**Manual Testing**

1. What is Software Testing?

Software Testing is a part of our software development process and it used for identify the defects in the software and the testing main objective Is to release a quality product to the client./

1. Software Quantity?

The quality is defined a justification for all the requirements of a customer in a product quality is not defined a product it will defined the customers mind.

Ex: Bug free, delivered on-time , budget, meet requirements as per expectations, maintainable

1. Product Vs Project?

**Project**: A software application developed for a specific customer based on the requirements is called Project.

**Product**: A software application developed for multiple customers based on the market requirements is called Product.

1. Why do we need Testing?

* Bug free
* The system meets software specifications as per customer requirements
* It meets a end user expectations
* Fixing the bugs before release the software

Ex: error, Defect/Bug, failure

1. Why the software has bugs normally?

* Miscommunication or no Communication
* Software complexity
* Programming errors ( compile errors)
* Changing requirements
* Lack of skilled testers

1. Software Development Life Cycle (SDLC)?

The software development life cycle is a process to follow the software industries the process shown below :

1. Requirements Analysis
2. Design
3. Development
4. Testing
5. Maintenance

7. Waterfall Model (Advantages & Dis Advantages) ?

**Advantages:**

* Quality of product is good
* If the client change a previous requirements in the software application is not allowed
* Chances of finding the bugs is less
* Investment is less on initial stage later the tester will be need
* Recommended for small projects

**Dis Advantages:**

* Requirement changes are not allowed
* If there is a defect in requirement that will continued in next phases
* Time investment is more it leads to high investment
* Testing will starts after development (coding).

8.Why Testing is Necessary?

the software applications is using any level of testing will cannot declare that there is no defect in the product

**Development:**

* The developer will always work as per client requirements
* But each level of software building (including system level) to access whether it is really works or not.
* Integration issues will caused when different units do not work together, even working independently
* To bring a individual units together and make a final product, and some defects may be possible in the source code when developers are working with different places.

**Testing Process:**

* Testing is a process of demonstrating that errors are not present in the product this approach is used in acceptance testing where if the application meets acceptance criteria, then it must be accepted by the customer.
* Testing is gives number of detects present which indirectly gives a measurement of software quality.
* More number of defects indicate bad software and bad processes of development

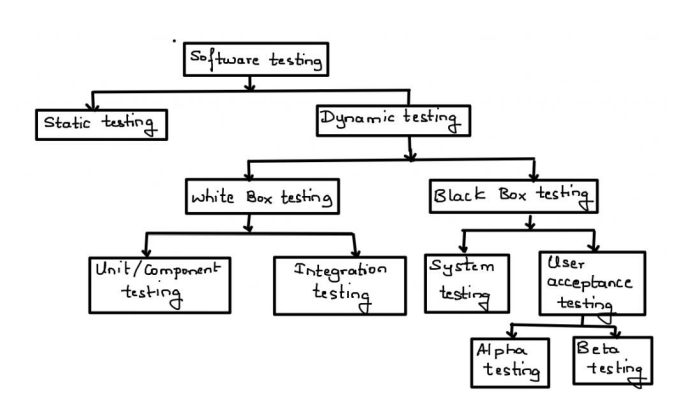
**Software Testing:**

* it is process to test an application to find out error in it
* checking the software is ok.
* The goal of software tester to find bug
* Verifying and validating that a software or application is bug free

**Testing Types:**

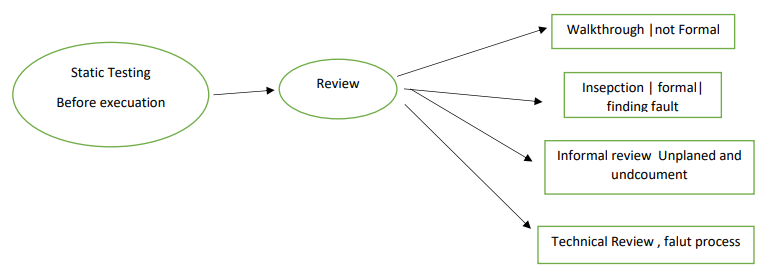
* **Manual Testing:** Manual testing is testing a software manually without using as tool or any script
* **Automation Testing:** When the tester writes scripts and uses another software to test the product. This process involves automation of a manual process.

