1.Testing is required for an effective performance of software application or product. It's important to ensure that the application should not result into any failures because it can be very expensive in the future or in the later stages of the development. It's required to stay in the business.

2.Desktop application

Web applications

enterprise applications

Mobile applications

3.To manage this level of complexity, a number of SDLC models or methodologies have been created, such as "waterfall"; "spiral"; "Agile software development"; "rapid prototyping"; "incremental"; and "synchronize and stabilize". SDLC can be described along a spectrum of agile to iterative to sequential.

4.The waterfall model is a sequential (non-iterative) design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, production/implementation and maintenance.

5.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.

6.Scrum is an agile way to manage a project, usually software development. Agile software development with Scrum is often perceived as a methodology; but rather than viewing Scrum as methodology, think of it as a framework for managing a process.

7.Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. 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.

8.

9.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.

10.The Sprint Planning Meeting is the first meeting to kick off the sprint. It is attended by the ScrumMaster, Development Team and the Product Owner, along with interested and invited stakeholders. The meeting is time boxed to 8 hours, so it's important to gather first thing, usually on a Monday morning.

11.Sprint Review Meeting. In Scrum, each sprint is required to deliver a potentially shippable product increment. This means that at the end of each sprint, the team has produced a coded, tested and usable piece of software. So at the end of each sprint, a sprint review meeting is held.

12.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

13.Backlog grooming is when the product owner and some, or all, of the rest of the team review items on the backlog to ensure the backlog contains the appropriate items, that they are prioritized, and that the items at the top of the backlog are ready for delivery

14.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.

15.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.

16.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.

17.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.

18.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.

19.

20.Defect life cycle, also known as Bug Life cycle is the journey of a defect cycle, which a defect goes through during its lifetime. It varies from organization to organization and also from project to project as it is governed by the software testing process and also depends upon the tools used

21.Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing can be done manually but is often automated.

22.Regression testing is the process of testing changes to computer programs to make sure that the older programming still works with the new changes. Regression testing is a normal part of the program development process and, in larger companies, is done by code testing specialists.

23.Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing.

24.Regression testing is the process of testing changes to computer programs to make sure that the older programming still works with the new changes. Regression testing is a normal part of the program development process and, in larger companies, is done by code testing specialists.

25.Sanity testing is performed when development team needs to know quick state of the product after they have done changes in the code or there is some controlled code change in a feature to fix any critical issue, and stringent release time-frame does not allow complete regression testing.

Smoke testing is conducted to ensure whether the most crucial functions of a program are working, but not bothering with finer details. (Such as build verification).

Smoke testing is normal health check up to a build of an application before taking it to testing in depth.

26.Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing can be done manually but is often automated.

27.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

28.Alpha testing is a type of acceptance testing; performed to identify all possible issues/bugs before releasing the product to everyday users or public. ... Alpha testing is carried out in a lab environment and usually the testers are internal employees of the organization.

Beta testing is often preceded by a round of testing called alpha testing.

@29.Black Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is NOT known to the tester

White Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester.

@30.he developed test cases are coverd all the functionality

of the application we can say testcases are enough.

Number of test cases can be determined by knowing the

31.This is a simple, if uncomfortable, situation to be in. Unfortunately, it does happen from time to time and you need to be ready for it.

The fact that the defect has been found close to the deadline is, in the short term, irrelevant. Your team has found a high severity defect, so you report it. Given the short timescales, you ensure that everyone who needs to know about it knows about it, so they have the information they need to determine -their- best course of action as soon as possible.

32.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.

33.SDLC starts with the analysis and definition phases, where the purpose of the software or system should be determined, the goals of what it needs to accomplish need to be established, and a set of definite requirements can be developed.

During the software construction or development stage, the actual engineering and writing of the application is completed. The software is designed and produced, while attempting to accomplish all of the requirements that were set forth within the previous stage.

Next in the software development lifecycle is the testing phase. Code produced during construction should be tested using static and dynamic analysis, as well as manual penetration testing to ensure that the application is not easily exploitable to hackers, which could result in a critical security breach. The advantage of using Veracode during this stage is that by using state of the art binary analysis (no source code required), the security posture of applications can be verified without requiring the use of any additional hardware, software or personnel.

34.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 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.

35.Load/Performance testing.

Compatibility testing.

Localization testing.

Security testing.

Reliability testing.

Stress testing.

Usability testing.

Compliance testing.

36.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. The mechanism for determining whether a software program or system has passed or failed such a test is known as a test oracle. In some settings, an oracle could be a requirement or use case, while in others it could be a heuristic. It may take many test cases to determine that a software program or system is considered sufficiently scrutinized to be released. Test cases are often referred to as test scripts, particularly when written - when they are usually collected into test suites.

