**SOFTWARE TESTING DOCUMENT**

**Software Testing** : It is a process used to evaluate and verifying, functionality and performance of software products or application before its release.

How to do software testing they are:

1. Requirement Analysis
2. Test Planning
3. Test Design
4. Test Environment
5. Test Execution
6. Defect Tracking
7. Test Reporting

**Objectives of Software Testing:**

1. **Detect Defects :**  Identify bugs or errors in the software before it is released to users.
2. **Ensure Quality :** Verify that the software meets quality standards and delivers a good user experience.
3. **Validate Requirements :** Confirm that the software meets the business and technical requirements specified.
4. **Prevent Issues :** Proactively identify potential problems to avoid future failures.
5. **Improve Performance :** Optimize the efficiency and speed of the software.

**Software Quality**: It refers to the degree to which a software product meets specified requirements, user expectations, and industry standards. It ensures the software performs reliably, efficiently, and securely while being maintainable and user-friendly.

There 5 views in the software Quality, they are TRANSCENDENT BASED, PRODUCT BASED, USER BASED, DEVELOPMENT & MANUFACTURE BASED, VALUE BASED.

In this there are types:

1. **Functionality:** The software's ability to provide the required features and functions correctly.
2. **Reliability:** The software's ability to operate consistently without failure under specified conditions.
3. **Performance Efficiency:** The software's responsiveness and resource utilization under various conditions.
4. **Usability:** Ease with which users can learn and operate the software.
5. **Maintainability:** Ease of modifying the software to fix bugs or make enhancements.
6. **Security:** The software's ability to protect data and operations from unauthorized access.
7. **Compatibility:** The ability of the software to function across various environments and devices.
8. **Portability:** Ease of transferring the software to different environments.

**Quality Software is reasonable are:**

1. Bug free
2. Delivered in time
3. Within Budget
4. Meets requirements (or) expectations
5. Maintainable

**Product vs Project**

**Product:** A product is an ongoing offering designed to meet customer needs and software application is developed for a specific customer based on the requirement then it is called PRODUCT.

**Project:**  A project is a temporary endeavor with a specific goal and timeframe and it is developed for the multiple customers based on market requirements then it is called PROJECT.

**Why do we need Testing**

**Software testing** is essential for ensuring the quality, functionality, and reliability of software applications. And ensure the bug free, customers requirements &software specifications and fixing the bugs identified after release is more expensive.

There are types in these they are:

1. Error
2. Defect/Bugs
3. Failure

**Why the software has bugs normally**

Software bugs are a natural occurrence in the development process due to the complexity of creating and maintaining software systems. They can arise from various factors, including human errors, system limitations, and unforeseen interactions within the software.

There are types in this:

1. Human Error
2. Complexity of Software
3. Changing Requirements
4. Inadequate Communication
5. Software Integration Issues
6. Third-party Dependencies
7. Insufficient Testing
8. Environment and Configuration Issues
9. Performance Constraints
10. Hardware Limitations
11. Time Constraints
12. Lack of Documentation

