**Date 19/11/2024**

**1. What is Testing in software**

Testing is a part of the software development life cycle. The main aim of Testing is to deliver the product to the customers without any issues like bugs or defects. In the testing process testers will identify, and then developers will overcome the problem.

**2. What is Quality in Software**

Quality of software is defined as a product or project that is free from errors and bugs which should satisfy the client's requirement within the specified period.

**Advantages of Software Quality**

Free from errors

Delivered within time

Maintainable

Meets all the Requirements

**3.Difference between product and project**

If the Software application is developed only for a particular customer according to their requirements is called a project.

If the software application is developed for multiple customers according to market requirements is called a product.

**4.What is Error**

Errors occur when the developer team fails to understand the requirements of the client and misunderstanding translates into the buggy code.

**5.What is Defect**

A defect is defined as a situation when the application is not working as per the requirement and the actual and expected result of the application or software

**7.Why software bugs occur normally**

1. Human Error: It occurs due to a misunderstanding of the logic.

2. Inadequate Testing:Lack of proper testing may lead to bugs.

3. Complexity of software: When the application has millions of code.

4. Time Constraint: Tight deadlines lead to fast delivery of the product.

5. Changing Requirements: The Requirements may change during the development phase.

**8. Software Development Life Cycle(SDLC)**

Software development life cycle (SDLC) is a structured process that is used to design, develop, and test good-quality software. The goal of the SDLC life cycle model is to deliver high-quality, maintainable software that meets the user’s requirements.

SDLC is a collection of these six stages:

1. **Planning and Requirement Analysis:** During this stage, developers gather and analyze requirements based on customer inputs and market surveys.

2. **Defining Requirements:** In this stage, the developer will know what functional and nonfunctional and technical requirements are needed by utilizing the SRS(Software Requirement Specification.

3. **Designing Architecture:** In SRS it gives reference to the software developers with multiple architectures for the design.After evaluating all the factors most practical and logical design is chosen for the development.

4. **Developing Product:** In this stage development of the product starts. The developer will use any of the programming languages by considering some of the protocols like compiler, interpreter, and debugger.

5. **Product Testing and Integration:** After Developing the product it should be tested in order to fix any bugs, or errors to overcome. At every stage, minimal testing is done at every SDLC. This product gives the quality of the SRS.

**6. Deployment and Maintenance of Products:** After testing, the product is released in terms of phases to the organization if it works smoothly then it is released fully to the organization based on the feedback they can improve their product.

9. **What is** **the** **Waterfall Model**

The waterfall model is useful in situations where the project requirements are well-defined and the project goals are clear.

Advantages:

1. Simple and easy to Understand.

2. Clear Requirements.

3. Ideal for smaller projects.

4. Easier Testing and Maintenance

Disadvantages

1. Poor Adaptability to Feedback
2. Risk of Overlooking Requirements
3. Not Suitable for Complex Projects
4. Customer Involvement is Minimal

**10.Why Testing is necessary**

Testing is a crucial part of the software development process, ensuring that a product meets its requirements and functions as intended.

1. Software Quality of the product.

2. Detects and fixes errors.

3. Satisfaction of the consumer.

4. Security.

5. Validates User Requirement.

6. Determining the Performance of the application

**11.What is The Cost of Bugs**

The Cost of Bugs in Testing refers to the financial and operational impact that software defects (or bugs) can have on a project. The cost can vary depending on the stage of the development process when the bug is identified:

**During Requirements/Design Phase**: The cost is typically lower because identifying and resolving defects early in the process is less expensive.

**During the Coding/Development Phase**: The cost increases because the bug may require rewriting significant portions of the code.

**During the Testing Phase**: At this point, the cost of fixing a bug is higher, as the testing team may have to go back and forth to ensure it’s fixed and retested.

**After Release**: This is when the cost is at its highest. A bug discovered after the product is released to customers can lead to costly patches, customer support, and a potential loss of customer trust or revenue.

**12.Types of Testing**

**Manual Testing:** Testing the software or application according to the needs of client requirements without using any automation tools is called Manual Testing.We can say that it is a procedure of verification and validation.