37.The Test Strategy Document is a living document that is created in the project's Requirements Definition phase, after the Requirements have been specified. The Test Strategy document describes the scope, approach, resources and schedule for the testing activities of the project.

38.Entry and Exit criteria are required to start and end the testing. It is must for the success of any project.

If you do not know where to start and where to finish then your goals are not clear. By defining exit and entry criteria you define your boundaries. For instance, you can define entry criteria that the customer should provide the requirement document or acceptance plan. If these entry criteria are not met then you will not start the project. On the other end, you can also define exit criteria for your project. For

instance, one of the common exit criteria in projects is that the customer has successfully executed

the acceptance test plan.

39.Its also called test-driven design, is a method of software development in which unit testing is repeatedly done on source code. Write your tests watch it fails and then refactor it. The concept is we write these tests to check if the code we wrote works fine. After each test, refactoring is done and then the same or a similar test is performed again. The process is iterated as many times as necessary until each unit is functionally working as expected. TDD was introduced first by XP. I believe I have explained enough in simple terms.

BDD is similar in many ways to TDD except that the word “test” is replaced with the word “Behaviour”. It’s purpose is to help the the folks devising the system (i.e., the developer) identify appropriate tests to write–that is, tests that reflect the behavior desired by the stakeholders. BDD is usually done in very English-like language helps the Domain experts to understand the implementation rather than exposing the code level tests.

40.BDD provides a new vocabulary and thus focus for writing a unit test. Basically it is a feature driven approach to TDD.

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

Usually the severity is defined in terms of financial loss, damage to environment, company's reputation and loss of life. Priority of a defect is related to how quickly a bug should be fixed and deployed to live servers. When a defect is of high severity, most likely it will also have a high priority

42.3-Point Software Testing Estimation Technique.

Use – Case Point Method:

Work Breakdown Structure.

Wideband Delphi technique.

Function Point/Testing Point Analysis.

Percentage of development effort method.

Percentage distribution.

Best Guess.

43.Testing the complete application:

Is it possible? I think impossible. There are millions of test combinations. It’s not possible to test each and every combination both in manual as well as in automation testing. If you try all these combinations you will never ship the product

2) Misunderstanding of company processes:

Some times you just don’t pay proper attention what the company-defined processes are and these are for what purposes. There are some myths in testers that they should only go with company processes even these processes are not applicable for their current testing scenario. This results in incomplete and inappropriate application testing.

3) Relationship with developers:

Big challenge. Requires very skilled tester to handle this relation positively and even by completing the work in testers way. There are simply hundreds of excuses developers or testers can make when they are not agree with some points. For this tester also requires good communication, troubleshooting and analyzing skill.

4) Regression testing:

When project goes on expanding the regression testing work simply becomes uncontrolled. Pressure to handle the current functionality changes, previous working functionality checks and bug tracking.

5) Lack of skilled testers:

I will call this as ‘wrong management decision’ while selecting or training testers for their project task in hand. These unskilled fellows may add more chaos than simplifying the testing work. This results into incomplete, insufficient and ad-hoc testing throughout the testing life cycle.

6) Testing always under time constraint:

Hey tester, we want to ship this product by this weekend, are you ready for completion? When this order comes from boss, tester simply focuses on task completion and not on the test coverage and quality of work. There is huge list of tasks that you need to complete within specified time. This includes writing, executing, automating and reviewing the test cases.

7) Which tests to execute first?

If you are facing the challenge stated in point no 6, then how will you take decision which test cases should be executed and with what priority? Which tests are important over others? This requires good experience to work under pressure.

8 ) Understanding the requirements:

Some times testers are responsible for communicating with customers for understanding the requirements. What if tester fails to understand the requirements? Will he be able to test the application properly? Definitely No! Testers require good listening and understanding capabilities.

------------

9) Automation testing:

Many sub challenges – Should automate the testing work? Till what level automation should be done? Do you have sufficient and skilled resources for automation? Is time permissible for automating the test cases? Decision of automation or manual testing will need to address the pros and cons of each process.

10) Decision to stop the testing:

When to stop testing? Very difficult decision. Requires core judgment of testing processes and importance of each process. Also requires ‘on the fly’ decision ability.

11) One test team under multiple projects:

Challenging to keep track of each task. Communication challenges. Many times results in failure of one or both the projects.

12) Reuse of Test scripts:

Application development methods are changing rapidly, making it difficult to manage the test tools and test scripts. Test script migration or reuse is very essential but difficult task.