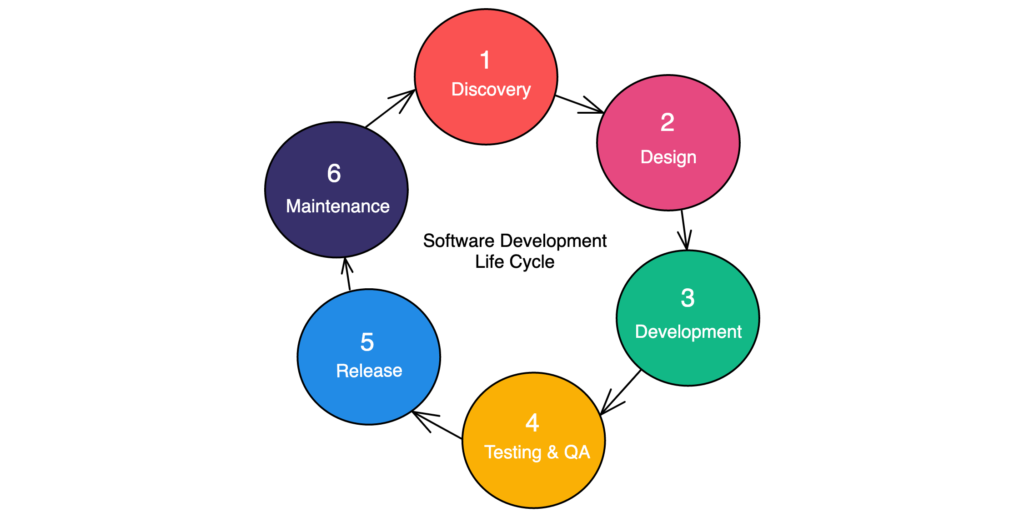
### ****How to Minimize Bugs****

1. **Clear Communication**: Ensure proper understanding of requirements through documentation and collaboration.
2. **Thorough Testing**: Implement unit testing, integration testing, system testing, and user acceptance testing.
3. **Code Reviews**: Regularly review code for errors and adherence to standards.
4. **Version Control**: Use tools to manage code changes and track bugs.
5. **Proactive Debugging**: Use debugging tools and techniques to identify and resolve issues during development.

**Software Development Life Cycle (SDLC)**

It is a structured process used to design, develop, test, and deploy software applications. It provides a systematic approach to software development, ensuring high-quality products that meet user requirements. The SDLC typically consists of a series of well-defined stages, each with specific goals and deliverables.

**SDLC DIAGRAM**



**Waterfall Models (Advantages and Disadvantages)**

The waterfall model is a linear and sequential approach to software development where each phase must be completed before the next begins. It's one of the earliest models used in software development and is straightforward in structure.

### ****Advantages of the Waterfall Model****

1. **Simple and Easy to Understand**
   * The sequential nature makes it easy to follow and manage, especially for smaller projects.
   * Each phase has clearly defined goals and deliverables.
2. **Structured Approach**
   * The rigid phase-by-phase structure ensures systematic progress, reducing confusion during development.
3. **Well-Documented Process**
   * Extensive documentation at each stage helps maintain clarity and serves as a reference for future projects.
4. **Ideal for Smaller Projects**
   * Suitable for projects with well-defined requirements that are unlikely to change.
5. **Clear Milestones**
   * Each phase has distinct milestones, making it easy to track progress and manage schedules.
6. **Testing Happens in Isolation**
   * Issues from previous phases are addressed before moving forward, ensuring that each phase is completed correctly.

### ****Disadvantages of the Waterfall Model****

1. **Inflexibility to Changes**
   * Changes in requirements or scope are difficult and costly to incorporate after the project begins.
2. **Late Testing Phase**
   * Testing occurs only after the implementation phase, making it harder to identify and address issues early in the development cycle.
3. **Unsuitable for Complex or Long-Term Projects**
   * The model doesn’t handle evolving requirements well, which is common in complex or long-term projects.
4. **High Risk**
   * If errors or misinterpretations occur early in the process (e.g., during requirements gathering), they may not be discovered until late stages, leading to significant rework.
5. **No Feedback During Development**
   * End-users or stakeholders don’t interact with the system until it's fully developed, which may result in a product that doesn’t meet their expectations.
6. **Resource Inefficiency**
   * Resources may be idle during specific phases

### ****When to Use the Waterfall Model****

* Projects with **well-defined requirements** that are unlikely to change.
* Smaller projects with clear goals and limited scope.
* Projects where extensive documentation is required.
* Systems requiring strict regulatory compliance, like defense or medical applications.

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| Simple and structured approach | Inflexible to changes |
| Clear documentation | Testing happens late |
| Easy to manage progress | High risk in evolving projects |
| High risk in evolving projects | Limited stakeholder feedback |

**Why Testing Is Necessary**

It helps identify errors, & improve product functionality, and avoid risks and it helps the team in understanding actual and expected outcomes so that they can improve the quality of their products.

