1. Why testing is required?

Ans: testing is required to point out the defects and errors that were made during development of the software.

2) What types of application we test

Ans:

* web applications
* desktop/windows applications
* Mobile applications
* ETL jobs
* 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

2) What is waterfall in SDLC?

Ans: water fall model is also known as linear sequential life cycle model. It is very successful approach for small projects. In this model testing starts at the end when development work is completed.

Phases in waterfall model:

* **Requirements**: This is the first phase of development where all the requirements gathered and documented.
* **Analysis**: In this phase we analyze all the gathered requirements whether the requirements are valid or invalid.
* **Design**: In this phase all the system design is analyzed and specified like hardware, system configuration and architecture or the system.
* **Implementation**: In this phase all the development works are performed and development components or units handed over to testing team.
* **Testing**: Once the development completed, testing phase starts and in this phase we test the each unit or component and make sure the developed components are working as expected. All the testing activities are performed in this phase.
* **Deployment**: Once testing is completed and make sure there is no bug or defect or any kind of issue, then project is deployed to production. Once product is deployed to production the end users start using the product.
* **Maintenance**: We always keep eye on the product and provide all the necessary bug or issue fixes if occurs in production or reported by end users. Also time to time we keep updated the product with new updates or patches if developed or available.

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

Ans: Agile as the name specifies something that to be done faster or quickly. On testing and development point of view Agile is completely a new methodology to speed up your testing and development activity. As the requirements change frequently and we need to complete the testing or development for the change request as quick as possible. In this situation agile methodology comes forward. Agile methodology should be performed from the starting of the project and should carry out till the project go live. Even some times this methodology carry forward for some more time until the project becomes stable.

Sprint

Scrum meeting

Backlog

Q: what is scrum methodology?

Ans: Scrum is a kind of short time meeting done every day and we discuss about every day work. It’s not like a general meeting but we fully focus on the whole day development and testing work. Also in this meeting we discuss about the daily work and priority modules. We plan for –

**Total time duration:** In scrum meeting we decide the total time duration for each sprint to test and develop. On daily basis we track the progress of each task assigned.

**Identifying the risk:** Another plan that we make in scrum meeting is to identify the risks. Risks like which tasks are more complicated or which task could have probability to be failed or which task could have more bugs etc.

**Cost**: This is another priority that we discuss in Scrum meeting that how much cost will be to develop or test any task. If development time or testing time will be exceeded will affect the expected or allocated cost. So we discuss how to balance time and cost.

**Resources**: How many resources required to test or develop each task is also an important factor that are discussed in scrum meeting. There are some other topics as well like potential of the work, backlogs etc. are also a part of discussion that should be happen in scrum meeting. In Scrum meeting generally we used to have one scrum master who decides how smoothly project development and testing are going on.

Q: what is daily standup meeting and what we discuss?

Ans: some software development methodologies envision daily team meetings to team members. The daily commitments allows participants to know about potential challenges as well as to coordinate efforts to resolve difficult and/or time consuming issues. The standup has particular value in agile software development processes.

Q: what is user story/feature/sprint back log items and tasks in user story?

Ans: **user story** is a tool used in agile software development to capture a description of a software feature from an end-user perspective. The user story describes the type of a user, what they want and why. A user story helps to create a simplified description of a requirement.

A user story may include several **Features**. For instance as a user I want to search for a job so I can find various features such as simple search, faceted navigation, sort, filter and advanced search feature might include back end development of search index, front end development of search box, an API to find the most relevant search results.

During the sprint planning meeting the team selects no. of product **backlog items** usually in form of user stories and identifies tasks necessary to complete.

Q: what is sprint planning and spring retro?

Ans: Sprint is nothing but it’s a part of scrum meeting where we decide which module should be developed or tested first. The piece of task is called sprint. **Sprint planning**, is we discuss on daily basis about the progress of particular sprint and sprint backlog in sprint meeting.

The **Sprint Retrospective** is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint. The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning.

Q: what is burn down chart and velocity?

Ans: The **burn down** is a chart that shows how quickly you and your team are burning through your customer's [user stories](http://www.agilenutshell.com/user_stories). It shows the total effort against the amount of work we deliver each iteration.

The rate of progress of a Scrum Team is called "**velocity**". It expresses the amount of e.g. story points completed per iteration. An import rule for calculating the velocity is that only stories that are completed at the end of the iteration are counted. Counting partially finished work (e.g. coding only - test missing) is strictly forbidden.

Q: what is product backlog item and sprint backlog items?

Ans: The **sprint backlog** is a list of tasks identified by the Scrum team to be completed during the Scrum **sprint**.

During the **sprint** planning meeting, the team selects some number of **product backlog items**, usually in the form of user stories, and identifies the tasks necessary to complete each user story.

Q: what is user acceptance criteria test cases?

Ans:

Q: what is v model?

Ans: V Model or Verification and Validation Model. In V Model there are some steps or sequences specified which should be followed during performing test approach. Once one step completes we should move to the next step. Test execution sequences are followed in V shape. Verification phase should be carried out from SDLC where validation phase should be carried out from STLC (Software Testing Life Cycle)

**Steps in V Model**

Basically there are 4 steps involved in STLC while performing V Model testing strategy.

