MANUAL TESTING

Testing:

**03-02-2025.**

Manual testing is a software testing process where test cases are executed manually by testers without using automation tools. It helps identify bugs, usability issues, and unexpected behaviour in a software application.

The process of identify the error/bugs/defects.

(or)

The process of checking withers the project or the product is satisfying the needs requirements of the client or not.

**Object** **of** **Testing:**

The object of testing is to ensure that software is reliable, efficient, and meets user requirements. Testing objectives help to define the goals of the testing process and the timeline for completing it.

1.Ensuring the quality of the project/product.

2.Ensuring the correctness of project/software developer.

3.Enusuring the robustness of the software developer.

4.Enusuring the reliability of the software developer.

5.Enusuring the product is bug free before shipment.

**Why We Need Testing:**

To ensure the quality of the project / product.

Software testing is essential to ensure that a product functions correctly, is free from defects, and meets user expectations. Without proper testing, software can have critical bugs that may lead to system failures, security breaches, financial losses, and poor user experience.

**Quality:**

It is defined as justification of all the requirement of the customer in an application.

Software quality refers to the ability of a software product meet customer requirements, function correctly and perform efficiently in different environments. It ensures the software is reliable, maintainable, Secure, and user friendly.

**Software Testing:**

It is a part of software development life cycle. software development testing is an activity to detect & identify the defects present in the developed software.

Need of software testing.

1.inorder order to deliver a very good quality software.

2.To check whether it is satisfying the needs of customer or not.

3.To check whether the developed software is a user friendly.

4.To reduce the maintenance cost.

5.Inorder to avoid the negative feedback from customers.

**Software Quality:**

Quality: Quality is defined as justification of all the requirements in a product.

1.Which is completely bug free.

2.When we deliver the software in time with no bugs.

3.If we develop the software with in the budget.

4.user friendly.

5.If needs to be maintainable.

**Product Vs Project:**

Product: If the software is developed based on the requirement of multiple customers in the market, then is known as product.

**Key Characteristics of a Product:**

Ongoing Lifecycle, customer centric, Value-Driven, iterative improvements, market focused.

**Smartphone (iPhone):**

A smartphone is a product because it is continuously manufactured, sold, and improved over time. New versions and updates are released regularly to improve performance, features, and customer satisfaction.

Project: If the software is developed based on the requirement of single customers, then it’s known as project.

**Key Characteristics of a Product:**

Temporary, defined objective, scope based, one time effort.

**Software Feature Development**

* Adding a newfeature to an existing software application is a project. It has a start date, specific goals, and will be completed once the feature is implemented and tested.

EX: Amazon Ajio------> Multiple products like cloths home items grocery.

Amazon AWS------> market available customer’s project.

Errors:

Incorrect human action.

Bug/defect: - deviation from the action behaviour/performance.

Failure: - The complete deviation from the action behaviour of the system software.

04-02-2025

Software Testing:

Software testing is an important process in the software application that was fully developed to identify defects, ensure quality, and verify that it meets specified requirements. It involves verifying and validating that a software application is free bugs.

Types of Software Testing:

Two Types:

1.Manual Testing.

2.Automation Testing.

Manual Testing:

It includes testing software manually without using any automation tool or script. In this type, the testes over the role of an end user and tests the software to identify any unexpected behaviour or bug.

Human tests will write the tests cases they run it and generate reports.

Advantages of Manual Testing:

1.Cheaper less cost.

2.it is accurate then automation testing procedure.

3.it has quick visual feedback.

4.its id ideal for testing minor changes.

5.Human judgment and intuition.

6.Suitable for UI (user Interference).

Disadvantages of Manual Testing:

1.Time consuming.

2.cost more expensive for long run.

3.we cannot record the process of manual testing.

4.Not suitable for large scale testing.

Automation Testing:

Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves automation of a manual process. Automation Testing is used to re-run the test scenarios that were performed manually, quickly, and repeatedly.

Categories of software testing.

Static Testing:

->Analysis of program is don without execution verifying the documents.

-> Static testing is performed in the early stage of development to avoid errors as it is easier to find sources of failures and it can be fixed easily.

->Static testing is no done while running the test process.

->it involves manual and automatic assessment of the software testing.

Dynamic Testing Techniques:

->Dynamic testing it is done while executing the testing process.

-> The purpose of dynamic testing is to confirm that the software product works in conformance with the business requirements.

->it provides more realistic result than static testing.

Types Dynamic Testing:

1.Black Box Testing. (functional testing)-high level UAT (testing) system.

2.Black box testing looks only at the functionality of the Application Under Test (AUT).

Types of black box:

1.systam testing.

2.UAT testing.

3.white box.

Levels of Testing.

white box Testing.

1.Unit Testing:

Its unit testing individual component of software tested. The purpose of this testing is that each module is working properly.

2.Integration Testing:

In integration Testing individual units are combined and tested as group (developer). individual components working when collaborate or integrated with other companies.

3.System Testing:

in this testing we can test whole application (complete / integrated software is tested) done by tester.

4.Acceptance Testing:

Test don to get position feedback from the customer

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

5.Smoke Testing:

Testing don only on the newly released software application.

