1)what is waterfall in SDLC?

-The sequential phases in Waterfall model are:

* **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

2)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. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.

At the end of the iteration a working product is displayed to the customer and important stakeholders.

What is Scrum Methodology??

Scrum is an [Agile framework](https://www.scrumalliance.org/why-scrum/core-scrum-values-roles) for completing complex projects. Scrum originally was formalized for software development projects, but it works well for any complex, innovative scope of work. The possibilities are endless. The Scrum framework is deceptively simple.

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 impediments in your way?

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

The sprint backlog is a list of tasks identified by the Scrum team to be completed during the [Scrum](https://www.mountaingoatsoftware.com/agile/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. Most teams also estimate how many hours each task will take someone on the team to complete.

It's critical that the team selects the items and size of the sprint backlog. Because they are the people committing to completing the tasks, they must be the people to choose what they are committing to during the Scrum sprint.

The sprint backlog is commonly maintained as a spreadsheet, but it is also possible to use your defect tracking system or any of a number of software products designed specifically for Scrum or agile. An example of a sprint backlog in a spreadsheet looks like this:



During the Scrum sprint, team members are expected to update the sprint backlog as new information is available, but minimally once per day. Many teams will do this during the daily scrum. Once each day, the estimated work remaining in the sprint is calculated and graphed by the ScrumMaster, resulting in a sprint burndown chart like this one.

The team does its best to pull the right amount of work into the Scrum sprint, but sometimes too much or too little work is pulled in during planning. In this case, the team needs to add or remove tasks.

Sprint Planning meeting??

In [Scrum](http://www.open.collab.net/nonav/community/swp/training/IntroToScrum/Intro_to_scrum.htm), every iteration begins with a [sprint planning meeting](http://www.open.collab.net/nonav/community/swp/training/SprintPlanningMeeting/SprintPlanningMeeting.htm). At this meeting, the [Product Owner](http://scrumreferencecard.com/scrum-reference-card/#Scrum-Roles) and the team negotiate which stories a team will tackle that sprint. This meeting is a time-boxed conversation between the Product Owner and the team. It’s up to the Product Owner to decide which stories are of the highest priority to the release and which will generate the highest business value, but the team has the power to push back and voice concerns or [impediments](http://scrummethodology.com/scrum-impediments/).

When the team agrees to tackle the work, the Product Owner adds the corresponding stories into the [sprint backlog](http://scrumreferencecard.com/scrum-reference-card/#Scrum-Artifacts). We usually recommend this be physically represented by moving a Post-It note or index card with a story written on it from the backlog into the [sprint backlog](http://scrumreferencecard.com/scrum-reference-card/#Scrum-Artifacts).

At this point, the Product Owner may choose to leave while the team decomposes the forecasted backlog items into tasks. This meeting is sometimes called Sprint Planning Part 2.

what is burndown chart and velocity??

Its purpose is to enable that the project is on the track to deliver the expected solution within the desired schedule. Simple **Burndown Chart**. The rate of progress of a Scrum Team is called "**velocity**". It expresses the amount of e.g. story points completed per iteration.

what is 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. See also **backlog** effort estimation unit.

Sprint Backlog Item??

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??

Bookmark

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. This means that for every single phase in the development cycle there is a directly associated testing phase. This is a highly disciplined model and next phase starts only after completion of the previous phase.

rification Phases

Following are the Verification phases in V-Model:

* **Business Requirement Analysis:** This is the first phase in the development cycle where the product requirements are understood from the customer perspective. This phase involves detailed communication with the customer to understand his expectations and exact requirement. This is a very important activity and need to be managed well, as most of the customers are not sure about what exactly they need. The acceptance test design planning is done at this stage as business requirements can be used as an input for acceptance testing.
* **System Design:** Once you have the clear and detailed product requirements, it.s time to design the complete system. System design would comprise of understanding and detailing the complete hardware and communication setup for the product under development. System test plan is developed based on the system design. Doing this at an earlier stage leaves more time for actual test execution later.
* **Architectural Design:** Architectural specifications are understood and designed in this phase. Usually more than one technical approach is proposed and based on the technical and financial feasibility the final decision is taken. System design is broken down further into modules taking up different functionality. This is also referred to as High Level Design (HLD).

The data transfer and communication between the internal modules and with the outside world (other systems) is clearly understood and defined in this stage. With this information, integration tests can be designed and documented during this stage.

* **Module Design:**In this phase the detailed internal design for all the system modules is specified, referred to as Low Level Design (LLD). It is important that the design is compatible with the other modules in the system architecture and the other external systems. Unit tests are an essential part of any development process and helps eliminate the maximum faults and errors at a very early stage. Unit tests can be designed at this stage based on the internal module designs.

Coding Phase

The actual coding of the system modules designed in the design phase is taken up in the Coding phase. The best suitable programming language is decided based on the system and architectural requirements. The coding is performed based on the coding guidelines and standards. The code goes through numerous code reviews and is optimized for best performance before the final build is checked into the repository.