* Unit Testing.
* Integration Testing.
* System Testing.
* Acceptance Testing.

Q: what is STLC?

Ans: 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.

* Requirement Analysis
* Test Planning
* Test Case Development
* Environment Setup
* Test Execution
* Test Cycle Closure

Q: what is defect?

Ans: A [**defect**](http://istqbexamcertification.com/what-is-defect-or-bugs-or-faults-in-software-testing/) 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.

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

Ans:

Q: defect lifecycle?

Ans: 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**](http://istqbexamcertification.com/what-is-a-defect-life-cycle/) is related to the bug found during testing.

**Different types of testing:**

Q: What is unit testing?

Ans: A unit is the smallest testable part of an application like functions, classes, procedures, interfaces. Unit testing is a method by which individual units of source code are tested to determine if they are fit for use. Unit tests are basically written and executed by software developers to make sure that code meets its design and requirements and behaves as expected.

Q: when do we use regression testing?

Ans: Regression testing is used when:

* Any new feature is added
* Any enhancement is done
* Any bug is fixed
* Any performance related issue is fixed

Q: What is integration testing?

Ans: Integration testing tests integration or interfaces between components, interactions to different parts of the system such as an operating system, file system and hardware or interfaces between systems.

Integration testing follows two approach known as ‘Top Down’ approach and ‘Bottom Up’ approach

Q: when do we use integration testing?

Ans:

Q: What is unit testing?

Q: what is UAT?

Ans: UAT means user acceptance testing. Testing whether the application is ready for deployment or not. Acceptance testing is classified into 2 phases.

* α-testing.
* β-testing.

Q: what is alpha and beta testing?

Ans:

α-testing: testing on the final build and tester’s environment in the presence of client /customer.

β-testing: testing the application by the client in the client environment in the presence of maintenance team/release team.

Q: when do we use white box testing and block box testing?

Ans: white box testing is used to verify internal structure of systems component.

Black box testing is used to verify input methods and output methods of system.

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

Ans:

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

Ans:

Q: when do we use automation testing?

Ans:

Q: what tester will do in each phase of SDLC?

Ans:

Q: difference between load and performance testing?

Ans:

Q: different types of non-functional testing types?

Ans: Nonfunctional testing types:

* load testing
* performance testing
* compatibility testing
* localization testing
* security testing
* reliability testing
* stress testing
* usability testing
* compliance testing

Q: what is test case?

Ans: A test case is a set of conditions or variables 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.

Q: what is test plan/test strategy document?

Ans: Test plan document contains different section like

       Types of testing

       Exit and Entry criteria

Q: what is TDD and BDD (cucumber framework)?

Ans:

TDD:

Test-driven development starts with developing test for each one of the features. The test might fail as the tests are developed even before the development. Development team then develops and refactors the code to pass the test.

Test-driven development is related to the test-first programming evolved as part of extreme programming concepts.

**Test-Driven Development Process:**

* Add a Test
* Run all tests and see if the new one fails
* Write some code
* Run tests and Refactor code
* Repeat

BDD:

Behavior Driven testing is an extension of TDD. Like in TDD in BDD also we write tests first and the add application code. The major difference that we get to see here are

* Tests are written in plain descriptive.
* Tests are explained as behavior of application and are more user focused.
* Using examples to clarify requirements.

**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**. Cucumber itself is written in **Ruby**, but it can be used to “test” code written in Ruby or other languages including but not limited to Java, C# and Python.

Q: what is priority and severity in defect?

Ans:

Q: how to estimate test cases?

Ans:

Q: what is most challenge defect u came across?

Ans:

Q: 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 (stageing) 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

Q: test design techniques:

Ans:

Q: if we don’t have time to test call test cases what we will do

Ans:

Q: how we learn the functionality of system?

Ans:

Q: what are the tools to manage defects/stories?

Ans:

Q: who will assign the work?

Ans:

Q: types of test metrics we use normally?

Ans: Types of test metrics:

Project level:

* requirements coverage
* defect distribution
* defect open and close rates
* execution trend

Department level:

* mean time to detect and mean time to repair
* defect removal efficiency
* overall testing trends
* defect trends
* burn down chart

Company level:

* customer reported issue percentage
* defect severity indexy
* mttr and mttd
* mean time between failure

Q: what is traceability matrix?

## Ans: A traceability matrix is primarily used in software development projects to trace, identify and verify that a specific functionality or component is being developed. Typically, a traceability matrix is a worksheet type document consisting of a table(s). Two different sets of values are compared against each other by placing an identifier for one set in the top row, and the other set on the left column.

Q: what are typical environments we have in projects?

Ans:

Q: what are different defect metrics and measurements we prepare?

Ans:

* **Project level:**

**Defect distribution** = \*status\*phase

**Defect open and close rates**= \*100

* **Department level:**

**Defect removal efficiency** = \* 100

**Defect trends**: this is a combination of several metrics including defect distribution, defect open and close rates DRE.

Q: What is staging environment?

Ans: The staging tier is an environment that is as identical to the production environment as possible. The purpose of the Staging environment is to simulate as much of the Production environment as possible. The Staging environment can also double as a Demonstration/Training environment.

Q: what is development environment?

Ans: 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.

Q: what is QA environment?

Ans: 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.

Q: what is production environment?

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