1)Why testing is required?

Ans: 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) What types of application we test

Ans: web applications

desktop/windows applications

Mobile applications

web services (SOAP/REST)

ETL jobs, database validations

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

There are following six phases in every Software development life cycle model:

1)Requirement gathering and analysis.

2)Design.

3)Implementation or coding.

4)Testing.

5)Deployment.

6)Maintenance.

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.

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

what is scrum methodology

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.

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.

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

what is sprint planning and spring retro

Ans:

Sprint planning:

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

In Scrum, the sprint planning meeting is attended by the product owner, ScrumMaster and the entire Scrum team. Outside stakeholders may attend by invitation of the team, although this is rare in most companies.

During the sprint planning meeting, the product owner describes the highest priority features to the team. The team asks enough questions that they can turn a high-level user story of the product backlog into the more detailed tasks of the sprint backlog.

The product owner doesn't have to describe every item being tracked on the product backlog. A good guideline is for the product owner to come to the sprint planning meeting prepared to talk about two sprint's worth of product backlog items.

To make an example really simple, suppose a team always finishes five product backlog items. Their product owner should enter the meeting prepared to talk about the top 10 priorities.

There are two defined artifacts that result from a sprint planning meeting:

1)A sprint goal

2)A sprint backlog

A sprint goal is a short, one- or two-sentence, description of what the team plans to achieve during the sprint. It is written collaboratively by the team and the product owner. The following are example sprint goals on an eCommerce application:

Implement basic shopping cart functionality including add, remove, and update quantities.

Develop the checkout process: pay for an order, pick shipping, order gift wrapping, etc.

The sprint goal can be used for quick reporting to those outside the sprint. There are always stakeholders who want to know what the team is working on, but who do not need to hear about each product backlog item (user story) in detail.

The success of the sprint will later be assessed during the sprint review meeting against the sprint goal, rather than against each specific item selected from the product backlog.

The sprint backlog is the other output of sprint planning. A sprint backlog is a list of the product backlog items the team commits to delivering plus the list of tasks necessary to delivering those product backlog items. Each task on the sprint backlog is also usually estimated.

Sprint retrospective:

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

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 ScrumMaster and the product owner should participate. You can schedule a scrum retrospective for up to an hour, which is usually quite sufficient. However, occasionally a hot topic will arise or a team conflict will escalate and the retrospective could take significantly longer.

Although there are many ways to conduct an agile sprint retrospective, our recommendation is to conduct it as a start-stop-continue meeting. This is perhaps the simplest, but often the most effective way to conduct a retrospective. Using this approach each team member is asked to identify specific things that the team should:

1)Start doing

2)Stop doing

3)Continue doing

There are many variations on this simple format. The ScrumMaster can facilitate this sprint retrospective meeting by asking everyone to just shout out ideas during the scrum. The ScrumMaster can go around the room asking each person to identify any one thing to start, stop or continue. Or, for example, he or she can tell everyone to focus on identifying something to stop this time because not much attention has been paid to things to stop in recent retrospectives.

After an initial list of ideas has been brainstormed, teams will commonly vote on specific items to focus on during the coming sprint. At the end of the sprint, the next retrospective is often begun by reviewing the list of things selected for attention in the prior sprint retrospective.

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 and sprint backlog 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.

what is user acceptance criteria test cases

Firstly, the criteria by which the software is considered to be “working” needs to be assembled. These are likely to be collated from the system requirements, and user stories. Next, a set of UAT test cases must be created. Centric defines a UAT test case as: ... Each case covers a specific usage scenario of the software.

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

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

what is defect?

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.

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

defect lifecycle

New:When the defect is logged and posted for the first time

Assign:After the tester has posted the bug,the lead of the tester approves that bug is genuine and he assign the bug to corresponding developer and the developer team.

Open:At this state the developer has started analyzing abd working on the defect fix

Fixed:when the developer makes accessary code changes and verifies the changes then he/she can make bug status as Fixed and the bug is passed to testing team

Verified:The tester tests the bug again after it got fixed by the developer.If the bug is not present in the software,be approved that the bug is fixed and changes the status is Verified.

Closed:Once the bug is fixed,and it is fixed, and it is retested by the tester, and 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

Reopen:If the bug still exits even after the bug is fixed bythe developer ,the tester changes the status to “reopen”.the bug goes to the life cycle once again.

Reject:If the developer feels that the bug is not genuine,he rejects the bug

Deffered: When the bug is changed to the defferedstate,itmeanse the bug is expected to be fixed to next release.this Can be due to various reasons.

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.

Different types of testing:

What is unit testing?

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 is often automated but it can also be done manually

when do we use regression testing?

Regression testing is performed when changes are made to the existing functionality of the software or if there is a bug fix in the software. Regression testing can be achieved through multiple approaches, if a test all approach is followed it provides certainty that the changes made to the software have not affected the existing functionalities, which are unaltered, in any way.

What is integration testing?

Ans: 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.

when do we use integration testing?

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e., unit testing.

when do we use smoke testing and sanity testing?

SMOKE TESTING:

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

Smoke testing originated in the hardware testing practice of turning on a new piece of hardware for the first time and considering it a success if it does not catch fire and smoke. In software industry, smoke testing is a shallow and wide approach whereby all areas of the application without getting into too deep, is tested.

A smoke test is scripted, either using a written set of tests or an automated test

A Smoke test is designed to touch every part of the application in a cursory way. It’s shallow and wide.

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.

SANITY TESTING:

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

A sanity test is a narrow regression test that focuses on one or a few areas of functionality. Sanity testing is usually narrow and deep.