**Benefits Of Manual Testing**

Fast Feedback

Easy Hiring

Less Expensive

In software testing, manual testing can be further classified into **three different types of testing**, which are as follows:

****White Box Testing****

****Black Box Testing****

****Grey Box Testing****

**Automation Testing:** Automation Testing is a type of testing in which we take the help of tools (automation) to perform the testing. It is faster than manual testing because it is done with some automation tools.

**Benefits of Automation Testing**

Find more bugs

Reduce Time

The process can be recorded

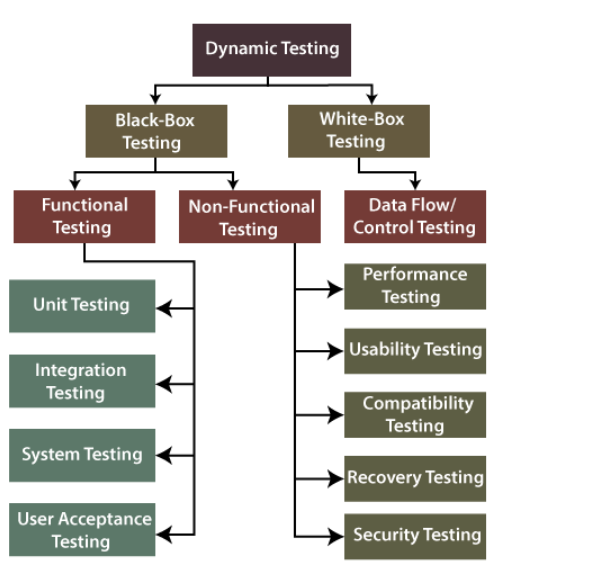
1. **Static Testing:** Static testing refers to the process of evaluating software without executing the code. It involves reviewing the code, documentation, and other software artifacts to find potential errors or issues before runtime**.**

**Types Of Static Testing**

1. **Software Inspection**: Software Inspection is a formal and structured process where a group of stakeholders systematically reviews software artifacts, such as code, design documents, or requirements, to identify defects and ensure compliance with standards.
2. **Structured Walkthroughs:** Structured Walkthroughs are informal peer review meetings where the developer of a software artifact (such as design, code, or test cases) presents the work to a group of peers for feedback.
3. **Technical Reviews** : Technical Reviews are focused discussions where a team of experts evaluates the technical aspects of a software artifact, such as its design, architecture, and implementation.

**Dynamic Testing:** Dynamic testing involves executing the software and observing its behavior during runtime to identify bugs, defects, and issues related to its functionality and performance**.**

**Levels of Dynamic testing**

****

**Black Box Testing:** It is a type of testing that focuses on validating the features and functions of the software as per the requirements and specifications.

**Day-3**

**Black-box testing is further classified into two types:**

1. **Functional Testing:**It is a type of software testing in which the system is tested against the functional requirements and specifications. Functional testing ensures that the requirements or specifications are properly satisfied by the application.

**2. Non Functional Testing:** Non-Functional Testing doesn’t focus on specific functionalities. Instead, it evaluates non-functional aspects such as performance, scalability, usability, security, and reliability.It provides detailed information on software product performance and used technologies.

**Functional Testing Types in Black Box Testing**

1. **Unit Testing:**It test individual Components in the software application and checks whether the functions are working correctly.
2. **Integration Testing:**Once we are successfully implementing the unit testing, we will go integration testing. It is the second level of functional testing, where we test the data flow between dependent modules or interface between two features is called **integration testing**.
3. **System Testing:**Whenever we are done with the unit and integration testing we will proceed with system testing.We will undergo each attribute and checks whether feature works according to the business requirement.
4. **User Acceptance:**It confirms that the system meets the user requirements.It is classified into 2 types
5. Alpha Testing:Conducted by the development team or QA team before release.
6. Beta Testing:Performed by end-users in a real-world environment.

### Types of Non-functional Testing

1. **Performance Testing:**In performance testing test engineer apply some load on the software application to check the response time,stability.
2. **Usability Testing:Usability Testing** is a type of software testing that evaluates how easy and user-friendly a software application is for end users. The primary goal is to identify usability issues, gather user feedback, and ensure the application meets user expectations for functionality and experience.

