1. Why testing is required?

Ans: **Testing is processes, in which the defects are identified, isolated, subjected for rectification & ensure that the product is defect free. In order to produce the quality product in the end and hence customer satisfaction.**

What types of application we test

Ans:Web Applications,Mobile Applications

1)what is SDLC and different phases in SDLC?

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

**SDLC contains six phases they are**

1. **Initial phase or requirements phase.**
2. **Analysis phase**
3. **Design phase**
4. **Coding Phase**
5. **Testing Phase**

2) what is waterfal in SDLC?

* Ans: **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
* **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
* **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
* **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* **Deployment of system:** Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market.
* **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

what is the process in **agile** model

Ans: **An agile development cycle is different. Instead, the initial planning and analysis is kept to a very high level, just enough to outline the scope of the development project. Then the team go through a series of iterations, analysing, designing, developing and testing each feature in turn within the iterations**.

what is scrum methodology

Ans: **Product Backlog Creation. ...**

**Sprint Planning and Sprint Backlog Creation. ...**

**Working on the Sprint. ...**

**Testing and Product Demonstration. ...**

**Retrospective and Next Sprint Planning.**

what is daily standup meeting and what we discuss

Ans**: 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.** **The stand-up is not meant to be a place to solve problems, but rather to make the team aware of current status.**

**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 impediments in your way?**

what is user story and tasks in user story

Ans: **A 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 user, what they want and why. A user story helps to create a simplified description of a requirement.**

what is sprint planning and spring retro

Ans: **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 retrospective is a meeting facilitated by the ScrumMaster at which the team discusses the just-concluded sprintand determines what could be changed that might make the next sprint more productive.**

**what is burndown chart and velocity**

what is product backlog item and sprint backlog items

Ans: **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. See also backlog effort estimation unit**

**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.**

what is user acceptance criteria test cases.

Ans:

what is v model?

**Ans: The various phases of the V-model are as follows:**

**Requirements** like BRS and SRS begin the life cycle model just like the waterfall model. But, in this model before development is started, a [system test](http://istqbexamcertification.com/what-is-system-testing/) plan is created.  The test plan focuses on meeting the functionality specified in the requirements gathering.

**The high-level design (HLD)** phase focuses on system architecture and design. It provide overview of solution, platform, system, product and service/process. An [integration test](http://istqbexamcertification.com/what-is-integration-testing/) plan is created in this phase as well in order to test the pieces of the software systems ability to work together.

**The low-level design** **(LLD)** phase is where the actual software components are designed. It defines the actual logic for each and every component of the system. Class diagram with all the methods and relation between classes comes under LLD. Component tests are created in this phase as well.

**The implementation** phase is, again, where all coding takes place. Once coding is complete, the path of execution continues up the right side of the V where the test plans developed earlier are now put to use.

**Coding:** This is at the bottom of the V-Shape model. Module design is converted into code by developers.

what is STLC?

Ans: **Software Testing Life Cycle refers to a testing process which has specific steps to be executed in a definite sequence to ensure that the quality goals have been met. In STLC process, each activity is carried out in a planned and systematic way. Each phase has different goals and deliverables. Different organizations have different phases in STLC**

Below are the phases of STLC:

1. Requirements phase
2. Planning Phase
3. Analysis phase
4. Design Phase
5. Implementation Phase
6. Execution Phase
7. Conclusion Phase
8. Closure Phase

what is defect?

Ans: **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**.

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

Ans: **Can raise this by clicking on the Defect button. Defect logging, a process of finding defects in the application under test or product by testing or recording feedback from customers and making new versions of the product that fix the defects or the clients feedback.**

defect lifecycle

Ans: **Bug or defect life cycle includes following steps or status:**

1. **New:** When a defect is logged and posted for the first time. It’s state is given as new.
2. **Assigned:** After the tester has posted the bug, the lead of the tester approves that the bug is genuine and he assigns the bug to corresponding developer and the developer team. It’s state given as assigned.
3. **Open:** At  this state the developer has started analyzing and working on the defect fix.
4. **Fixed:** When developer makes necessary code changes and verifies the changes then he/she can make bug status as ‘Fixed’ and the bug is passed to testing team.
5. **Pending retest:**  After fixing the defect the developer has given that particular code for retesting to the tester. Here the testing is pending on the testers end. Hence its status is pending retest.
6. **Retest:**  At this stage the tester do the retesting of the changed code which developer has given to him to check whether the defect got fixed or not.
7. **Verified:** The tester tests the bug again after it got fixed by the developer. If the bug is not present in the software, he approves that the bug is fixed and changes the status to “verified”.
8. **Reopen:** If the bug still exists even after the bug is fixed by the developer, the tester changes the status to “reopened”. The bug goes through the life cycle once again.
9. **Closed:** Once the bug is fixed, it is tested by the tester. If the tester feels that the bug no longer exists in the software, he changes the status of the bug to “closed”. This state means that the bug is fixed, tested and approved.
10. **Duplicate:** If the bug is repeated twice or the two bugs mention the same concept of the bug, then one bug status is changed to “duplicate**“.**
11. **Rejected:** If the developer feels that the bug is not genuine, he rejects the bug. Then the state of the bug is changed to “rejected”.
12. **Deferred:** The bug, changed to deferred state means the bug is expected to be fixed in next releases. The reasons for changing the bug to this state have many factors. Some of them are **priority**of the bug may be low, lack of time for the release or the bug may not have major effect on the software.
13. **Not a bug:**  The state given as “Not a bug” if there is no change in the functionality of the application. For an example: If customer asks for some change in the look and field of the application like change of colour of some text then it is not a bug but just some change in the looks of the  application.

