**Why testing is required?**

Ans: Software testing is required to check if the system is working as per design

**What types of application we test**

Ans: web applications  (we open these applications in browser , ex: gmail.com) desktop/windows applications (we run from our desktop, ex: notepad, word)

Mobile applications (we run from mobile devices, android, ios, windows phone)

Web services (SOAP/REST) (we use them as part of web applications / desktop applications/mobile applications)

ETL jobs, database validations : these jobs don’t have UI and they run in back ground to load the data (informatica jobs, ssis job etc)

Back end/batch programs/windows services

**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?**

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project.

The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap.

The different phases of waterfall model are as follows:

* Requirement Gathering and analysis
* System Design
* Implementation
* Integration and Testing
* Deployment of system
* Maintenance

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model".

**what is the process in agile model?**

In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

**what is scrum methodology**

Scrum is a subset of Agile. It is a lightweight process framework for agile development, and the most widely-used one.

A “process framework” is a particular set of practices that must be followed in order for a process to be consistent with the framework. (For example, the Scrum process framework requires the use of development cycles called Sprints, the XP framework requires pair programming, and so forth.)

“Lightweight” means that the overhead of the process is kept as small as possible, to maximize the amount of productive time available for getting useful work done.

A Scrum process is distinguished from other agile processes by specific concepts and practices, divided into the three categories of Roles, Artifacts, and Time Boxes. These and other terms used in Scrum are defined below. Scrum is most often used to manage complex software and product development, using iterative and incremental practices. Scrum significantly increases productivity and reduces time to benefits relative to classic “waterfall” processes. Scrum processes enable organizations to adjust smoothly to rapidly-changing requirements, and produce a product that meets evolving business goals. An agile Scrum process benefits the organization by helping it to

Increase the quality of the deliverables

Cope better with change (and expect the changes)

Provide better estimates while spending less time creating them

Be more in control of the project schedule and state

**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.

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

A user story is typically functionality that will be visible to end users. Developing it will usually involve a programmer and tester, perhaps a user interface designer or analyst, perhaps a database designer, or others.

A task, on the other hand, is typically something like code this, design that, create test data for such-and-such, automate that, and so on. These tend to be things done by one person.

**what is sprint planning and spring retro**

In Scrum, the sprint planning meeting is attended by the product owner, ScrumMaster and the entire Scrum team. Outside stakeholders may attend by invitation of the team, although this is rare in most companies.

During the sprint planning meeting, the product owner describes the highest priority features to the team. The team asks enough questions that they can turn a high-level user story of the product backlog into the more detailed tasks of the sprint backlog.

**Sprint retrospective:**

The sprint retrospective is a meeting facilitated by the ScrumMaster at which the team discusses the just-concluded sprint and determines what could be changed that might make the next sprint more productive. The sprint review looks at what the team is building, whereas the retrospective looks at how they are building it.

The retrospective includes three main questions/points for discussion:

What went well during the sprint cycle?

What went wrong during the sprint cycle?

What could we do differently to improve?

The sprint retrospective is an important mechanism that allows a team to continuously evolve and improve throughout the life of a project.

It is important that everyone, including the team, product owner, and ScrumMaster, get a chance to air their opinions in an open, honest, yet constructive atmosphere. It often also helps management to get feedback from the team about the work and progress of project.

**what is burndown chart and velocity**

Burndown charts show work remaining over time. Work remaining is the Y axis and time is the X axis. The work remaining should jig up and down and eventually trend downward.

The Scrum books define a sprint burndown chart as a place to see daily progress, and a product burndown chart as where to show monthly (per sprint) progress.

**Velocity**

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.

Once established, velocity can be used to plan projects and forecast release and product completion dates.

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

**Product Backlog Item**

In Scrum, a product backlog item ("PBI", "backlog item", or "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**

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.

**what is user acceptance criteria test cases**

User acceptance testing (UAT) is the last phase of the software testing process. During UAT, actual software users test the software to make sure it can handle required tasks in real-world scenarios, according to specifications. UAT is one of the final and critical software project procedures that must occur before newly developed software is rolled out to the market.

**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.

Under V-Model, the corresponding testing phase of the development phase is planned in parallel. So there are Verification phases on one side of the .V. and Validation phases on the other side. Coding phase joins the two sides of the V-Model

**what is STLC?**

Software Testing Life Cycle (STLC) is defined as a sequence of activities conducted to perform Software Testing.

It consists of series of activities carried out methodologically to help certify your software product.

Diagram - Different stages in Software Test Life Cycle

[](http://cdn.guru99.com/images/stories/software-test-life-cycle.jpg)

what is defect?

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

defect lifecycle

**Different types of testing:**

**What is unit testing?**

Unit testing of software applications is done during the development (coding) of an application.

The objective of unit testing is to isolate a section of code and verify its correctness. In procedural programming a unit may be an individual function or procedure

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. Unit testing is usually performed by the developer.

**when do we use regression testing?**

Regression testing is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features.

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

This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that old code still works once the new code changes are done.

Regression Testing is required when there is a

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

**What is integration testing?**

Ans: 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'.

**when do we use integration testing?**

Integration Testing is performed after Unit Testing and before System Testing.