**Types of testing**

1. **Integration testing**
2. **Unit testing**
3. **Functional testing**
4. **Security testing**
5. **Performance testing**
6. **Regression testing**
7. **Stress testing**
8. **Acceptance testing**

**TYPES OF TESTING**

**There two of testing they are:**

1. **Manual Testing**
2. **Automation Testing**

**Manual Testing: It is** a software testing method where a tester manually uses a software application to identify defects and issues. And without using any automated tool or any script.

There are types in manual testing they are

1. Black box
2. White box
3. Grey box
4. Acceptance box
5. Integration box

**Automation Testing:** It is the practice of using software tools and scripts to automate the execution of tests in software development and quality assurance processes.

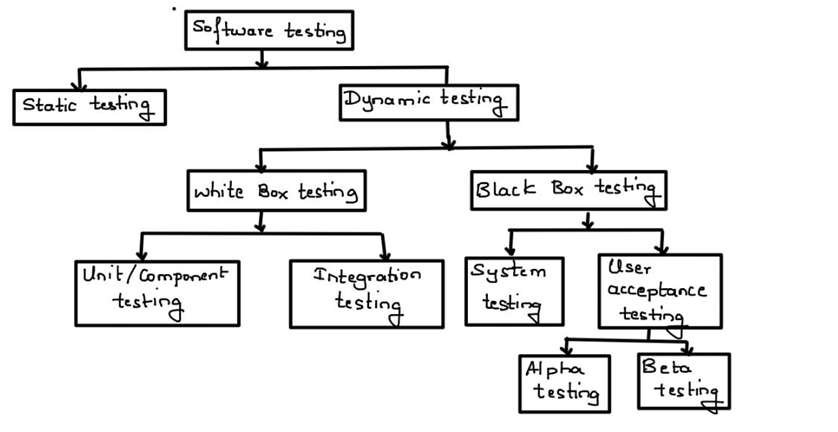
**Static testing:** A static analysis can only be performed if the system being simulated does not depend on time, and if the loads being applied are constant. And it will analyse the code requirements documents & design documents. We don’t executed the code.

**Dynamic testing:**  It is a software testing method that involves evaluating a software system's runtime behavior by providing input and validating the output. We will executed the code.

**Types of dynamic testing**

Some types of dynamic testing include unit testing, integration testing, system testing, acceptance testing, functional testing, regression testing, performance testing, and usability testing.





**Reviews:** It is a software testing technique that helps to identify and remove the errors in a software design and documentation.

**Types**

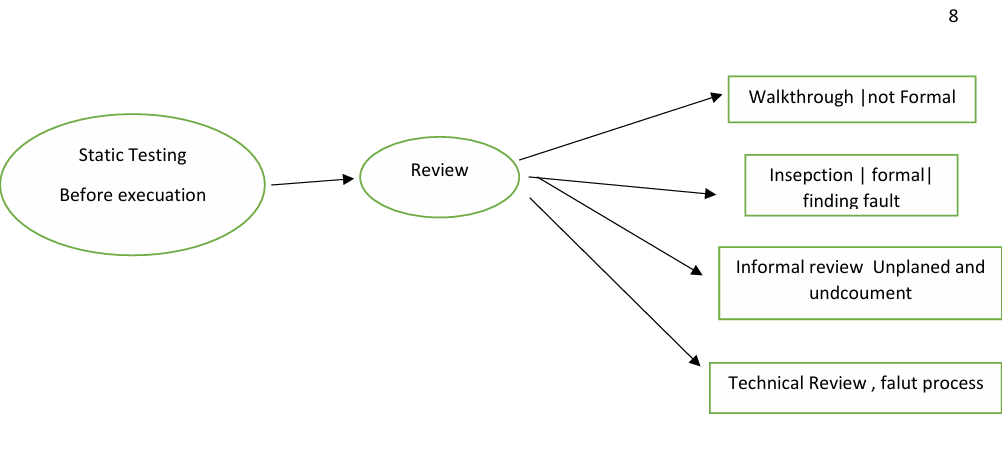
There are several types of reviews, including:

**Informal**: A casual review where co-workers provide informal feedback on documents.

**Walkthrough**: The author explains the document to the team, who can ask questions and take notes & involves the author of a document guiding participants through the document

**Inspection**:  A review that's usually led by a moderator and discussing with the group members.

**Technical or peer review**: Peers review technical specifications to find errors.



**Dynamic testing techniques:** It isused to evaluate the behavior of software during runtime.

**There are types in that they are:**

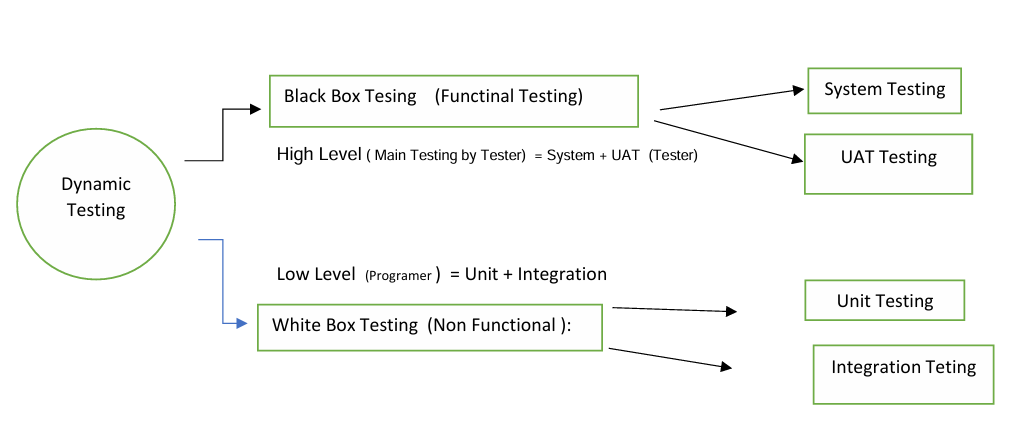
**Black-box testing:** In Tests the software's functionality from the end-user perspective, without looking at its internal workings. In this the code will not be visible. It is high level.

**White-box testing**

Tests the software's internal structures and workings, such as specific functions, methods, or code snippets. Testing each line of the code is know as white box testing. In this the code will be visible. Developers will use this techniques & some techniques like path testing, code coverage. It is low level.

**Grey-box testing**

Combines black-box and white-box testing, where the tester has some knowledge of the software's internal workings.



**Levels of testing:**

There are 4 types of testing

1. Unit testing
2. System testing
3. Integration testing
4. Acceptance testing

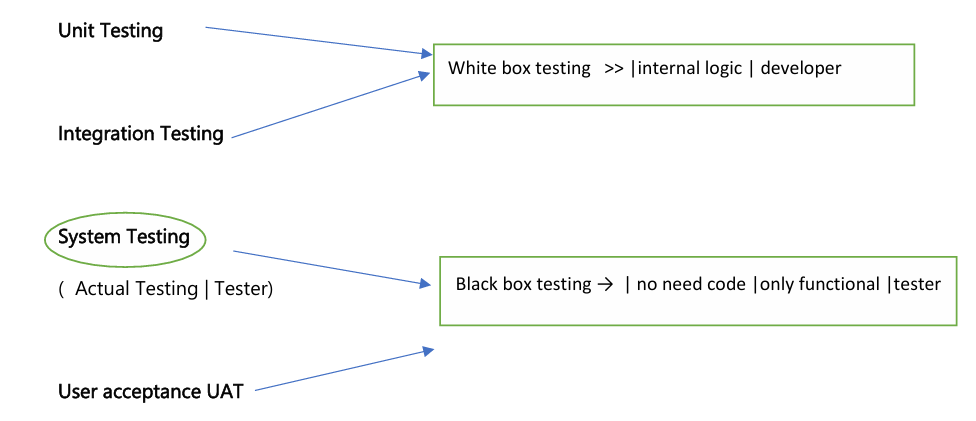
**Unit testing:** It is the type of functional testing, where individual units or components of a software are tested. It fall into white box testing category as code generally developers perform this unit testing**.**

**System testing:** System testing is a type of software testing where entire application is taken into the consideration for testing. It fall under black box testing. In this END-END testing will be performed. It will be mainly focuses on

1. User interface testing
2. Functional testing
3. Non Functional testing
4. Usability testing

**Integration testing:** system integration testing is a type of software testing in which user test interaction and communication b/w two or more system. The main goal of this is to interfaces & date flow b/w the different modules.it is done after the unit testing.