Smoke Testing is a preliminarytest performed to check whether the basicfunctionalitiesofasoftwareapplicationareworking before proceeding with detailed testing.

**05-02-2025**

**Software Development Life Cycle (SDLC):**

SDLC Software Development Life Cycle is a process used by software industry to design, develop and test software. SDLC or software development life cycle is methodology that define the entire procedure of software development step by step.

**Stages of the SDLC.**

**1.Requirements gathering:**

This phase involves gathering information about the software requirement from stakeholders such as customers end users and business analysts.

**2.Analysis:**

Create a Software Requirement Specification (SRS) document.

**3.Desing Architecture:**

SRS is a reference for software design to come up with the best architecture for the software. The overall architecture of the software data structures, and interfaces.

High level design (HLD)

Low level design (LLD)

**4.Developing Coding:**

developers write the actual code based on design documents.

developers use a specific programming code as per the design in the DDS.

**5.Testing:**

The development team combines automation and manual testing to check the software for bugs. Fix bugs and issues.

After the code is generated, it is tested against the requirements to make sure that the products are solving the needs addressed and gathered during the requirements stage.

**6.Deployment:**

After the code is generated, it is tested against the requirements to make sure that the products are solving the needs addressed and gathered during the requirements stage. Conduct final testing and user acceptance testing (UAT). Release the product to users.

**7.Maintenace:**

In the maintenance phase among other tasks the team fixed

bugs resolve customer issues, and Manges software changes.

Provide updates, patch, and bug fixes. Enhance software based on use feedback.

**Software Development Life Cycle Models:**

Software Development life cycle is a spiritual models used in project management that defines the stages include in an information system development project from an initial feasibility study to the maintenance of the completed.

1. Water falls model.
2. Agile Methodology.
3. Spiral model.
4. Incremental model.
5. Prototype model.
6. V-model.

**1.Waterfall model:**

The water fall model is a universally accepted SDLC model. In the method the whole process of software development is divided into various phases. The water fall model is software development model used in the context of large complex project.

🡪Older version to develop a software.

🡪it is an old and tradition method of developing a software.

🡪By sequence execution line by line.

🡪Initial investment is less because no tester involved.

🡪Testing will be done only after completing the software was developed.

🡪it is suitable for only for smaller projects.

🡪Requirements in waterfall model are fixed.

🡪budget friendly

🡪difficult to measure the progress of the project.

🡪client involvement is very less.

\*Phases of waterfalls model:

**1.Requirements:**

All software requirements are collected, analysed, and documented in a software Requirement specification (SRS) document.

There is little to no scope for changes once finalized.

**2.System design:**

Once the requirements are understood, the design phase begins. This involves creating a detailed design document that outlines the software architecture, user interface, and system components.

**3.Implementaion(coding):**

Developers write code based on the system design. The Development phase include implementation involves coding the software based on the design specifications.

**4.Teating:**

In this testing phase the software is tested as a whole to ensure that it meets the requirements and is free from defects. Testing is deferent types of testing and produces relevant report.

**5.Deployment:**

The final software produce is delivered and installed in the client environment. Once the software has been tested and approved it is deployed to the production environment.

6.Maintenance:

Software is monitored for issues and necessary updates bug fixes and enhancements are made.

**Advantages:**

1. **Simple and Easy to Understand**: The process is linear and sequential, making it easy for teams to follow.
2. **Clear Documentation**: Since each phase is completed before the next begins, detailed documentation is produced for each stage, ensuring clarity and consistency.
3. **Structured Approach**: The well-defined phases make it easy to manage and track project progress, particularly in projects where requirements are stable and unlikely to change.
4. **Ideal for Small Projects**: Waterfall works well for small-scale projects with clear and fixed requirements.
5. **Easy to Manage for Non-technical Stakeholders**: The clear structure allows stakeholders to easily understand the progress and status of the project.

**Disadvantages:**

1. **Inflexible to Changes**: Once a phase is completed, it’s difficult to go back and make changes. This makes it less suitable for projects where requirements may evolve.
2. **Late Testing**: Testing happens after the development phase, which can lead to discovering defects or issues late in the process, resulting in delays or increased costs.
3. **Not Suitable for Complex Projects**: For larger, more complex projects with evolving requirements, the rigid structure of Waterfall can be limiting.
4. **Risk of Miscommunication**: If the initial requirements are misunderstood or poorly defined, the entire project may be affected because changes in later stages are costly and complicated.
5. **Time-Consuming**: Since each phase must be completed before moving to the next, the process can be slow, particularly in cases where there's a need for extensive revisions after testing.

**2.Agile methodology:**

The AgileModel is an iterativeandincrementalapproach to software development, focusing on flexibility, collaboration, and customer feedback. Unlike the WaterfallModel, where development follows a strict sequence, Agile allows for continuous improvements through multipleiterations (or sprints).

🡪latest method of software development.

🡪requirements of clints are taken throughout the process.

🡪changes at can be made at any stages.

🡪mostly used to develop large project.

🡪client not want to wait for the whole software to develop.

