1) Why testing is required?

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

2) 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

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 waterfal in SDLC?

Ans: 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.

3)what is the process in agile model ?

Ans : 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.

4) what is scrum methodology ?

Ans : Scrum is a management and control process that cuts through complexity to focus on building software that meets business needs. Management and teams are able to get their hands around the requirements and technologies, never let go, and deliver working software, incrementally and empirically.

5) 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, is sometimes referred to as a stand-up, a morning roll-call or a daily scrum.

1) What did you do yesterday?

2) What will you do today?

3) Are there any impediments in your way?

7) 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.

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.

8) what is burndown chart and velocity ?

Ans :  Burndown Chart is a visual measurement tool that shows the completed work per day against the projected rate of completion for the current project release.

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.

8) what is product backlog item and sprint backlog items ?

Ans : The Product Backlog is an ordered list of everything that might be needed in the final product of the project. In other words, parts of the expected final product (a wish list). All items are described in simple, non-technical, business language, and all of them are presentable to every stakeholder. Every requirement and every change in the project will be reflected in it.

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.

9) what is user acceptance criteria test cases ?

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

WHEN WE WILL USE USER ACCEPTANCE TESTING :-

Business Requirements must be available

Application Code should be fully developed

Unit Testing, Integration Testing & System Testing should be completed

No Show stoppers, or High or Medium defects in the System Integration Test Phase

Only Cosmetic errors are acceptable before UAT

Regression Testing should be completed with no major defects

All the reported defects should be fixed and tested

Traceability matrix for all testing should be completed

UAT Environment must be ready

Sign off mail or communication from System Testing Team that the system is ready for UAT execution.

10) what is v model?

Ans : 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.

11) 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; however the basis remains the same.

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

12) what is defect?

Ans : A Software Defect / Bug is a condition in a software product which does not meet a software requirement (as stated in the requirement specifications) or end-user expectations (which may not be specified but are reasonable).

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

Ans : Defects and failures basically arise from:

* [Errors in the specification](http://istqbexamcertification.com/what-is-black-box-specification-based-also-known-as-behavioral-testing-techniques/), design and implementation of the software and system
* Errors in use of the system
* Environmental conditions
* Intentional damage
* Potential consequences of earlier errors

**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 tracking is an important process in software engineering as Complex and business critical systems have hundreds of defects. One of the challenging factors is Managing, evaluating and prioritizing these defects. The number of defects gets multiplied over a period of time and to effectively manage them, defect tracking system is used to make the job easier.

13) defect life cycle ?

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.

[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.
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](http://istqbexamcertification.com/what-is-the-difference-between-severity-and-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.

14)Different types of testing:

Ans : **Software Testing Types:**

**Black box testing** – Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.

**White box testing** – This testing is based on knowledge of the internal logic of an application’s code. Also known as Glass box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

**Unit testing** – Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. may require developing test driver modules or test harnesses.

**Integration testing** – Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

**Functional testing** – This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black-box type testing geared to functional requirements of an application.

**System testing** – Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.

**End-to-end testing** – Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

**Sanity testing** – Testing to determine if a new software version is performing well enough to accept it for a major testing effort. If application is crashing for initial use then system is not stable enough for further testing and build or application is assigned to fix.

**Regression testing** – Testing the application as a whole for the modification in any module or functionality. Difficult to cover all the system in regression testing so typically automation tools are used for these testing types.

**Acceptance testing** -Normally this type of testing is done to verify if system meets the customer specified requirements. User or customer do this testing to determine whether to accept application.

**Load testing** – Its a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system’s response time degrades or fails.

**Stress testing** – System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.

**Performance testing** – Term often used interchangeably with ‘stress’ and ‘load’ testing. To check whether system meets performance requirements. Used different performance and load tools to do this.

**Usability testing** – User-friendliness check. Application flow is tested, Can new user understand the application easily, Proper help documented whenever user stuck at any point. Basically system navigation is checked in this testing.

**Install/uninstall testing** – Tested for full, partial, or upgrade install/uninstall processes on different operating systems under different hardware, software environment.

**Recovery testing** – Testing how well a system recovers from crashes, hardware failures, or other catastrophic problems.

**Security testing** – Can system be penetrated by any hacking way. Testing how well the system protects against unauthorized internal or external access. Checked if system, database is safe from external attacks.

**Compatibility testing** – Testing how well software performs in a particular hardware/software/operating system/network environment and different combination s of above.

**Comparison testing** – Comparison of product strengths and weaknesses with previous versions or other similar products.