**Acceptance testing:** It is the last phase of the software testing process. And it is also know as UAT, FAT (Final acceptance testing) & RBT(Red box testing).it is done by the client and customers for there satisfaction.

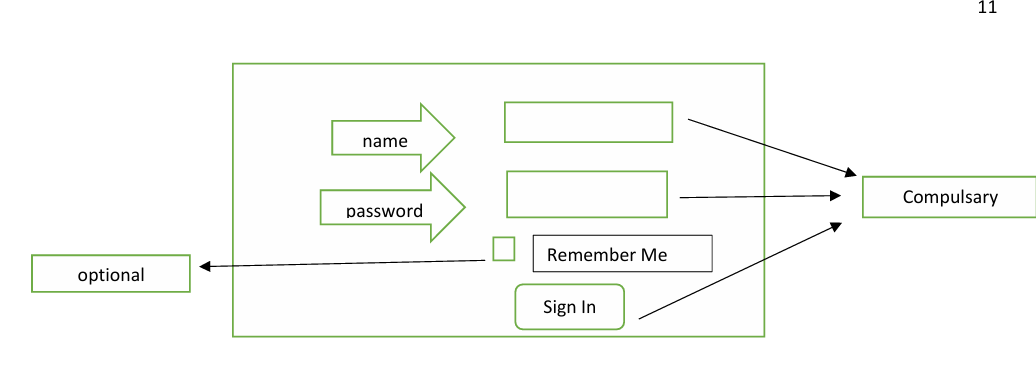


**Testing types:**

1. **Functional testing:** In will check if theAction will be performed or not**.** *example: Button clicks*
2. **Non Functional testing:** In this function that verifies the errors and it will check point to point of line and it verifies that each function of the software application.
3. Load test
4. Stress test
5. Volume test
6. Security test

**Smoke Testing:** In this whenever we get a new build we will check the major functionality of the application is working good or not**.**

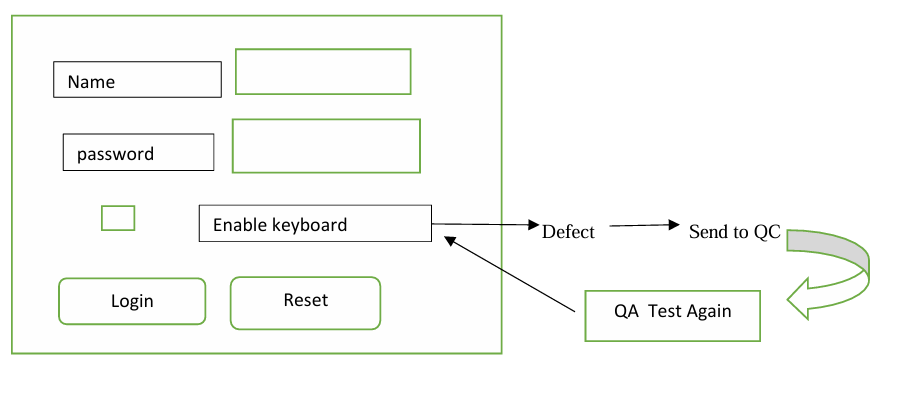
**For example:** Login functionality If If we are not able to login, it is difficult to perform further actions.

