**Quality Assurance:** is focusing on preventing defects [ Verification, Product Right, start of development process, Low level activity]

**Quality Control:** is focusing on identifying the defects [ Validation, Right Product, end of the development process, High level activity].

**What is software testing?**

Is process which includes set of activities to make sure s/w application is working properly according to the client’s requirement.

**Why software testing is necessary?**

Software testing is important because s/w bugs could be expensive or human loss.

**Severity in defect** defines the **impact** that a given defect on the application.

**Priority in defect** defines the order in which it should be resolved.

**What is automation Testing?**

Is a process of automating the existing manual process to test application using an automation tools to find the defects, in this process executing the test scripts and generating results are performed automatically by automation tools, the most popular automation tools are HP QTP, Selenium WebDriver, etc.

Benefits of automation testing

* Saves time, money and faster execution
* Code reusability
* Easy reporting
* Enable parallel execution -combination of different o/s and browsers
* More reliable
* Mostly used for Regression, Smoke and Sanity testing
* Minimal manual intervention
* Maximum coverage

**What is Test Plan?**

Test plan is detailed document which consist of what to test, when to test, how to test.

Test plan ID – unique number assigned.

Test plan description – use of the test plan/purpose if the test plan.

Test items – details of functionalities/features [how many to features].

Features to be tested.

Features not to be tested.

Entry Criteria.

Exit Criteria.

Suspension Criteria.

Roles and Responsibilities.

Staff and training needs

Test Approach

Risk and Assumptions

**What is Test strategy?** [ prepared by Project Manager]

Is a **High level document** which defines the **approach for the software testing**, It is basically derived from **BRS**. It is static document which sets the standards for testing so not updated often

.

**What are Test Cases?**

A test case is a set of actions executed to verify a particular feature/functionality of software application.

A process of developing a **test case** can help find the problems in the requirement or design or an application.

Once requirements are finalized/freezed we will create Test cases.

**What is the process of creating a test script?**

Step 1: **Understanding the requirement** document thoroughly. [ in absence of document, we could understand any point of reference we have in previous version of application or wireframe]

Step 2: after understanding the requirement we make a list of **what areas** **to be tested** in the s/w. The **outcome of this step** is **list of test scenarios**.

Step 3: once we have test scenarios, we concentrate on **how to test**. This phase involves writing detailed steps about a particular feature, test data and expected result.

Once these three steps are done, we are ready for testing.

**What is the importance of the Test Data and how do you prepare it?**

Test Data is data that is used to execute the tests on software application. The test data needs to be precise and exhaustive to uncover the defects.

We prepare Test Data from SRS and usually provided by the Developer.

**Testing Techniques [ Risk based testing]**

1. **Boundary Value Analysis (BVA)**

min-1 min+1 middle max-1 max+1 [ Test data]

1. **Equivalence Class Portioning [ ECP]**

Valid Invalid [Test data]

1. **Error guessing [random]**

**7 principles of s/w testing**

1. **Testing shows presence of bugs** [ without testing we cannot find bug]
2. **Exhaustive testing is not possible** [ testing in various combination is not possible]
3. **Early testing** [ testing should involve in analysis stage]
4. **Defect clustering** [ where more defects that area should be retested, regression testing]
5. **Pesticide paradox** [if no defect found we need amend the Test Case]
6. **Testing is context dependent** [ depends upon s/w eg: security, performance]
7. **Absence of error is fallacy** [ s/w cannot guaranteed 100% bug free]

**What will you do if you find the defect?**

Once the defect is found we **log it as a new defect with all details.**

*Details provided are:*

Tile.

Description.

Expected Result.

Actual Result.

Project details.

Environment details.

Steps to reproduce.

Type of defect – functional, Integration, database.

Priority of the defect.

Severity of the defect.

Attachments/screenshots.

Test Data

**What is the cost of the defect?**

The cost of the defect is measured by the impact of the defects and when the defect was found. If the defect is found in requirement gathering it is cheap to fix it. The correction of specifications can be done, and it can be resumed. If the defect is found in design, then it can be corrected and resumed. If defect is found in User acceptance, then the cost of fixing the bug will expensive.

**What is Smoke test and Sanity test?**

Smoke test is to confirm whether the build is stable or not. It confirms to test further or not.

Sanity test is similar to smoke test but is performed on production environment before the end users start using the product.

**What is Regression testing and Retesting?**

Regression testing is carried out to ensure that the existing functionality is working fine and there are no side effects of any changes or enhancement done to the application.

Retesting is carried out to ensure that the particular defect has been fixed and its functionality working as expected.