Software testing basically two types static testing and dynamic testing



**Static Testing Techniques:**

* Analysis of a program carried out without executing the program
* Done during verification process before development
* As we know 85% errors found in design phase
* is code is tested in static testing ? “No” || the documentation is tested
* software development starts , continue and ends with documentation
* early documentation is used to define the software to be built
* later documentation covers the software training, installation, and operation (user guides).
* The primary goal of static testing is reduce defect by reducing defects in the documentation from which the software is developed



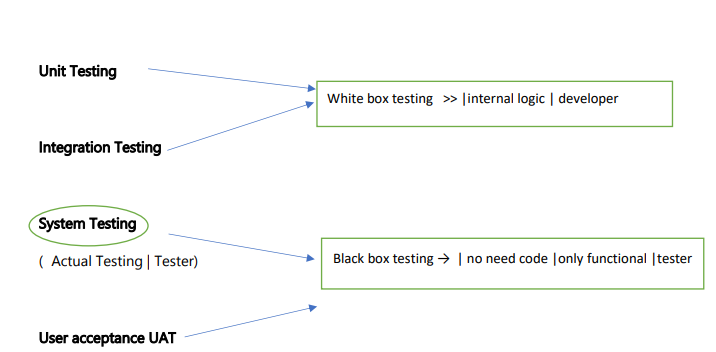
**Dynamic Testing Techniques:**

* **Unit Testing:** is unit testing individual component of software tested. The purpose of this testing is each module is working properly it focus on the smallest unit of software design

Ex: in a program we are checking if loop, method or function is working fine

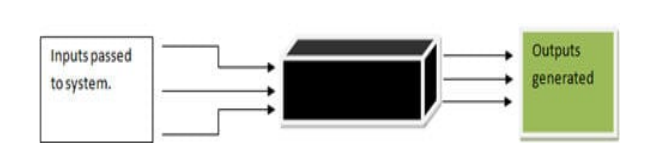
* **Integration Testing**: in integration testing individual units are combined and tested as group (developer) like (i) Top-down, (ii) Bottom-up (iii) Sandwich (iv) Big-Bang the main purpose of integration testing to check modules are communicating each other as DFD Data Flow Diagram which is specified in TDD (Technical Document Diagram).
* **System Testing**: In this testing we can test whole application (complete/integrated software is tested) done by tester.
* **Acceptance Testing**: a level of software Testing in which software is tested for user acceptance UAT done at client location where software is actually used **Alpha Testing:** done by tester in company in presence of customer

**Beta Testing**: done by customer to check software is ok, satisfy requirement



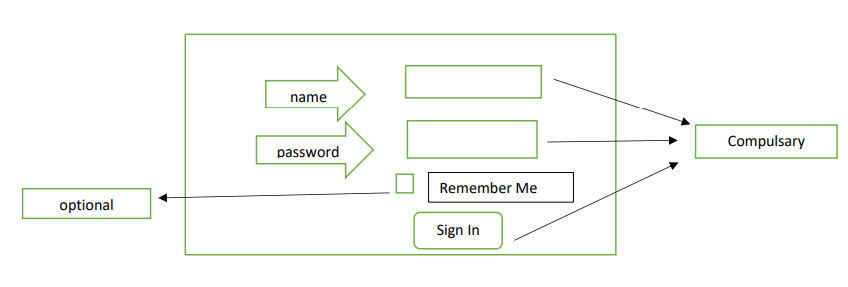
**Testing Types:**

* **Functional Testing:** is the process through which QA’s determine if a piece of software is acting in accordance with pre-determined requirements. It is uses black-box testing techniques, in which the tester has no knowledge of the internal system logic. **Functional testing is only concerned with validating if a system works as intends** Functional testing is a type of black box testing “does this actually work?” The ultimate of functional testing is to ensure that software works according to specifications and user expectations like input values , test cases, compare actual and expected output