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

Smoke Testing is a kind of Software Testing performed after software build to ascertain that the critical functionalities of the program is working fine. It is executed "before" any detailed functional or regression tests are executed on the software build. The purpose is to reject a badly broken application, so that the QA team does not waste time installing and testing the software application.

Sanity testing is a kind of Software Testing performed after receiving a software build, with minor changes in code, or functionality, to ascertain that the bugs have been fixed and no further issues are introduced due to these changes. The goal is to determine that the proposed functionality works roughly as expected. If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.

**what is UAT?**

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.

**what is alpha and beta 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 whitebox techniques. The aim is to carry out the tasks that a typical user might perform. Alpha testing is carried out in a lab environment and usually the testers are internal employees of the organization. To put it as simple as possible, this kind of testing is called alpha only because it is done early on, near the end of the development of the software, and before beta testing.

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.

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

A software testing technique whereby explicit knowledge of the internal workings of the item being tested are used to select the test data. Unlike black box testing, white box testing uses specific knowledge of programming code to examine outputs. The test is accurate only if the tester knows what the program is supposed to do.

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance.

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

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

**when do we use automation testing?**

Test engineers strive to catch them before the product is released but they always creep in and they often reappear, even with the best manual **testing processes**. Test Automation software is the best way to increase the effectiveness, efficiency and coverage of your software testing.

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

**In**phases of **SDLC**:

* Requirements gathering and Analysis

In this phase of SDLC, suitable necessities of system are accumulated. All adjacent methods should be in focus. All types of estimation and examination of user needs are done in this phase.

* System Design

In the second phase a basic system planning is done. After collecting the all statistics and data, a system design is done.

* Implementation

In the next phase implementation of project is done. Respect to the system design, correct development is made to expand that design. According to the project programming language will be chosen.

* System Testing

After the implementation phase, system testing phase take place to recognize the result of application. Testing is done to recognize the original result and the predictable result.

* Operation Maintenance

It is the ultimate phase of SDLC, where the application which is implemented is spread to users who are answerable for conserving and using it for appropriate actions. The implemented application should be available for any adjustment to do in coding.

**difference between load and performance testing?**

Performance testing is the testing, which is performed, to ascertain how the components of a system are performing, given a particular situation. Resource usage, scalability and reliability of the product are also validated under this testing. This testing is the subset of performance engineering, which is focused on addressing performance issues in the design and architecture of software product.

Load testing is meant to test the system by constantly and steadily increasing the load on the system till the time it reaches the threshold limit. It is the simplest form of testing which employs the use of automation tools such as LoadRunner or any other good tools, which are available. Load testing is also famous by the names like **volume testing** and **endurance testing**.

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

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

**what is test case?**

A **test case**, in software engineering, 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.

**what is test planning/test strategy document**

Test plan document contains different section like

Types of testing :

Exit and Entry criteria :

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

**Cucumber** is a testing **framework** which supports Behavior Driven Development (**BDD**). It lets us define application behavior in plain meaningful English text using a simple grammar defined by a language called Gherkin.

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

**Severity** is defined as the degree of impact a **defect** has on the development or operation of a component application being tested

Priority is defined as 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.

**how to estimate test cases?**

1. 3-Point Software Testing Estimation Technique.
2. Use – Case Point Method:
3. Work Breakdown Structure.
4. Wideband Delphi technique.
5. Function Point/Testing Point Analysis.
6. Percentage of development effort method.
7. Percentage distribution.
8. Best Guess.

**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 (end users) and reproduce the issue with in staging environment

Create defect in defect tool under the production release version

developers will fix the issuewe (QA) test the issue on production version code (stageing) 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

**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?**

* Stryka: Details: Stryka is a cutting-edge enterprise test management tool, built from the ground up using the latest web and mobile technologies.
* Bugzilla:
* Lean Testing.
* JIRA:
* Mantis:
* Trac:
* Redmine:

**who will assign the work?**

QA lead

**types of test metrics we use normally?**

**Process Metrics:** It can be used to improve the process efficiency of the SDLC ( Software Development Life Cycle)

**Product Metrics:** It deals with the quality of the software product

**Project Metrics:** It can be used to measure the efficiency of a project team or any tools being used by the team members

Manual test metrics is classified into two classes

**Base Metrics**

**Calculated Metrics**

Base metrics is the raw data collected by Test Analyst during the test case development and execution (**# of test cases executed, # of test cases**). While, 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**).

**What is requirement traceability Matrix?**

The Requirements Traceability Matrix (RTM) is a document that links requirements throughout the validation process. The purpose of the Requirements Traceability Matrix is to ensure that all requirements defined for a system are tested in the test protocols.

**What is development Environment?**

A development environment is a collection of procedures and tools for developing, testing and debugging an application or program.

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

**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.

**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.

**how to deal the production defects?**

1. we should check whether the bug falls within our testing scope or not.
2. Find the test set where we have covered the scenario under which the bug arisen
3. Check out the configuration related to the module where the production bug is found. In most of the cases it will be due to configuration changes
4. Check out for the change in SIT and PROD operating systems & dbms versions including any upgrades or patches installed on systems etc
5. Prepare a RCA report which should contain all the factors mentioned in above steps
6. Follow up with support /Dev team to check the status on the ticket which was opened for the production bug.