**Alpha testing** – In house virtual user environment can be created for this type of testing. Testing is done at the end of development. Still minor design changes may be made as a result of such testing.

**Beta testing** – Testing typically done by end-users or others. Final testing before releasing application for commercial purpose.

15)What is unit testing?

Ans : **Unit testing** – Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. may require developing test driver modules or test harnesses.

16)when do we use regression testing?

Ans : 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.

17)What is integration testing?

Ans: **Integration testing** – Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

18)when do we use integration testing?

Ans : Meaning: We normally do Integration [testing after “Unit testing”](http://www.softwaretestinghelp.com/unit-testing/).

Once all the individual units are created and tested, we start combining those “Unit Tested” modules and start doing the integrated testing. So the meaning of Integration testing is quite straight forward- Integrate/combine the unit tested module one by one and test the behavior as a combined unit.

19)when do we use smoke testing and sanity testing?

Ans : **Smoke testing** is a type of [software testing](http://istqbexamcertification.com/what-is-a-software-testing/) which ensures that the major functionalities of the application are working fine. It is a non-exhaustive testing with very limited test cases to ensure that the important features are working fine and we are good to proceed with the detailed testing.

**when we do use smoke testing** :

* The purpose of the smoke testing is to ensure that the critical functionalities of an application are working fine.
* This is a non-exhaustive testing with very limited number of test cases.
* It is also known as Build verification testing where the build is verified by testing the important features of the application and then declaring it as good to go for further detailed testing.
* Smoke testing can be done by developers before releasing the build to the testers and post this it is also tested by the testing team to ensure that the build is stable enough to perform the detailed testing.
* Usually smoke testing is performed with positive scenarios and with valid data.
* It is a type of shallow and wide testing because it covers all the basic and important functionalities of an application.
* Usually the smoke testing is documented.
* Smoke testing is like a normal health check up of the build of an application.

**Sanity testing** : Sanity testing, a software testing technique performed by the test team for some basic tests. The aim of basic test is to be conducted whenever a new build is received for testing.

After completion of regression testing the Sanity testing is started to check the defect fixes & changes done in the software application is not breaking the core functionality of the software.

**Here are the few consolidated points of Sanity testing**:

* Sanity testing follows narrow and deep approach with detailed testing of some limited features.
* Sanity testing is typically non-scripted.
* Sanity testing is a sub-set of regression testing.
* Sanity testing is cursory testing to prove software application is working as mention in the specification documents & meets the user needs.
* Sanity testing is used to verify the requirements of end users are meeting or not.
* Sanity testing to check the after minor fixes the small section of code or functionality is working as expected & not breaking related functionality.

20)what is unit testing?

Ans : **Unit testing** – Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. may require developing test driver modules or test harnesses.

21)what is UAT?

Ans : User acceptance testing (UAT) is the last phase of the software testing process. UAT directly involves the intended users of the software. UAT can be implemented by making software available for a free beta trial on the Internet or through an in-house testing team comprised of actual software users.

22)what is alpha and beta testing?

Ans : **Alpha testing** – In house virtual user environment can be created for this type of testing. Testing is done at the end of development. Still minor design changes may be made as a result of such testing.

**Beta testing** – Testing typically done by end-users or others. Final testing before releasing application for commercial purpose.

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

Ans :

|  |  |  |
| --- | --- | --- |
| Criteria | Black Box Testing | White Box Testing |
| Definition | 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. |
| Levels Applicable To | Mainly applicable to higher levels of testing:[Acceptance Testing](http://softwaretestingfundamentals.com/acceptance-testing/)  [System Testing](http://softwaretestingfundamentals.com/system-testing/) | Mainly applicable to lower levels of testing:[Unit Testing](http://softwaretestingfundamentals.com/unit-testing/)  [Integration Testing](http://softwaretestingfundamentals.com/integration-testing/) |
| Responsibility | Generally, independent Software Testers | Generally, Software Developers |
| Programming Knowledge | Not Required | Required |
| Implementation Knowledge | Not Required | Required |
| Basis for Test Cases | Requirement Specifications | Detail Design |

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

Ans :

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

Ans : 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. 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.

26)when do we use automation testing?

Ans : Why do you automate?

I automate for multiple reasons:

* In order to break software more quickly or more effectively in a repeatable manner.
* In order to help me execute complex test cases in a more timely manner.
* In order to speed up testing.
* In order to help me set up or tear down test data (in which case my test case may or may not be automated).
* In order to improve programming skills

27)what tester will do in each phase of SDLC?