🡪agile methodology is delivering the piece of software which was developed.

🡪testing is done in piece of don in software development.

🡪Delivery time is early/less time.

🡪development cost is very less.

🡪bugs are detected and fixed early.

🡪shopping ex: Flipkart amazon Myntra social media Facebook twitter.

**Agile method has 3 basics principles:**

1. Costumer no need to wait till the whole software is developed.
2. Delivering the piece of software which contains same functionality which is developed and tested.
3. We can adopt accept the requirements changes from the customer at any point of process.

**Advantages:**

1.requirement changes are allowed at any piece of proses.

2.Relise will be very fast.

3.customer no need to wait for longer time.

4.easy to adopt.

5.improved customer satisfaction.

6.continuous improvement.

**Disadvantages:**

1. Since Agile promotes flexibility and continuous changes based on customer feedback, there is a risk of scopecreep.

2.Agile relies on highly skilled and self-organizing teams.

3. Agile requires constantcustomerinvolvement to provide feedback and guide the direction of the project.

4. Agile methodologies require frequentmeetings such as daily stand-ups, sprint planning, reviews, and retrospectives.

**06-02-2025**

**Spiral model:**

The spiral model is a software development life cycle (SDLC). model that provides a systematic and iterative approach to software development.

Overcome the throwback back of a water fall model.

1 cycle 1 new relies of software development v1

Phases of spiral model

1.Planning

2.Risk analysis.

3.Engineering.

4.Evallution.

1.Planning:

Spiral begins a new planning phase based on the results.

Gather requirements and define project scope.

2.Risk analysis:

The risk associated with project are identified and evaluated.

Develop strategies to mitigate these risks

3.Engineering:

This is developed based on the requirements gathered in the previous iteration.

Design and implement develop test.

4.Evallution.

Decide whether to proceed, make modifications, or terminate the project. Plan the next iteration in the spiral.

**Advantages:**

1.Frequent relies will be three.

2.Requirements changes will be adopted easily.

3.Customers no need of waiting until the next changes is done.

4.Testing is done in every done.

**Disadvantages:**

1.Time consuming.

2.evry cycle of spiral model looks like waterfall model.

3.there is no testing in requirement and design phase

----------------------

**4.Increemnetal model:**

Requirement are divided into multiple module each module goes through SDLC.

The Incremental Model is a software development approach where the system is built and delivered in small, manageable increments rather than as a single complete product.

**Phase of incremental module:**

1.Requirement analysis.

2.disign & development.

3.testing.

4.delivery integration.

**Advantages:**

Early delivery: The customer gets a partially function product early.

Flexibility: Changes can be made in future increments.

Easier Testing: each model is tested individually reducing.

**Disadvantages:**

No proper Planning.

Cost is very high.

Not suitable for rapid changes.

**5.prototype.**

The prototype model is one of the most popularly used SDLC.

This model is used when the customer does not know the exact project requirements beforehand.

Phases of prototype:

1.requirement gathering and analyst.

2.quick decision.

3.bulid prototype.

4.evaluion.

5.product.

**Advantages of prototype:**

Better requirement clarity reduces misunderstandings between developers and user.

EarlyDetectionofIssues Users can identify missing features or problems early.

Support early product marketing.

**Disadvantages of prototype:**

Time consuming, Higher cost, scope creep, not suitable for all projects.

**Cloud Computing:**

cloud computing is a technology that allows users to access and store data, application over the internet instead of using local servers or personal computing. CloudComputing means storing and accessing the data and programs on remote servers that are hosted on the internet instead of the computer’s hard drive or local server.

Rescuers like servers, databases storage, networking and software.

Analysis over the internet.

What types of cloud computing.

1. Infrastructure as a service (IAAS).
2. Platform as a service (PAAS).
3. Software as a service (SAAS).
4. Function as a service (FAAS).

**Deployment models:**

1.public cloud:

A public cloud is a cloud computing environment where computing resources are provided by a third-party vendor and shared among multiple users over the internet.

🡪AWS, google cloud, Microsoft cloud.

🡪 **Microsoft Azure** – Provides cloud infrastructure, AI, and hybrid cloud solutions

2.Private cloud:

🡪ON/OFF premises cloud.

🡪A private cloud is a cloud computing environment that is exclusively used by a single organization.

🡪Unlike the public cloud, where resources are shared among multiple users, a private cloud offers greater control, security, and customization because the infrastructure is dedicated to one entity.

🡪They will give companies direct controller the data.

3.Hybrid Deployment model:

It comes up with a combination of elements of both private and public clouds providing seamless data and application processing in between environments.

🡪 Uses public cloud for scalability and private cloud for security-sensitive workloads.

🡪 Extends AWS services to on-premise data centres.

🡪 Allows hybrid cloud deployment across multiple environments.

4.Community cloud:

🡪A community cloud is a cloud computing model where multiple organizations with similar needs share the same cloud infrastructure.

🡪 It provides a balance between security, cost-effectiveness, and collaboration while being more customizable than a public cloud and more cost-efficient than a private cloud.

🡪shared infrastructure, higher security, collaboration & resource, cost effective.