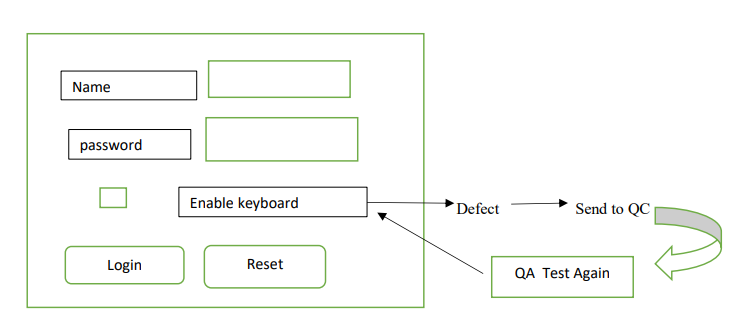


Ex. Login functionality, registration functionality

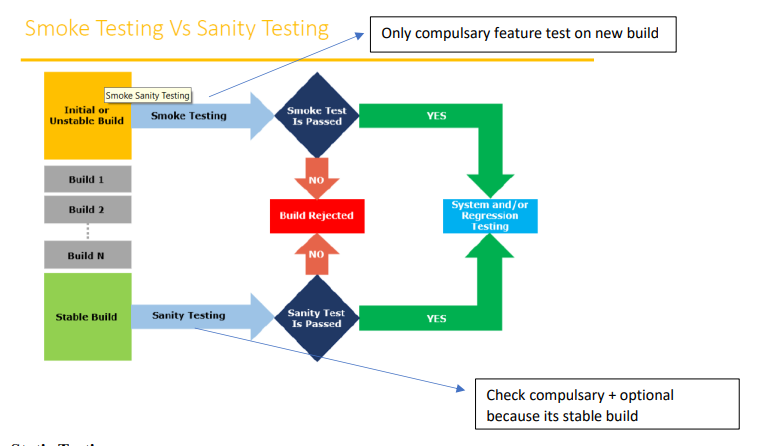
* **Non Functional Testing:** Load testing, Reliability, The readiness of a system, usability testing Ex: checking how many people can simultaneous basket
* **Black box testing:** Monitoring internal structure check internal logic. Done by developer
* **Smoke testing:** testing on newly released build compulsory requirement it is first testing on newly released build (Build Verification Testi check the deployment software build is stable or not

****

* **Sanity Testing:** (Testing on newly released build then check compulsory + optional) in the above fig all field are compulsory then it is sanity testing
* **Retesting:** Testing functionality once again

****

* **Regression Testing (Re-running if code changes):** it is overall testing whenever new change is occurred re-running functional + nonfunctional test code change , does not impact existing functions after changes in the program. This type of testing makes sure that the whole component works properly even after adding components to the complete program.

****

**Static Testing:** The basic process of testing is Review, Inspection, Walkthrough Test software without executing software (just test documents) This test done for avoid bugs in early stage (review testing ) look and feel why static testing **early defect detection and correction**, **to get fewer defect at a later state of testing**

**Static Technique has 3 types:**

**Review:** review before development i.e simple document read the document s/b correct or not document s/b correct and complete requirement review , design review , Test plan review, review testing review testing can do anybody, manger | developer | tester |coworker etc.

**Walkthrough:** it is informal , anytime, not planned, done when ever required author of document will explain to their team

**Inspection:** Most formal , 3-8 people in meeting Proper meeting, Schedule which is intimated by mail

**Dynamic Testing:** The main purpose of dynamic testing is to test software behavior with dynamic variables dynamic testing requires code to be executed. Static testing -> just analyze the code, no need execution

* **Alpha Testing :** Its is final testing in development Advantage : immediate solution is possible
* **Beta Testing :** it is 1st testing in client side . it is also called user user acceptance testing UAT Disadvantage : no immediate solution if defect is found
* **Installation Testing :** providing required resources at client location It is type of testing in which test engineer check deployment process is successful as per user guideline Deployment document /user manual : it is document prepared by project manager
* **Usability Testing :** checking application for user friendliness
* **Monkey Testing** : used for game testing, used for random input To check the application or system will crash
* **Portability Testing** : Developed application Should support multiple environment
* **Forced error Testing :** to check valid error message will display
* **Exploratory Testing** : When test engineer does not have idea of functional testing then he is learning through exploring application
* **End to End Testing** : We can check all internal component for successful response Internal component like Client , Network, Server Database etc are working fine Means Testing internal component
* **Security Testing** : Checking Security of application
* **Reliability Testing** : The Developed application Should Support Longer Duration i.e. Stability
* **Audit** : it is independent evolution of software .
* **Inspection** : it is formal evolution of software
* **Concurrency Testing** : multiuser Testing
* **Debugging** : executing program line by line for finding errors.

