**Harish Kumar- QA Phone: (469) 607-6077**

**Room: K204 Email:harishsrini15@gmail.com**

1. **Why testing is required?**

Testing is required for an effective performance of software application or product.

1. **What types of application we test?**

Web applications

Desktop/windows applications

Mobile applications

Web services (SOAP/REST)

ETL jobs, database validations

Back end/batch programs/windows services

1. **What is SDLC and different phases in SDLC?**

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

1. **What is waterfall in SDLC?**

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

1. **What is agile method?**

Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through  
collaboration between self-organizing cross-functional teams.

1. **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.

1. **What is scrum methodology?**

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. Scrum itself is a simple framework for effective team collaboration on complex software projects.

1. **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?

Standing, rather than sitting, reinforces the idea that the meeting is intended to be short and discourages wasted time. The stand-up is not meant to be a place to solve problems, but rather to make the team aware of current status. If discussion is needed, a longer meeting with appropriate parties can be arranged.

1. **What is product back log items?**

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.

1. **What is user story/feature/sprint back log items and tasks in user story?**

Epic - Create a web site (T shirt size) (L)

Story - Create the Home page (1 point)

Task - Create a button on the home page that is red (1 hour)

Features are what a system is doing. User stories are just one way amongst others to capture features.

Epic - Is simply a story, but is considered so large that it needs to be broken down into multiple stories.

Story - Is essentially a requirement that is in a low enough detail that it can be estimated.

Task - Developers, testers may break the story down further into tasks to allow them to estimate, develop and test it. (If a story has a lot of tasks, it could be worth classing it as an epic and breaking it down into multiple stories.)

1. **What is sprint planning and spring retro?**

Sprint planning is a collaborative effort involving a Scrum Master, 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.

Spring retro: The sprint retrospective is usually the last thing done in a sprint. Many teams will do it immediately after the sprint review. The entire team, including both the Scrum Master and the product owner should participate.

1. **What is Sprint review?**

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.

1. **What is Sprint grooming?**

Product backlog refinement—sometimes called product backlog grooming in reference to keeping the backlog clean and orderly—is a meeting that is held near the end of   
one sprint to ensure the backlog is ready for the next sprint.

1. **What is burndown chart and velocity**

Burndown Charts. The burndown is a chart that shows how quickly you and your team are burning through your customer's user stories.

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 and sprint backlog items**

The Product Backlog is an ordered list of everything that might be needed in the product and is the single source of requirements.

The Sprint Backlog is a forecast (…) what functionality will be in the next Increment and the work needed to deliver that functionality. The Sprint Backlog makes visible all of the work (…) The Sprint Backlog is a plan (…)

1. **What is user acceptance criteria test cases**

User acceptance is a type of testing performed by the Client to certify the system with respect to the requirements that was agreed upon. This testing happens in the final phase of testing before moving the software application to Market or Production environment.

1. **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.

1. **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.

1. **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.

1. **How to arise a defect and what we specify while logging defect?**

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 is related to the bug found during testing.

1. **Defect Life cycle?**

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.

1. **Different types of testing:**

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.

Incremental integration testing – Bottom up approach for testing i.e. continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately done by programmers or by testers.

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 – Regression testing is nothing but full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.

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 – It’s 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.

1. **When do we use regression testing?**

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

1. **When do we use integration testing?**

In Integration Testing, individual software modules are integrated logically and tested as a group.

A typical software project consists of multiple software modules, coded by different programmers. Integration testing focuses on checking data communication amongst these modules.

1. **When do we use smoke testing and sanity testing?**

Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine

For Example a typical smoke test would be - Verify that the application launches successfully, Check that the GUI is responsive ... etc.

Sanity Testing is done to check the new functionality / bugs have been fixed.

The objective of the testing is to verify the "rationality" of the system in order to proceed with more rigorous testing

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 black box and white box techniques. The aim is to carry out the tasks that a typical user might perform.

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.

1. **What is UAT?**

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.

UAT is also known as beta testing, application testing or end user testing

1. **What is alpha and beta testing?**

Alpha testing. Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to 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.

1. **When do we use white box testing and block box testing?**

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.

Mainly applicable to higher levels of testing: Acceptance Testing, System Testing.

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.

Mainly applicable to lower levels of testing: Unit Testing, Integration Testing.

1. **What we will do if we don't have a time to test all stories/ execute test cases?**

Since it’s rarely possible to test every possible aspect of an application, every possible combination of events, every dependency, or everything that could go wrong, risk analysis is appropriate to most software development projects. Use risk analysis to determine where testing should be focused. This requires judgment skills, common sense and experience. The checklist should include answers to the following questions: · Which functionality is most important to the project’s intended purpose? · Which functionality is most visible to the user? · Which functionality has the largest safety impact? · Which functionality has the largest financial impact on users? · Which aspects of the application are most important to the customer? · Which aspects of the application can be tested early in the development cycle? · Which parts of the code are most complex and thus most subject to errors? · Which parts of the application were developed in rush or panic mode? · Which aspects of similar/related previous projects caused problems? · Which aspects of similar/related previous projects had large maintenance expenses? · Which parts of the requirements and design are unclear or poorly thought out? · What do the developers think are the highest-risk aspects of the application? · What kinds of problems would cause the worst publicity? · What kinds of problems would cause the most customer service complaints? · What kinds of tests could easily cover multiple functionality? · Which tests will have the best high-risk-coverage to time-required ratio?

1. **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. That's a different discussion, though.

For what it's worth, this has happened to me a couple of times over the past dozen or so years. It's a horrible feeling but as long as you've done your best, there's nothing more you can do.