Ans : the role of a tester is to develop[prepare] the testcase, during sdlc,

both tester and developers are provided with same requirement, initially they interact with each other during requirement analysis phase, once they have understood, dev team will be busy with developing it and tester will be busy in preparing testcase,

dev team will develop some of the module and give it for testing as build 1and begin developing other modules ,tester will test build 1 by executing testcase,and reply to dev, regarding any issued faced,or bug found, and inturn dev team willl respond, this is what tester do during sdlc. till the product becomes stable, once the product is free of bugs n meets clients requirements, they deliver it to them

28)difference between load and performance testing?

### Ans : 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.

### 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.

30)different types of non-functional testing types?

## Ans : Types of Non Functional Testing

1. Performance Testing

2. Load Testing

3. Stress Testing

4. Volume Testing

5. Failover Testing

6. Security Testing

7. Compatibility Testing

8. Usability Testing

9. Scalability Testing

30)what is test case?

Ans)  test case is a document, which has a set of test data, preconditions, expected results and post conditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

31)what is test planning/test strategy document

Ans: Test plan document contains different section like

Types of testing :

Exit and Entry criteria :

32)what is TDD and BDD (cucumber framework)

Ans : Test-driven development (TDD) is a software development process that relies on the repetition of a very short development cycle: requirements are turned into very specific test cases, then the software is improved to pass the new tests, only.

behavior-driven development (BDD) is a [software development process](https://en.wikipedia.org/wiki/Software_development_process) that emerged from [test-driven development](https://en.wikipedia.org/wiki/Test-driven_development) (TDD). Behavior-driven development combines the general techniques and principles of TDD with ideas from [domain-driven design](https://en.wikipedia.org/wiki/Domain-driven_design) and [object-oriented analysis and design](https://en.wikipedia.org/wiki/Object-oriented_analysis_and_design) to provide software development and management teams with shared tools and a shared process to collaborate on software development.

33)what is priority and severity in defect?

Ans : **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.

**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.

34)how to estimate test cases?

Ans :

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.

35)what is most challenge defect u came across?

Ans : Biggest challenge as a tester is trying to make other testers aware of the constant need to learn our occupation.

Too many of them are just doing their daily chores and forget to read and join the testing discussion.

The 2nd challenge, is making developers aware of this profession, and having them learn a bit too - as they need it for Unit testing.

36)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 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.

37)What are test design techniques ?

Ans : Structured analysis and design technique (SADT) is a systems engineering and software engineering methodology for describing systems as a hierarchy of functions. SADT is a structured analysis modeling language, which uses two types of diagrams: activity models and data models.

38)if we dont have time to test call test cases what we will do ?

Ans :

39)how we learn the functionality of system?

Ans :

40)what are the tools to manage defects/stories?

Ans :

* Stryka: 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

1. Be online
2. Enable quick collecting of story titles
3. Enable to map/share/group the collected stories
4. Allow open discussion around stories
5. Enable collaboration for collecting stories and elaborating their meanings

41)who will assign the work?

Ans : Generally *test* lead *assign the work* to *test* engineer. It depends on the organization where *Test* leads and *Test* engineers.

42)types of test metrics we use normally

Ans

43)what is traceability matrix?

Ans : 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.

44)what are typical environments we have in projects ?

Ans :

* Identify activities
* Estimate times and resources
* Identify relationships and dependencies
* Identify schedule restraints
* Create the schedule

45)what are different defect metrics and measurements we prepare ?

Ans : Test Metrics are used to,

1. Take the decision for next phase of activities such as, estimate the cost & schedule of future projects.
2. Understand the kind of improvement required to success the project
3. Take decision on process or technology to be modified etc.

**Metrics can be defined as “STANDARDS***OF**MEASUREMENT***”.**

Software Metrics are used to measure the quality of the project. Simply, Metric is a unit used for describing an attribute. Metric is a scale for measurement.

Suppose, in general, “Kilogram” is a metric for measuring the attribute “Weight”. Similarly, in software, “How many issues are found in thousand lines of code?”, h*ere No. of issues is one measurement & No. of lines of code is another measurement. Metric is defined from these two measurements*.

46)What is staging environment?

Ans: Strategic environmental assessment (SEA) is a systematic decision support process, aiming to ensure that environmental and possibly other sustainability aspects are considered effectively in policy, plan and program making.

47)what is development environment ?

Ans : A deployment environment is a collection of configured clusters, servers, and middleware that collaborate to provide an environment to host software modules. For example, a deployment environment might include a host for message destinations, a processor or sorter of business events, and administrative programs.

48)what is QA environment ?

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

49)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.