**What type of test you have automated?**

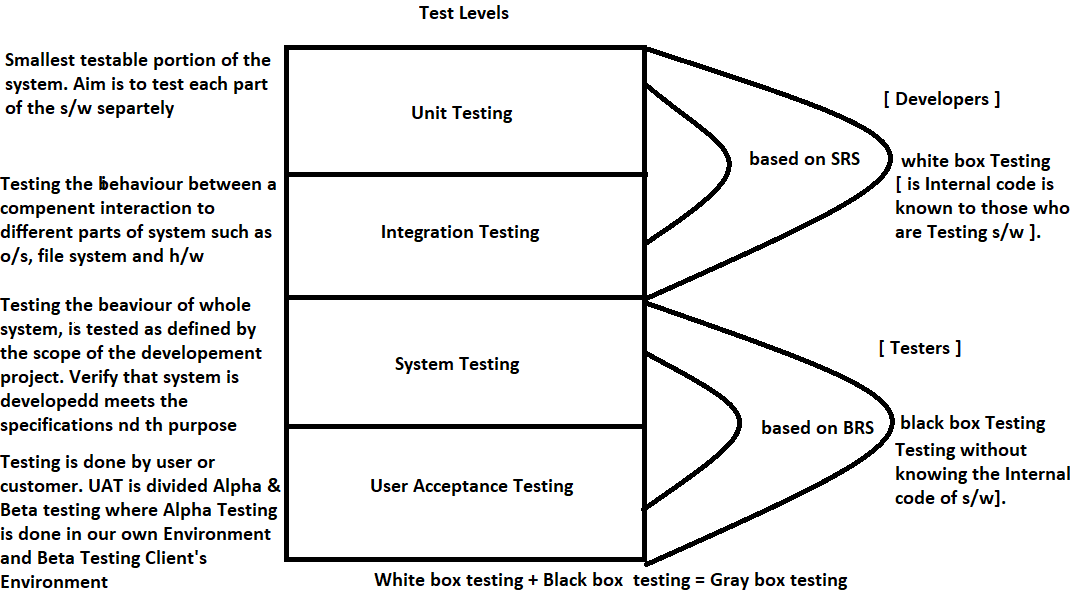
Our main **focus** is to **automate test cases** to do **Regression** testing**, Smoke** testing **and Sanity** testing.

**What is a dynamic testing technique? [ It is done during validation process. Software tested by executing on computer].**

* **Unit Testing**: A Unit testing is the smallest testable part of an application. It is done by Developers.
* **Integration Testing:** Testing the Integrations/interfaces between a component interaction to different parts of the system such as o/s, file system and h/w. done by Developers.
* **System Testing:** Testing the behavior of the whole system (or) application, is tested as defined by the scope of the development project. Referred as final test to verify that system is developed meets the specifications and the purpose.

System Testing involves both functional and non- functional requirement of testing.

* **User Acceptance Testing:** Requirements are met, this testing is done by user or customer.



**Functional testing**: testing **functions** of **component/system** described in **requirement specifications.**

[ It answers questions “can a user can do this”].

**Non-Functional testing**: is the testing of the **quality characteristic of a component/system**. Includes reliability testing, usability testing,etc

**Maintainability testing:** defines how easily it can be maintained.

**Efficiency testing:** amount of code, testing recourses required by program to perform particular function, number of test cases executed divided by unit of time.

**Portability testing:** to determine how easily the application moved from one environment to other, results are measured in terms of time and documentation is updated.

**Baseline testing:** refers to validation of documentation and specifications on which test cases would be designed.

**Compliance testing:** it is related to IT standards followed by the company.

**Endurance testing:** testing system with significant load over extended period of time to verify how system behaves.

**Load testing:** system’s performance is as expected under normal and expected condition [ different users access at same time].

**Performance testing:** to determine overall performance of system [ speed, response time ].

**Security testing:** to ensure no loopholes which could lead to any data loss and threat, including testing authentication, authorization, integrity and availability.

**Volume testing:** to verify the behavior of application when large volume of data is involved [eg: generate a report from large databases and report printed correctly]

**Recovery testing:** evaluates that the system terminates in case of any failure and the data is recovery appropriately from h/w or s/w failure.

**Failover testing:** if system fails then creating a backup system to get the system back.

**Localization and Internationalization testing:** verify the application works on different languages and international settings.

**What will happen if you don’t have time for testing?**

We do ad hoc testing, testing the application randomly without any requirements, planning and documentation, is also called Monkey testing.

**Tell me how you decide priority and severity?**

It depends upon the defect which is found and at what stage it was found.

If the defect has **impact on application crashing** or **unable to test further**. It will be reported as critical and priority will be high.

For eg:- if a application crashes when a remote link is clicked, if the clicking a remote is very rare by the user but impact of the application crashing is severe, then the severity is high but priority is low.

**Severity types**

**Critical** [if there is no acceptable alternative method to achieve required results, termination]

**Major** [ if there is alternative available to achieve required results, termination]

**Moderate** [ if the defect does not result in termination but s/w produces incorrect results]

**Minor** [ if defect does not result in termination and desired results achieved easily]

**Cosmetic** [ look and feel of application]

**Priority types**

**High** [ need to fix immediately]

**Medium** [ not to fix immediately to but if time permits]

**Low** [ can be fixed in next phase/sprint]