****

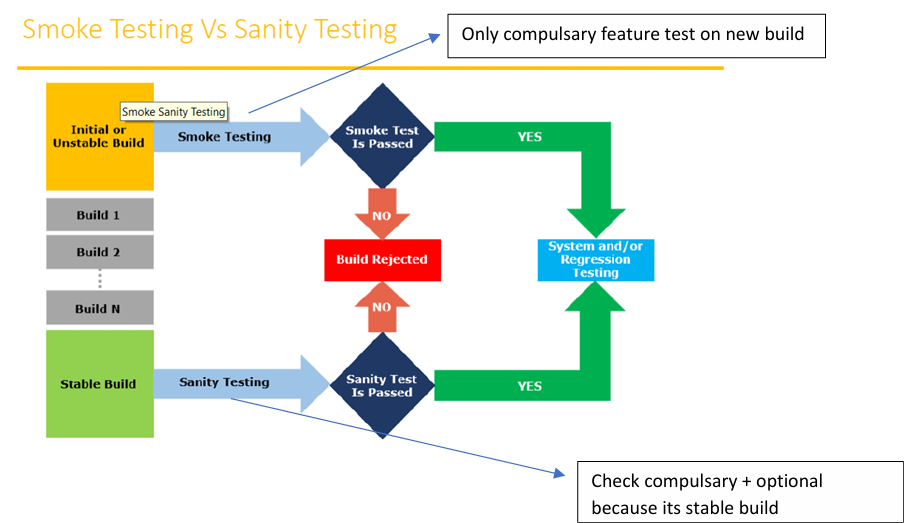
**Sanity Testing:** It will performed whenever we get a build with that build there will be a defect fix. So we will test the defect thoroughly.

**For example:** There is a text box which is having some defect and the developer has fixed it, has given the build. Now we have to test that.

**Re-testing:** It is the process of verifying that specific defects and bugs that was found. We have to check weather it is correct or not.



**Smoke Testing VS Sanity Testing**

****

**Static testing:** Itjust analyse the code, no need execution

**Dynamic Testing:** A testing which is done after code development & it requires code to be executed.

**Types:**

**Alpha testing:** It is a final testing in development & it will get immediate output.

**Beta Testing:** It is 1st testing in client side. it is also called user acceptance testing UAT & it will not respond immediate as Alpha testing.

**Installation Testing:** It is to check whether the software is properly installed or not with its necessary data & working as expected or not Deployment document /user manual: it is document prepared by project manager

**Usability Testing:** It is a technique that evaluates how easy or difficult user find product, website, or app.

**Monkey Testing:** It is used for game testing, used for random input, & random inputs or actions are given to the software system to test its behaviour. To check the application or system will crash.

**Portability Testing:** It Includes verifying that the software application can be installed & run on different platforms, that is uses hardware & software resources and that it is compatible with different version of the operating system & third-party software.

**Forced error Testing:** It is to check valid error message will display.

**Exploratary Testing :** When test engineer does not have idea of functional testing then he is learnig through exploring application

**End to End Testing:** We can check all internal componant for successful response Internal componant like Client, Network, Server Database etc are working fine Means Testing internal componant

**Security Testing:** Checking Secuirity of application

**Relability Testing:** The Developed application Should Support Longer Duration i.e. Stablity

**Audit:** it is independent evolution of software.

**Inspection:** it is formal evolution of software.

**Concurrency Testing:** multiuser Testing.

**De bugging**: executing program line by line for finding errors.

**Significant of design phase in the SDLC process:**

It is a Step by step process.

It is a process used by the software industry to design, develop and test software. SDLC aims is to produce the high-quality software that meets the customers expectation, reaches completely within time and with reasonable cost.

* Design is the second phase of SDLC.
* The input for this phase is SRS (software requirements specification) document which was prepared in requirement gathering phase and it is prepared on the bases of SRS document.
* In this phase multiple design documents are prepared which are documented in a SDD (software design document).
* SDD contains information like: Date design, Architecture design, Interface design
* There are two phases:

1. HLD (High level or System design)
2. LLD (Low level or program design)

* In design there will be prepare the blue print for the application.
* There are seven steps to design process

1. We have to gather the information
2. Analysis
3. Design
4. Coding
5. Testing
6. Deployment & Maintenance
7. we release the final product

