaluating software during or at the end of the development process to determine whether it specified requirements.

**Difference between Verification and Validation:**

- Verification is Static Testing where as Validations is Dynamic Testing.

- Verification takes place before validation.

- Verification evaluates plans, document, requirements and specification, where as Validation evaluates product.

- Verification inputs are checklist, issues list, walkthroughs and inspection ,where as in Validation testing of actual product.

- Verification output is set of document, plans, specification and requirement documents where as in Validation actual product is output.