1. Why testing is required?

Ans: For the effective performance of an application or a product

2) What types of application we test

Ans: Web based, Windows based, Mobile

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: Waterfall is a sequential(non-iterative) design process in which progress is seen through the phases of conception, initiation, analysis, design, construction, testing, production /implementation and maintenance. In this the outcome of one phase acts as a input to another phase.

Requirement Analysis 🡪 System Design 🡪 Implementation 🡪 Testing 🡪Deployment🡪 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. 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.



4)what is scrum methodology

Ans: Scrum is a subset of Agile. It is a lightweight process framework for agile development, and the most widely-used one.

Scrum has three roles: Product Owner, Scrum Master, and Team.

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.

We discuss:

1) What did you do yesterday?

2) What will you do today?

3) Are there any impediments in your way?

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

A user story is typically functionality that will be visible to end users. Developing it will usually involve a programmer and tester, perhaps a user interface designer or analyst, perhaps a database designer, or others. It goes to Product backlog.

A task, on the other hand, is typically something like code this, design that, create test data for such-and-such, automate that, and so on. These tend to be things done by one person. It goes to sprint backlog.

what is sprint planning and 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 ScrumMaster and the product owner should participate.

**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.

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

The **sprint backlog** is a list of tasks identified by the Scrum team to be completed during the Scrum **sprint**. During the **sprint** planning meeting, the team selects some number of **product backlog** items, usually in the form of user stories, and identifies the tasks necessary to complete each user story.

what is user acceptance criteria test cases

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.

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)** isthe 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.

* Requirement Analysis
* Test Planning
* Test Case Development
* Environment Setup
* Test Execution
* Test Cycle Closure

what is defect?

When actual result deviates from the expected result while testing a software application or product then it results into a defect. Hence, any deviation from the specification mentioned in the product functional specification document is a defect. In different organizations it’s called differently like bug, issue, incidents or problem.