A sanity test is usually unscripted.

A Sanity test is used to determine a small section of the application is still working after a minor change.

Sanity testing is a cursory testing, it is performed whenever a cursory testing is sufficient to prove the application is functioning according to specifications. This level of testing is a subset of regression testing.

Sanity testing is to verify whether requirements are met or not, checking all features breadth-first.

what is UAT?

In software development, user acceptance testing (UAT) - also called beta testing, application testing, and end user testing - is a phase of software development in which the software is tested in the "real world" by the intended audience.

what is alpha and beta testing?

alpha testing:

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

This is a form of internal acceptance testing performed mainly by in-house software QA and testing teams. Alpha testing is the last testing done by test teams at development site after the acceptance testing and before releasing the software for beta test. Alpha testing can also be done by potential users or customers of the application. But still this is a form of in-house acceptance testing.

beta testing:

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

This is a testing stage followed by internal full alpha test cycle. This is the final testing phase where companies release the software for few external user groups outside the company test teams or employees. This initial software version is called as beta version. Most companies gather user feedback in this release.

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

White-box testing:is a method of testing the application at the level of the source code.These test cases are derived through the use of the design techniques like control flow testing, data flow testing, branch testing, path testing, statement coverage and decision coverage as well as modified condition/decision coverage. White-box testing is the use of these techniques as guidelines to create an error free environment by examining any fragile code. These White-box testing techniques are the building blocks of white-box testing, whose essence is the careful testing of the application at the source code level to prevent any hidden errors later on.These different techniques exercise every visible path of the source code to minimize errors and create an error-free environment.

Black-box testing:is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance.

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

Use risk analysis, along with discussion with project stakeholders, to determine where testing should be focused.

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. This requires judgement skills, common sense, and experience.

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

when do we use automation testing?

what tester will do in each phase of SDLC?

difference between load and performance testing?

different types of non-functional testing types?

what is test case?

what is test planning/test strategy document

Ans: Test plan document contains different section like

Types of testing :

Exit and Entry criteria :

what is TDD and BDD (cucumber framework)

TDD: 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: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. Its defined in a GWT format, GIVEN WHEN & THEN.

what is priority and severity in defect?

Defect Priority states the order in which a defect should be fixed. Higher the priority the sooner the defect should be resolved.Defects that leave the software system unusable are given higher priority over defects that cause a small functionality of the software to fail.

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.

what is most challenge defect u came across?

how to deal the production defects?

Ans: normally end user will report this issue.

we need to talk to them (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

\*\*\*test design techniques:

There are various types of designing techniques, each of which is suitable for identifying a particular type of error. Hence, the challenge lies in selecting the right set of relevant test design techniques for the particular application. Software test design techniques can be broadly classified into two major categories: static and dynamic techniques.

Static Test Design Techniques:

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

Static testing refers to the testing of software manually or with the help of tools. Static testing is usually carried out during the early phase of software development life cycle. Static testing is useful for testing multiple aspects of a software, including source code, functional and requirement specifications, and design documents and models.

Static testing can be further divided into two categories based upon whether it is performed manually or with the help of tools.

Dynamic testing:

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

Dynamic testing involves execution of the test object (application) on a computer. The input data is fed into the test object (application) and the program is executed. In dynamic testing, various variable quantities such as memory usage, response time, CPU usage and overall performance of the software are analyzed against the expected output. Dynamic testing is carried out during the validation process. Dynamic test design techniques can be further classified into:

Specification-based (black-box, also known as behavioral techniques)

Structure-based (white-box or structural techniques

Experience- based

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

how we learn the functionality of system?

what are the tools to manage defects/stories?

who will assign the work?

\*\*\*types of test metrics we use normally

Test Metrics are used to,

Take the decision for next phase of activities such as, estimate the cost & schedule of future projects.

Understand the kind of improvement required to success the project

Take decision on process or technology to be modified etc.

Importance of Software Testing Metrics:

As explained above, Test Metrics are the most important to measure the quality of the software.

what is traceability matrix?

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 testprotocols.

what are typical environments we have in projects

we maintain three environments: development environment, staging environment,production environment.

The meaning of development environment is intuitive: it's the environment used during development of the application.

what are different defect metrics and measurements we prepare

The number of defects is one of the measures used to measure test effectiveness. One of the side effects of the number of defects is that all bugs are not equal. So it becomes necessary to weight bugs according to their criticality level. If we are using the number of defects as the metric measurement the following are the issues:

The number o```f bugs that originally existed significantly impacts the number of bugs discovered, which in turns gives a wrong measure of the software quality.

All defects are not equal so defects should be numbered with a criticality level to get the right software quality measure.

What is staging environment

This is the release candidate, and this environment is normally a mirror of the production environment. The staging area contains the "next" version of the application and is used for final stress testing and client/manager approvals before going live.

what is development environment

Changes made by developers are deployed here so integration and features can be tested. This environment is rapidly updated and contains the most recent version of the application

 vending machine here r the test cases.

1. check wether the wending machine is working or not.

2.deciede which one to buy.

3.check for availability

4.check for the price.

5.insert the money in the slot.

6.press the product number.

7.collect the product.

8. collect the diffence in change .

what is QA environment

Environment for quality assurance; this provides a less frequently changed version of the application which testers can perform checks against. This allows reporting on a common revision so developers know whether particular issues found by testers has already been corrected in the development code.

what is production environment

production environment: This is the currently released version of the application, accessible to the client/end users. This version preferably does not change except for during scheduled releases.