13) Testers focusing on finding easy bugs:

If organization is rewarding testers based on number of bugs (very bad approach to judge testers performance) then some testers only concentrate on finding easy bugs those don’t require deep understanding and testing. A hard or subtle bug remains unnoticed in such testing approach.

14) To cope with attrition:

Increasing salaries and benefits making many employees leave the company at very short career intervals. Managements are facing hard problems to cope with attrition rate. Challenges – New testers require project training from the beginning, complex projects are difficult to understand, delay in shipping date!

44.By design we mean to create a plan for how to implement an idea and technique is a method or way for performing a task. So, Test Design is creating a set of inputs for given software that will provide a set of expected outputs.

45.

46.An operating system has three main functions: (1) manage the computer's resources, such as the central processing unit, memory, disk drives, and printers, (2) establish a user interface, and (3) execute and provide services for applications software.

47.Provides a centralized repository for tracking defects across projects.

Provides automated notifications of resource assignments.

Ability to define defect resolution status in order to map back to your defect management process.

Ability to provide management reporting, like the number of open defects grouped by various criteria such as open defects by project, severity, and priority.

48.Manager will assign the work

49.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.

50.

51.

52.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.

53.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.

54.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.

55.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.

56.There will always be bugs that get past a tester and land in production. I have even had bugs that where in my face, we researched it, thought it was a fluke, because we couldn't reproduce it and then released the issue into production.

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

This is a simple, if uncomfortable, situation to be in. Unfortunately, it does happen from time to time and you need to be ready for it.

The fact that the defect has been found close to the deadline is, in the short term, irrelevant. Your team has found a high severity defect, so you report it. Given the short timescales, you ensure that everyone who needs to know about it knows about it, so they have the information they need to determine -their- best course of action as soon as possible.

You must -absolutely- not ever hold off from reporting an issue, at least to your local management structure. That would, at the very least, ruin the reputation of your team and could potentially have much more serious consequences.

The next thing to do is to determine the answer to the obvious question: "Why was this found so late?". There are many reasons why this situation could arise - your test preparation could've been too light, you could've mis-prioritised some work, there may simply have been too much to do. As a member of the test team, you need to know what caused the issue and therefore how you can reduce the risk of it happening again.

Obviously, we don't live in a perfect world and it's possible that no action may be taken to resolve the defect before release. It's entirely possible that it makes more sense to go live with a bug and then release a quick fix, than to decide not to release at all. For what it's worth, this has happened to me a couple of times over the past dozen or so years

what tester will do in each phase of SDLC?

Software Development Life Cycle (SDLC) is the procedure of mounting application with appropriate investigation, plan, execution and preservation.

Testing stages have lot of significance in SDLC due to a most important part in executing and fault rectification. SDLC phases are followed by testing and implementation sequence of all software. Here we are describing the 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.

what is load testing and performance testing?

Load testing: Load testing is performance testing technique using which the response of the system is measured under various load conditions. The load testing is performed for normal and peak load conditions.

Performance testing: Performance testing, a non-functional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workload. Performance testing measures the quality attributes of the system, such as scalability, reliability and resource usage.

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 a testcase?

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.

what is Exit and Entry criteria?

Exit criterion is used to determine whether a given test activity has been completed or NOT. Exit criteria can be defined for all of the test activities right from planning, specification and execution.

Exit criterion should be part of test plan and decided in the planning stage.

Examples of Exit Criteria:

* Verify if All tests planned have been run.
* Verify if the level of requirement coverage has been met.

Entry criterion is used to determine when a given test activity should start. It also includes the beginning of a level of testing, when test design or when test execution is ready to start.

## Examples for Entry Criterion:

* Verify if the Test environment is available and ready for use.
* Verify if test tools installed in the environment are ready for use.

What is TDD and BDD?

***TDD*** is an iterative development process. Each iteration starts with a set of tests written for a new piece of functionality. These tests are supposed to fail during the start of iteration as there will be no application code corresponding to the tests. In the next phase of the iteration Application code is written with an intention to pass all the tests written earlier in the iteration. Once the application code is ready tests are run.

What is severity and priority in defect?

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.

Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the developer has to fix it at the earliest. The priority status is set based on the customer requirements. **For example:** If the company name is misspelled in the home page of the website, then the priority is high and severity is low to fix it.

How to estimate test cases?

List of Software Test Estimation Techniques

* Work Breakdown Structure
* 3-Point Software Testing Estimation Technique
* Wideband Delphi technique
* Function Point/Testing Point Analysis
* Use – Case Point Method
* Percentage distribution
* Ad-hoc method

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

**One test team under multiple projects:** Challenging to keep track of each task. Communication challenges. Many times results in failure of one or both the projects.