Validation Phases

Following are the Validation phases in V-Model:

* **Unit Testing:** Unit tests designed in the module design phase are executed on the code during this validation phase. Unit testing is the testing at code level and helps eliminate bugs at an early stage, though all defects cannot be uncovered by unit testing.
* **Integration Testing:** Integration testing is associated with the architectural design phase. Integration tests are performed to test the coexistence and communication of the internal modules within the system.
* **System Testing:** System testing is directly associated with the System design phase. System tests check the entire system functionality and the communication of the system under development with external systems. Most of the software and hardware compatibility issues can be uncovered during system test execution.
* **Acceptance Testing:** Acceptance testing is associated with the business requirement analysis phase and involves testing the product in user environment. Acceptance tests uncover the compatibility issues with the other systems available in the user environment. It also discovers the non functional issues such as load and performance defects in the actual user environment.
* **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. ... Test Execution. Test Cycle Closure.

Below are the phases of STLC:

* Requirements phase
* Planning Phase
* Analysis phase
* Design Phase
* Implementation Phase
* Execution Phase
* Conclusion Phase
* Closure Phase

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.
* When actual result deviates from the expected result while testing a software application or product then it results into a defect. Hence, any deviation from the specification mentioned in the product functional specification document is a defect. In different organizations it’s called differently like bug, issue, incidents or problem.
* When the result of the software application or product does not meet with the end user expectations or the software requirements then it results into a Bug or Defect. These defects or bugs occur because of an error in logic or in coding which results into the [**failure**](http://istqbexamcertification.com/what-is-a-failure-in-software-testing/)or unpredicted or unanticipated results.

**Additional Information about Defects / Bugs:**

While testing a software application or product if large number of defects are found then it’s called Buggy.

Because of the following reasons the software [**defects**](http://istqbexamcertification.com/what-is-defect-or-bugs-or-faults-in-software-testing/)arise:

– The person using the software application or product may not have enough knowledge of the product.

– Maybe the software is used in the wrong way which leads to the defects or[**failures**](http://istqbexamcertification.com/what-is-a-failure-in-software-testing/).

– The developers may have coded incorrectly and there can be defects present in the design.

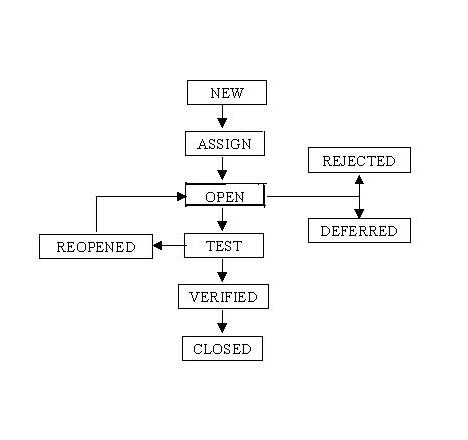
– Incorrect setup of the testing environments.

To know when defects in [**software testing**](http://istqbexamcertification.com/what-is-a-software-testing/) arise, let us take a small example with a diagram as given below.

We can see that **Requirement 1** is implemented correctly – we understood the customer’s requirement, designed correctly to meet that requirement, built correctly to meet the design, and so deliver that requirement with the right attributes: [**functionally**](http://istqbexamcertification.com/what-is-functional-testing-testing-of-functions-in-software/), it does what it is supposed to do and it also has the right[**non-functional**](http://istqbexamcertification.com/what-is-non-functional-testing-testing-of-software-product-characteristics/) attributes, so it is fast enough, easy to understand and so on.

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

The bug has different states in the Life Cycle. The Life cycle of the bug can be shown diagrammatically as follows:**[](http://istqbexamcertification.com/wp-content/uploads/2011/12/Bug-life-cycle1.jpg)**

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

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

In such situation, Firstly, I with my team members, 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.  
  
Its better to send the release late, but it should be bug free. Whats the fun, if we send the release on time, but it has many defects, that the customer is not satisfied.

**when do we use automation testing?**

**Bookmark**

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

when srs document is given by the client ,test lead prepares test plan.and test engineer prepares test cases. and they will be having a mom(minites of meeting)in middle .when the build is released we start executing testcase and if we found any defect .we will be reporting to developer and  he fixes it as bug and  we perform regression testing and we release the build to maintenece

**difference between load and performance testing?**

***Performance Testing = how fast is the system?Load Testing = how much volume can the system process?***

### ****Performance Testing****

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. Does it meet the required response time?

### ****Load Testing****

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.

Load testing is usually started as low numbers and gradually increased over a given period of time until it reaches the desired load on the system and then it ramps down.

**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 is a document, which has a set of test data, preconditions, expected results and postconditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution postcondition.

## Typical Test Case Parameters:

* Test Case ID
* Test Scenario
* Test Case Description
* Test Steps
* Prerequisite
* Test Data
* Expected Result
* Test Parameters
* Actual Result
* Environment Information
* Comments

### TDD- Test Driven Development, Behaviour Driven Development & Acceptance TestDriven Development

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

There are two key things in defects of the [**software testing**](http://istqbexamcertification.com/what-is-a-software-testing/). They are:

1)     Severity

2)     Priority

What is the difference between Severity and Priority?

**1)  Severity**:

It is the extent to which the [**defect**](http://istqbexamcertification.com/what-is-defect-or-bugs-or-faults-in-software-testing/) can affect the software. In other words it defines the impact that a given defect has on the system.**For example:** If an application or web page crashes when a remote link is clicked, in this case clicking the remote link by an user is rare but the impact of  application crashing is severe. So the severity is high but priority is low.

Severity can be of following types:

* **Critical:**The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable and there is no acceptable alternative method to achieve the required results then the severity will be stated as critical.

**how to estimate test cases?**