**Some of the most popular SDLC models are:** no matter which module is used each has same phases

* Waterfall Model
* V- Shape Model
* Incremental Life Cycle Model
* Spiral Model

SDLC : Software development life cycle it is process used by software company to develop, test software

1. Requirement analysis

2. Design (blue print)

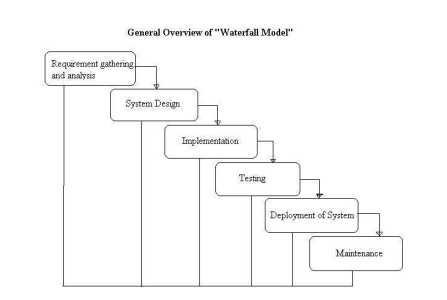
3. Coding or development

4. Testing

5. Maintenance

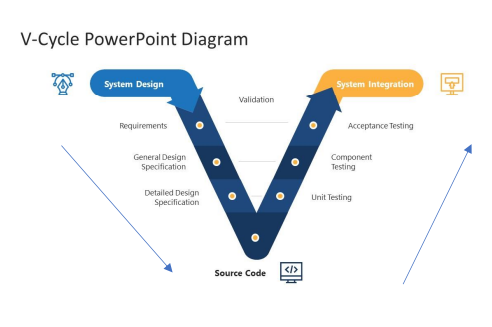
**Waterfall Module:** it is old traditional model it is linear model i.e steps 1 after another Each phase must complete to start new phase (i.e called one after another)

* In waterfall module quality of product is food because every phase has clear Documentation
* SRS (System requirement specification) not changed hence no bug
* Initial investment is less because no tester involved
* No change in middle
* Testing will start after coding



**V shaped Model:** Verification and validation verification -> done before development check we are doing correct -> verifying document -> because no software ready -> verification =before s/w = static -> review ->walkthrough -> inspection-> static testing = verifying doc

Validation -> testing software -> done after software ready -> done right ->validation = after s/w = dynamic -> dynamic testing = unit -> integration - > system -> user acceptance



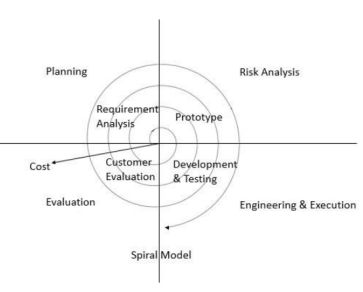
System design = before development = verification = static testing = verify documents

System integration=after devep = validation=dynamic testing = unit |integration|system|UAT in V model = testing is involved every phase

Disadvantage : more documents

**Spiral Model:**

* Spiral Model is iterative model.
* Spiral Model overcome drawbacks of Waterfall model
* We follow spiral model whenever there is dependency on the modules
* In every cycle new software will be released to customer
* Software will be released in multiple versions so it is also called version control model.



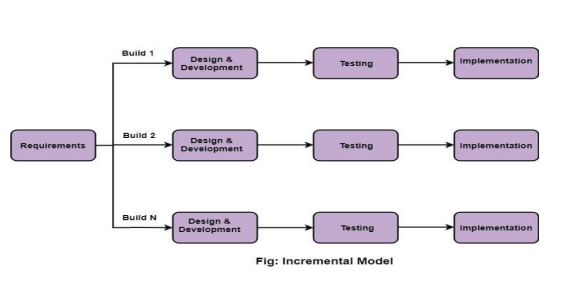
Advantages:

* Testing is done in every cycle , before going to the next cycle.
* Customer will get to use the software for every module.
* Requirement changes are allowed after every cycle before going to the next cycle

Disadvantages:

* Requirement changes are NOT allowed in between the cycle
* Every cycle of spiral model looks like waterfall model.
* There is no testing in requirements & design phase.

**Increment Model**: requirement are divided into multiple module each module goes through SDLC phase i.e analysis, design ,coding, testing, maintenance Requirement -> module 1 + module 2 +…module n



When use incremental model: A Project has lengthy development schedule. When the requirement are superior.

Disadvantages: Need for good planning, Total cost is high