1. **When do we use automation testing?**

* 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 timelier 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

1. **What tester will do in each phase of SDLC?**

The Role of a Tester in SDLC

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 it’s fixed by the developer the tester performs a regression testing

1. **Difference between load and performance testing?**

Load test: any test that involves to put a determined load on an application to verify how it behaves (i.e.: response time);

Performance test: it is a load test limited by the load defined by the specification of the application - the test is to verify or confirm that the application will work at the planned performance;

Stress test: there are many meanings for stress, but I tried to educate my customers to use the term stress only when you want to load the application beyond the specification definition, to understand the maximum capacity of application before it breaks. It helps for example to foresee any problems in a near future and prepare IT teams to be aware of the application capacity;

And there is Stability test: execute a load test during a long period of time to understand how stable an application in a long run is. It helps to determine, for example, if an application has memory leaks, which is very difficult to find in a normal load test.

1. **Different types of non-functional testing types?**

Examples of non-functional tests include:

* Load/Performance testing.
* Compatibility testing.
* Localization testing.
* Security testing.
* Reliability testing.
* Stress testing.
* Usability testing.
* Compliance testing.

1. **What is test case?**

A 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.

1. **What is test planning/test strategy document?**

A test plan for software project can be defined as a document that defines the scope, objective, approach and emphasis on a software testing effort.

Ans: Test plan document contains different section like

Types of testing:

1. **Exit and Entry criteria?**

Entry Criteria for QA testing is defined as “Specific conditions or on-going activities that must be present before a process can begin”. In the Systems Development Life Cycle it also specifies which entry criteria are required at each phase. Additionally, it is also important to define the time interval or required amount of lead time that an entry criteria item is available to the process. Input can be divided into two categories. The first is what we receive from development. The second is what we produce that acts as input to later test process steps.

The type of required input from development includes:

1. Technical Requirements/Statement of Need
2. Design Document
3. Change Control
4. Turnover Document

The type of required input from test includes:

1. Evaluation of available software test tools
2. Test Strategy
3. Test Plan
4. Test Incident Reports

By referencing the Entry Exit Criteria matrix, we get the clarity of the deliverables expected from each phase. The matrix should contain “date required” and should be modified to meet the specific goals and requirements of each test effort based on size and complexity.

EXIT CRITERIA

Exit Criteria is often viewed as a single document commemorating the end of a life cycle phase. Exit Criteria is defined as “The specific conditions or on-going activities that must be present before a life cycle phase can be considered complete. The life cycle specifies which exit criteria are required at each phase”. This definition identifies the intermediate deliverables, and allows us to track them as independent events.

The type of output from test includes:

1. Test Strategy
2. Test Plan
3. Test Scripts/Test Case Specifications
4. Test Logs
5. Test Incident Report Log
6. Test Summary Report/Findings Report

By identifying the specific Exit criteria, we are able to identify and plan how these steps and processes fit into the life cycle. All of the Exit Criteria listed above, less the Test Summary/Findings Report; act as Entry Criteria to alter process.

1. **What is TDD and BDD (cucumber framework)?**

It’s 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.

Behavior-driven development combines the general techniques and principles of TDD with ideas from domain-driven design

DDD-Domain Driven Testing

BDD is similar in many ways to TDD except that the word “test” is replaced with the word “Behavior”. Its purpose is to help 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. It’s defined in a GWT format, GIVEN WHEN & THEN.

1. **What is priority and severity in defect?**

In software testing, defect severity can be defined as the degree of impact a defect has on the development or operation of a component application being tested.

Higher effect on the system functionality will lead to the assignment of higher severity to the bug. Quality Assurance engineer usually determines the severity level of defect.

1. **How to estimate test cases?**

* 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.

1. **What is most challenge defect u came across?**
2. If a defect is found and later is not reproducible then steps to reproduce that defect is a challenge.
3. I was testing a mobile app A, there was multiple features which could be added as a bookmark. While going through one flow, it was working fine.

But if via one flow, the bookmark was added and removed for Y feature and the bookmark section was accessed from second flow, then automatically bookmark was added for Y feature. After spending one full day, it took 17 steps to find the exact issue.

1. There was another app B having a field that had maximum character limit of 13 characters, I entered 12 characters alphabets and 13th character as an emoji since emoji takes 2 characters the app crashed. After this was fixed, I was not allowed to enter an emoji then I copied emoji from other place and pasted in that field again the app crashed.
2. There was one more app C, where s filed had limit of 30 characters. I copied 31 characters from notepad in mobile and pasted it in the field and it accepted.

Basically, I have many such scenarios so I do not know what exactly you are looking for.

1. **How to deal the production defects?**

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 (staging) and release the fix to production after we verify.

We have to create a defect on current **sprint/release** so that developer will add this code to the current sprint/release

1. **Test design techniques?**

Broadly speaking there are two main categories of Test Design Techniques. They are:

* Static Techniques
* Dynamic Techniques

1. **What are the tools to manage defects/stories?**
   1. Bugzilla
   2. Jira
   3. Lean Testing
   4. Mantis
   5. Trac
   6. Redmine
   7. HP ALM/Quality Center
   8. FogBugz
   9. IBM Rational ClearQuest
   10. Lighthouse
   11. Zoho bug tracker
   12. The Bug Genie
   13. Bug Host
   14. DevTrack
   15. BugNET
   16. eTraxis
2. **What is traceability matrix?**

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.

1. **What is staging environment?**

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.

1. **What is development environment?**

A development environment is a collection of procedures and tools for developing, testing and debugging an application or program. The development environment normally has three server tiers, called development, staging and production.

1. **What is QA environment?**

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 Wave set application.

1. **What is production environment?**

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.