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

Name

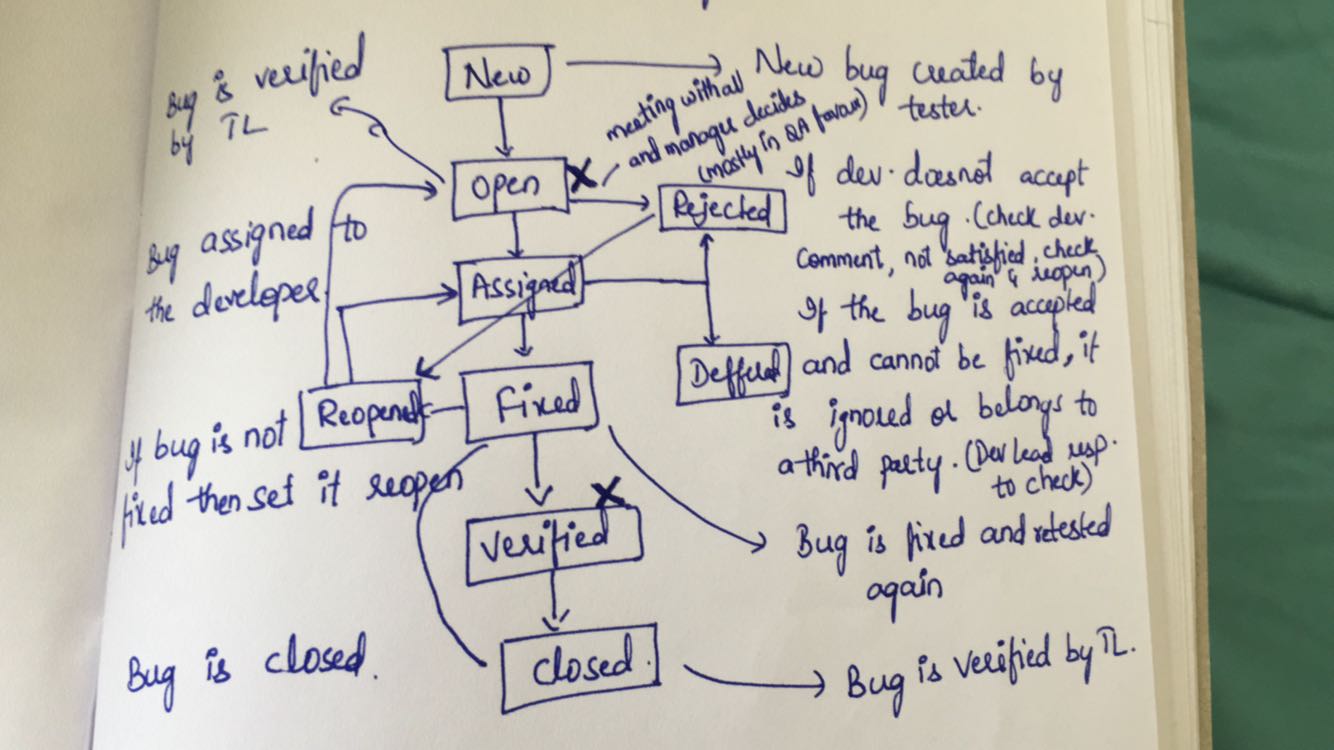
Description

Steps to replicate

Verion/build

Screenshots…etc

defect lifecycle

****

Different types of testing?

* Unit Testing
* Integration Testing
* Functional Testing
* System Testing
* Stress Testing
* Performance Testing
* Usability Testing
* Acceptance Testing
* Regression Testing
* Beta Testing

when do we use regression testing?

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

Regression testing is usually performed after verification of changes or new functionality. But this is not the case always. For the release taking months to complete, regression tests must be incorporated in the daily test cycle. For weekly releases regression tests can be performed when functional testing is over for the changes.

when do we use integration testing?

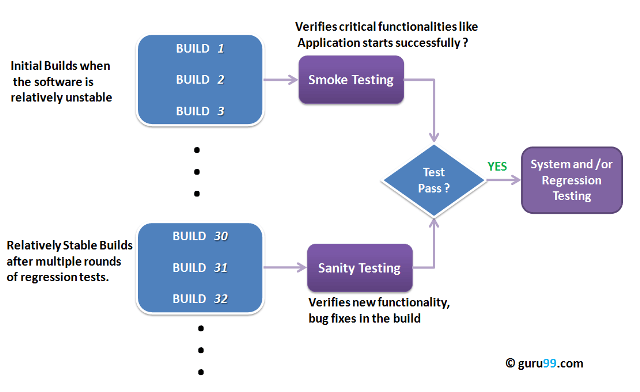
We normally do Integration [testing after “Unit testing”](http://www.softwaretestinghelp.com/unit-testing/).

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.

The main function or goal of Integration testing is to test the interfaces between the units/modules.

1. Bottom up approach(drivers)
2. Top down approach.(stubs)

when do we use smoke testing and sanity 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.

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.*

User acceptance testing (UAT), otherwise known as *Beta, Application*, or *End-User Testing*, is often considered the last phase in the web development process, the one before final installation of the software on the client site, or final distribution of it.

UAT is usage of the software by people from the intended audience and recording and correcting of any defects which are discovered. It’s the closest thing to a “\_real world\_” test available. It gives users the chance to interact with the software and find out if everything works as it should if features have been overlooked, miscommunicated, not communicated, and so on.

**what is alpha and beta 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 blackbox and whitebox techniques. The aim is to carry out the tasks that a typical user might perform. Alpha testing is carried out in a lab environment and usually the testers are internal employees of the organization. To put it as simple as possible, this kind of testing is called alpha only because it is done early on, near the end of the development of the software, and before 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.

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

The Differences Between [Black Box Testing](http://softwaretestingfundamentals.com/black-box-testing/) and [White Box Testing](http://softwaretestingfundamentals.com/white-box-testing/) are listed below.

|  |  |  |
| --- | --- | --- |
| **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 |

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

Use risk analysis to determine where testing should be focused.

Considerations can include:

* 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 functionalities?
* Which tests will have the best high-risk-coverage to time-required ratio?

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

In such situation, Firstly, I with my team members, will try to find out the cause for it. And with discussion with developers also, we will try to fix that out. And will work extra hours to get it fixed, and do testing again.  
  
And if the defect will take time to be resolved, then we will talk to Project Manager, that we can't send this release, as it has this defect, and we are working on it.  
  
Its better to send the release late, but it should be bug free. Whats the fun, if we send the release on time, but it has many defects, that the customer is not satisfied.

One thing we have to do is measure the severity and frequency of that defect .  
  
If it is having High Severity and low frequency of occurance then we can make the bug as known issue and move it to maintainance phase.  
  
If it is having high severity and high occurence then we have to fix the bug immediately and test.

when do we use automation testing?