**Compatibility Testing:**It checks whether the software operates correctly across different environments, including:

1. Browsers
2. Operating systems

**White Box Testing:**

**White Box Testing** a type of software testing where the tester has access to the internal workings of the application. The tester tests the software with knowledge of its source code, design, and internal logic.

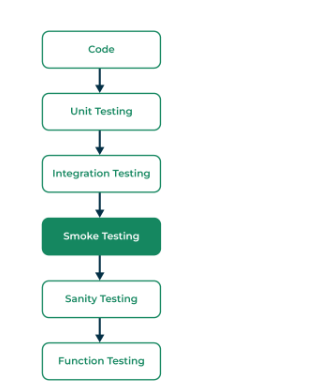
**Levels of White Box testing**

**1. Unit Testing:** Unit Testing is the lowest level of testing and involves testing individual units or components of the software, such as functions, methods, or classes, in isolation.

**2. Integration Testing:** It Examines how different parts of the application work together.It is done after unit testing to make sure components work well both alone and together.

1. **Regression Testing:** Regression Testing ensures that recent changes in the code (bug fixes, enhancements,or updates) do not negatively affect the existing functionality of the application.

**SMOKE TESTING**



* **Smoke Testing** is a preliminary level of testing used to quickly assess the basic functionality of a software build to ensure that it is stable enough for further testing.
* In software testing, smoke testing checks for critical issues that would prevent the application from functioning at a basic level.
* Smoke testing is a process where the software  build is deployed to a software assurance  environment and verified to ensure the application’s stability.
* We verify whether the important features are working and there are no showstoppers in the build that are under testing. It is a mini and quick regression test of major functionality.
* Smoke testing shows that the product is ready for testing. This helps in determining if the build is flawed to make any further testing a waste of time and resources.
* Tools used for smoke testing selenium,Junit,Jest

**SANITY TESTING:**

* Sanity testing is performed to ensure that the code changes that are made are working properly.
* Sanity testing is a stoppage to check whether testing for the build can proceed or not.

**RERESSION TESTING:**

* Regression Testing involves checking the quality after any changes made to the code .
* The goal is to verify that the software still performs as expected after modifications, ensuring that no new defects have been introduced into previously working features.

**Levels of Dynamic Testing**

**Installation Testing:**

* **Installation Testing** is the process of verifying that a software application can be successfully installed, configured, and uninstalled in the target environment.
* The goal is to ensure that the software works as expected in the installation process and that the user can perform tasks like setting up the application, configuring settings, and uninstalling it without issues.

**Monkey Testing**

**Monkey Testing** is an informal and random form of software testing where the tester interacts with the application in an unpredictable or random manner without any predefined test cases or plan.

**Portability Testing:**

* **Portability Testing** is a type of software testing that ensures a software application can be transferred from one environment to another and function as intended.
* That involves testing the software's ability to run on different operating systems, platforms, or hardware configurations without any issues.

**Forced Error Testing:**

**Forced Error Testing** is a type of software testing where the tester deliberately causes errors in the application to observe how it handles them.

**SDLC MODELS**

1. **Waterfall Model:**

* It is simple model which is easy to understand and use.
* It is also known as linear sequential model because each outcome of the one stage is input to the next stage.Then execution happens sequences manner .
* In order to avoid the overlaps in phases every stage is completed before moving to the next stage.

**Life Cycle of Water fall model**

**Requirement Phase**

**Design**

**Development**

**Testing**

**Deployment**

**Maintainence**

* In the first phase gathering the requirements from the stakeholders and analyzing them understanding the scope and objective of the application.
* Once the requirements are defined then designing the structure of the application which involves architecture,interfaces and component.
* The Development phase is an implementation which involve coding the software based on the design specifications.This phase consists the unit testing because each component is tested that the software is expecting.
* In testing phase the whole software is tested whether it is meeting the requirements and also it should be free from bugs.
* Once it is tested and Approved then it is deployed into the production environment.
* The final phase of the Waterfall Model is maintenance, which involves fixing any issues that arise after the software has been deployed and ensuring that it continues to meet the requirements over time.