Different types of testing?

Ans: Different types of tests

**GUI testing**,

Functional testing,

**Regression testing**,

**Smoke testing**,

**load testing**,

**stress testing**,

security testing,

**stress testing**,

ad-hoc testing etc., are carried out to complete **system testing**.

when do we use regression testing?

Ans: 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

when do we use integration testing?

* Ans: A Module in general is designed by an individual software developer whose understanding and programming logic may differ from other programmers. Integration testing becomes necessary to verify the software modules work in unity
* At the time of module development, there are wide chances of change in requirements by the clients. These new requirements may not be unit tested and hence integration testing becomes necessary.
* 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.

when do we use smoke testing and sanity testing?

Ans: 1) To ensure the major part of the application are working as expected before the show down(furtur testing)

2) To verify the environment is up and running, which would cause chaos

3) To identify potential "Show Stoppers" before starting full fldge testing thus avoiding turn-around time

4) To ensure fixes for "Show Stoppers" don't re-iterate

5) To gain confidence about the risk in the project

6) To uncover latent defects in the application

7) To plan/prepare for Re-testing and Regression testing before hands

8) To ensure changes incorporated/replaced onto the application are as behaving as expected

Sanity performed on each and every build given for testing to signal back to dev team that build has passed smoke/sanity and we can now proceed further.

what is unit testing?

Ans: **Unit Testing** is a [level of software testing](http://softwaretestingfundamentals.com/software-testing-levels/) where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output. In procedural programming a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class.

what is UAT?

Ans:***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 alpha and beta testing?

Ans: **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.**

**Beta version of the software is released to a limited number of end-users of the product to obtain feedback on the product quality. Beta testing reduces product failure risks and provides increased quality of the product through customer validation.**

**It is the final test before shipping a product to the customers. Direct feedback from customers is a major advantage of Beta Testing. This testing helps to tests the product in real time environment.**

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

* Ans**: Initially requirements and specifications of the system are examined.**
* **Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly . Also some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.**
* **Tester determines expected outputs for all those inputs.**
* **Software tester constructs test cases with the selected inputs.**
* **The test cases are executed.**
* **Software tester compares the actual outputs with the expected outputs.**
* **Defects if any are fixed and re-tested. Initially requirements and specifications of the system are examined.**
* **Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly . Also some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.**
* **Tester determines expected outputs for all those inputs.**
* **Software tester constructs test cases with the selected inputs.**
* **The test cases are executed.**
* **Software tester compares the actual outputs with the expected outputs.**
* **Defects if any are fixed and re-tested.**

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?

Ans: **will try to find out the cause for it. And with discussion with developers also, we will try to fix that out. And will work extra hours to get it fixed, and do testing again.  
And if the defect will take time to be resolved, then we will talk to Project Manager, that we can't send this release, as it has this defect, and we are working on it.**

**If it is having High Severity and low frequency of occurance then we can make the bug as known issue and move it to maintainance phase.**

when do we use automation testing?

Ans: **Regression Testing: For re-testing preexisting 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 & 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 viable manual alternative exists**

what tester will do in each phase of SDLC?

Ans **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 its fixed by the developer the tester performs a regression testing**

difference between load and performance testing?

Ans**: Performance Testing measures the response time of an application with an expected number of users. The aim of this is to get a baseline and an indication of how an application behaves under normal conditions.**

**Load Testing is measuring the response time when the application is subjected to more than usual number of users.  
The response time will increase, i.e. the application will be slower under heavy load, but the aim of load testing is to see whether the application can sustain the increased load on the server or will it crash and kill the servers.**

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?

Ans: **A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly.**

**The process of developing test cases can also help find problems in the requirements or design of an application.**

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)

**Protractor just makes it easier to test Angular application. Its a framework for testing Angular applications.**

**It has Automatic Waiting for elements and you can access specific Angular elements by.model and by.binding. Also it implements the Page Objects in the framework making tests more maintainable if used correctly. Probably there are some more enhancements not listed on the front-page of Protractor.**

what is priority and severity in defect?

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?

Ans:

what are the tools to manage defects/stories?

Ans:HP,ALM

who will assign the work?

Ans:teamlead

types of test metrics we use normally

**Ans: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**

what is traceability matrix?

Ans: **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**.

what are typical environments we have in projects

Ans: **Identify activities**

**• Estimate times and resources**

**• Identify relationships and dependencies**

**• Identify schedule restraints**

what is development environment

Ans: **In computer program and software product development, the development environment is the set of processes and programming tools used to create the program or software product. The term may sometimes also imply the physical environment.**

what is QA environment

Ans: **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 Waveset application.**

what is production environment

Ans: **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.**

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

what are weakness and strong points

Ans:

Strengths:  
- Positive thinking.  
- Honest.  
- Friendly nature.  
  
Weakness:  
  
- I can't say no when some one asks favor.

What is staging environment

Ans: **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. Stagingprovides a true basis for QA testing because it precisely reproduces what is in production.**