If you need to run the test cases a lot number of times in a test cycle –

if the number of test cases is high, and if these will be run more than once then automation testing can provide you with better results. –

some features have higher chances of failing than the others. Such high priority feature is better to be tested with an automation testing tool –

If we need need to run the test cases in a predetermined order

If the test case need to be updated constantly

If you are planning on simultaneous running of test cases

If you need to need to test single functionality with multiple data sets

what tester will do in each phase of 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 its fixed by the developer the tester performs a regression testing

difference between load and performance testing?

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

* 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 especification of the application - the test is to verify or confirm that the application will work at the planned performance;

different types of non-functional testing types?

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

what is test case?

A **test case** is a set of conditions or variables under which a tester will determine whether a system under **test** satisfies requirements or works correctly. The process of developing **test cases** can also help find problems in the requirements or design of an application.

what is test plan/test strategy document

Ans: Test plan document contains different section like

Types of testing :

Exit and Entry criteria :

Criteria comes in every phase of testing( Static & Dynamic testing) They are 4 types of criteria. 1. Enter Criteria 2. Suspension Criteria 3. Resuspension Criteria & 4. Exit Criteria 1. Enter criteria: This is the first stage and starting of the Phases2. Suspension Criteria: when your sending the document to verify by developer& Test lead.at that time Suspension Criteria comes screen3. Resuspension Criteria: there is any changes want to done in your document . at that time Resuspension Criteria comes on Screen.4. Exit Criteria: You completed your testing .at that time Exit Criteria comes on Screen

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.

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.

* BDD focuses on the behavioural aspect of the system rather unlike the TDD focuses on the implementation aspect of the system.

what is priority and severity in defect?

The **priority** status is set based on the customer requirements. While **Severity** is the extent to which the**defect** can affect the software.

how to estimate test cases?

1. **Think of Some Buffer Time**
2. **Consider the Bug Cycle**
3. **Availability of All the Resources for Estimated Period**
4. **Can We Do Parallel Testing?**
5. **Estimations Can Go Wrong – So re-visit the estimations frequently in initial stages before you commit it.**
6. **Think of Your Past Experience to Make Judgments!**
7. **Consider the Scope of Project**
8. **Are You Going to Perform Load Testing?**
9. **Do You Know Your Team?**

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 and reproduce the issue with test logins

Create defect in defect tool under the production release version

developers will fix the issue

we (QA) test the issue on production version code 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 review steps

**What do we review**?  – Everything created has to be reviewed. The following are some of the common artifacts reviewed:

1. Test plan
2. Test scenarios
3. Test templates
4. Test cases
5. Test data
6. Reports…etc

**Why to review?** – For exactly the same reason we test the software, example:

1. To uncover errors
2. To check for completeness
3. To make sure the standards and guidelines are adhered to or not …etc.

**How to review? –**The following are the list of activities involved:

1. Define the criteria – Have a checklist of what to look for?
2. Perform the check
3. Record your results
4. Share, discuss and implement the changes required
5. Version control the documents involved
6. Sign off and use the doc as intended.

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

We do Adhoc testing

how we learn the functionality of system?

what are the tools to manage defects/stories?

Bugzilla

Jira

HP ALM

IBM Clear Quest

who will assign the work?

Testing Team Lead

types of test metrics we use normally

* **Process Metrics:** It can be used to improve the process efficiency of the SDLC ( Software Development Life Cycle)
* **Product Metrics:** It deals with the quality of the software product
* **Project Metrics:** It can be used to measure the efficiency of a project team or any tools being used by the team members

Manual test metrics is classified into two classes

* **Base Metrics**
* **Calculated Metrics**

what is traceability matrix?

A **testing** requirements **traceability matrix** is a document that traces and maps user requirements, usually requirement IDs from a requirement specification document, with the test case IDs. ... The **traceability matrix** document is prepared to show clients that the coverage is complete.

what are typical environments we have in projects

what is development environment

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**.

what is QA environment

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.

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.

what are different defect metrics and measurements we prepare

what are weakness and strong points

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.

how to deal the production defects?

The best thing you can do is learn from it and prevent the same in the future. I write an automated test-case for each defect found in production, since these are the brittle parts of the application.

Second I would plan a [root cause analyses](https://en.wikipedia.org/wiki/Root_cause_analysis) session and use the [5 whys](https://en.wikipedia.org/wiki/5_Whys) technique to find the cause. Now find a solution to improve here and make it future proof.