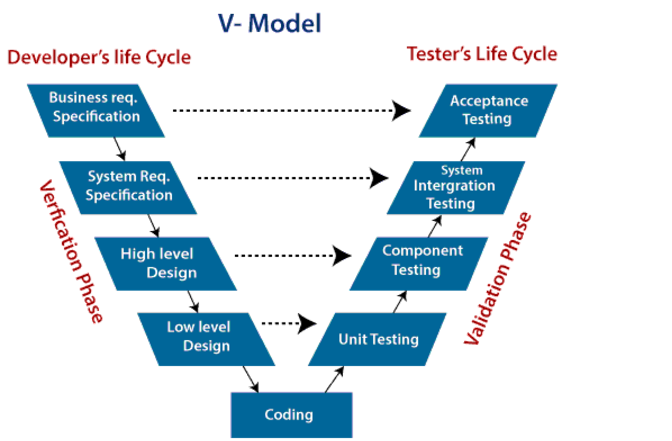
**Advantages of Waterfall model**

* Easy to Understand.
* Properly Defined.
* Properly Documented
* It is suitable for smaller projects where the needs are well understood.

**Disadvantages**

* Backtracking is not possible.
* Changes are not allowed because phases are dependent.
* Two teams cannot work parallel.
* Difficult to accommodate changes.

1. **V-Shaped Model**



* The V Model is also known as Validation model and Verification and it is an extension of the waterfall model.
* The V Shape gives the relationship between the testing phase and development phase
* Testing is planned in parallel to the corresponding development phase.
* **Verification** is a static analysis it is done without executing the code
* **Validation** is a dynamic analysis it is done with executing the code.
* V shaped model is chosed for the small to medium sized projects where the requirements are clearly defined.

**Advantages**:

* Easy to Understand.
* Testing Methods like planning, test designing happens well before coding.
* This saves a lot of time. Hence a higher chance of success over the waterfall model.
* Works well for small plans where requirements are easily understood.

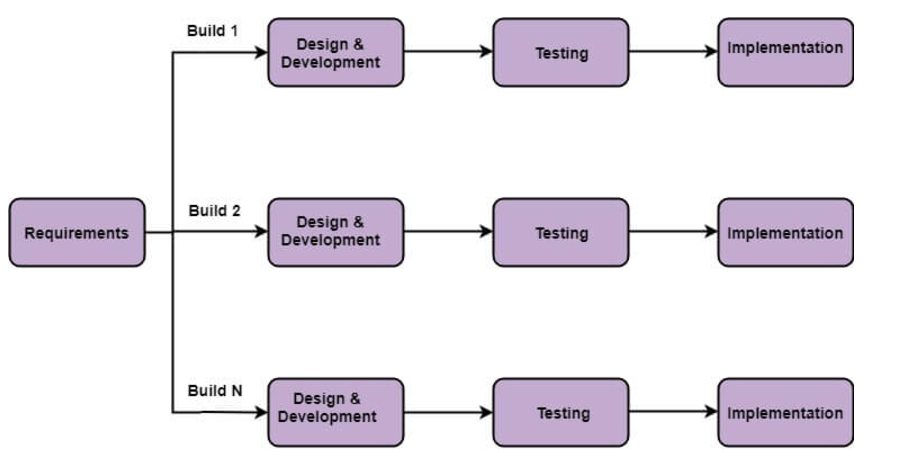
**Disadvantages**

* Not a good for a complex project.
* If any changes happen in the midway, then the test documents along with the required documents, has to be updated.
* The V-Model can be time-consuming, as it requires a lot of documentation and testing.
* This model does not support iteration of phases.

1. **Incremental Life Cycle Model**

In Incremental model first implemented basic features and it is delivering to the client. Then after many iterations are implemented and delivered to the customer until the desired application is built.

It is also known as Sucessive version model.



Incremental model is used when the application is too lengthy and the when the client wants the quick release of the product.

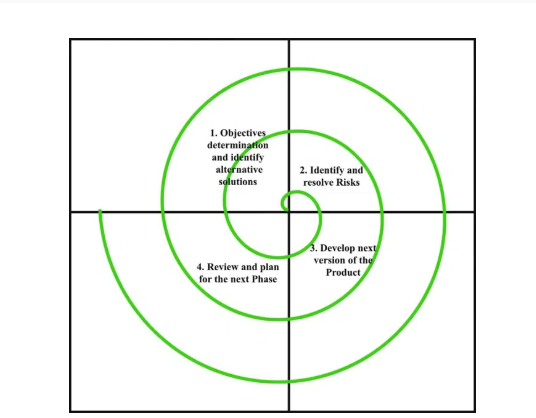
## **Advantage**

* Errors are easy to be recognized.
* Easier to test and debug
* More flexible.
* Simple to manage risk because it handled during its iteration.
* The Client gets important functionality early.

## **Disadvantage**

* Need for good planning
* Total Cost is high.
* Well defined module interfaces are needed.

**Spiral Model**



Spiral model is combination of waterfall model and iterative model .It provides support to risk handling.

In diagrammatic representation,looks like a spiral with many loops.The exact number of loops vary from project to project.Each loop of spiral is called phase.

The exact number of phases needed to develop a product can be varied by the project manager depending on the project risk.

Each phase of the model is divided into four quadrants.

1. **Objectives determination and identify Alternative solutions:**Requirements are gathered from the customers and objectives are identified and elaborated and analyzed at the start of the phase.
2. **Identify and Resolve Risks:**In the second quadrant,all the possible solutions are evaluated and selects the best possible solution.Then the best solution associated with risk is chosen and resolved.At the end of the quadrant,prototype is built for best possible solution.
3. **Develop next version of the product:**In the next Phase,the identified features are developed and verified through testing.At the end the next version of the software is available.
4. **Review and plan for the next phase:**Customers evaluate the developed version of the software.In the end,Planning for the next phase is started.

**Advantages**:

* Good for large projects
* Customer Satisfaction
* Risk Handling
* Iterative and Incremental

**Disadvantages**

* Complex
* Expensive
* Difficult in time management

**Difference between Testing and Debugging**

|  |  |
| --- | --- |
| Testing | Debugging |
| The process of Identifying errors in software by executing | The process of analyzing and fixing the identified errors. |
| It is performed by the testers. | It is performed by the developers. |
| It happens after the Development Phase to identify issues. | It happens after the testing to fix it. |
| The purpose is to find bugs,errors | To locate the exact cause of a defect. |
| Test management Tools are used like Selenium,Junit | Debugging tools like visual studio debugger,chrome devtools |
| Identifying that a login is fails when the username and password is entered. | Fixing the issue in the code causing the login function to fail. |

Difference Between Black Box Testing and White Box Testing

|  |  |
| --- | --- |
| **Black Box Testing** | **White Box Testing** |
| Testing the software without the knowledge of its internal code or structure. | Testing the software will have knowledge of its internal code and structure. |
| It is Performed By Testers | It is performed by Developer. |
| No programming knowledge is required. | Programming knowledge is required |
| No Implementation knowledge is required. | Implementation Knowledge is required. |
| Black-box testing is external testing. | White-box testing is internal testing. |
| Ex:Functional Testing Usability Testing System Testing Acceptance Testing | Ex:Unit Testing Code Coverage Analysis Security Testing |
| It is called Functional Testing | It is called Structural testing |

**Difference between static and dynamic testing**

|  |  |
| --- | --- |
| **Static testing** | **Dynamic testing** |
| This testing is done without executing the code. | This testing is done with executing the code. |
| Static testing is done before the compilation | Dynamic testing is done after the compilation |
| To identify potential defects in code, design, or documentation. | To ensure the software behaves as expected by running it. |
| This testing is done in verification stage. | This testing is done in validation stage. |
| No code execution | Involves actual execution |
| The cost is less for detecting the errors and fixing. | The cost is more for finding and fixing. |

**Difference between verification and validation**