**Software Testing 03-02-2025**

**Testing:**

* Testing is done to identify the bugs/errors of the product/project.
* Testing is done to check whether the project/product meets the requirements of the customer.

**Objectives of the testing:**

* We have to ensure the quality of the product/project.
* To ensure whether the software meets the specific requirements or not.
* To ensure the product/project is user-friendly or not.
* To minimize the risk of failures in production.
* To validate the software functionality, reliability and robustness.
* To detect errors/bugs.

**What is the need of the testing?**

* To ensure the quality of the product/project.
* **Quality:** Here, the quality defined as whether the developed software meets the requirements of the customer.
* To ensure the system is secure from potential threats.
* Fixing the bugs after release of the product/project is more expensive.

**Software Testing:**

* It is the part of the SDLC (Software Development Life Cycle).
* Software testing is done to identify the bugs/errors in the developed software.

**Need for Software Testing?**

* We have to deliver the good quality software product/project.
* To check whether it is satisfying the needs of the customer or not.
* To check whether the developed software is user-friendly or not.
* To minimize the maintenance cost.
* To avoid the negative feedbacks.
* To reduce the risk of failures in production

**Quality Software:**

* The quality software is ensured, when it is completely error free/bug free.
* And, we have to deliver the software in-time.
* We have developed a user-friendly software.
* The developed software should be maintainable.
* To build a quality software, we have to meet the specific requirements of the customer.

**Difference between Product and Project:**

**Product:**

If the software is developed based on the requirements of the multiple customers to ensure the quality of the product in the market is the product.

**Project:**

If the software is developed based on the requirements of the single customers to ensure the quality of the project in the market is the project.

**Error:** An incorrect human action. Error raises due to several reasons like design issues, coding issues, or system specification issues and leads to issues in the application.

**Bugs/Defects:** Deviation from expected behaviour to actual behaviour of the product/project. The most common type of bug is crash.

**Failure:**  If any bug/error raised during the execution of the product/project at the end-user side is called failure.Failure is detected by the end-users.

**Why the software has the bugs?**

* Due to programming errors.
* Changing requirements frequently.
* Software complexity
* Due to Unskilled testers.

**--------------------------------------------------------------------------------------------------------------------------**

**Software Testing Types: 04-02-2025**

There are 2 types of software testing.

**1.Manual Testing**

🡪It is the process where human testers write the test-cases for testing the software to generate reports accordingly.

🡪The tester checks all the essential features of the application.

🡪It is conducted by the experienced tester to accomplish the testing process.

**2.Automation Testing (Test Automation)**

🡪Automation testing is a type of testing in which we take the help of tools to perform testing.

🡪In this type of testing, we will write test scripts instead of test cases.

🡪It is faster than the manual testing.

🡪Automation needs some human support to create initial testing scripts.

**Advantages of Manual testing**

* Live testing—If any bugs/errors raised, when the application is live, these kinds of errors are solved using manual testing.
* Less programming
* Low-cost investment
* Adaptability to change
* It is accurate than automation testing procedure
* Minor changes in the test-cases are testing by manual testing

**Disadvantages of Manual testing**

* Time consuming
* Possibility of errors
* Testing limitations—Manual testing is not good for load and performance testing. The performance testing requires large no. of users and gadgets.

**Difference between Manual and Automation testing**

|  |  |
| --- | --- |
| **Manual Testing** | **Automation Testing** |
| **🡪**Guarantee user-friendliness | **🡪**Doesn’t guarantee user-friendliness |
| **🡪**Time consuming is more | **🡪**Time consuming is less |
| **🡪**Not accurate at all times | **🡪**Highly accurate |
| **🡪**It can’t be acclaimed as reliable | **🡪**Comparatively more reliable |
| **🡪**Less programming | **🡪**More programming |
| **🡪**Not much expensive | **🡪**Expensive |
| **🡪**In terms of hiring resources | **🡪**In terms of purchasing resources |

**Categories/Techniques of Software testing**

There are 2 types of techniques in software testing.

**1.Static Testing:**

🡪Static testing techniques are testing techniques that are used to find the defects in an application without executing the code.

🡪This type of testing is done to avoid the errors at the early stage of the development.

**2.Dynamic Testing**

**🡪**Dynamic testing techniques are testing techniques that are used to test the dynamic behaviour of the application under test, that is by the execution of code-base.

🡪The main purpose of the dynamic testing is to test the application using dynamic inputs.

**Types of Dynamic testing**

1.White-box testing

2.Black-box testing

**1.White-box testing:**

**🡪**The developers can perform the white-box testing.

🡪It is used to test internal logic and codebase.

🡪Requires the knowledge of the codebase.

🡪The main aim of the white-box testing is to check on how system is performing based on the code.

**2.Black-box testing:**

**🡪**Black-box testing is mainly performed by the testers.

**🡪**It is used to test the functionality without knowing the internal code.

**🡪**There is no need of programming languages.

**🡪**The black-box testing is again classified into 2 types:

* Functional
* Non-functional

**Levels of Testing:**

**1.Unit testing:**

**🡪**Unit testing is the process of testing individual components and units.

🡪The primary objective to perform unit testing is to test the correctness of the remote code and validate the unit components with their performance.

**2.Integration testing:**

**🡪**In integration testing individual units are combined and tested as group. And it is second level of functional testing.

🡪The developers and test engineers perform the integration testing.

🡪The main purpose of the integration testing is to identify the faults in the integrated units.

**3.System testing:**

🡪System testing is used to check the end-to-end flow of an application.

🡪System testing is also known as end-to-end testing as the testing environment is similar to the production environment.

🡪It is the third level of the functional testing. In this, we go through all the modules of an application.

**4.User Acceptance testing:**

🡪UAT is final phase of testing before a system software eis deployed.

🡪UAT is done by the customer for their satisfaction and check whether the application is working according to the business requirements or not.

🡪UAT has 2 types:

**Alpha testing:** Here the testing is done internally by the development team and QA testers.

**Beta testing:** Here the testing is done by the selected group of the external customers/users before the final launch.

**5.Smoke testing:**

🡪Smoke testing comes into the picture at the time of receiving the build software from the development team.

🡪The purpose of the smoke testing is to determine whether the build the software is testable or not. It is done at the time of ‘building software’.

**Difference between Bug and Defect:**

|  |  |
| --- | --- |
| **Bug** | **Defect** |
| **🡪**An issue is found during testing before release. | **🡪**When the application is not working as per the requirements is known as defect. |
| **🡪**It is identified by the testers during the development/testing. | **🡪**It is identified by the developers. |
| **🡪**Bug is noting but fault in the software. | **🡪**Defect is variable of the output. |

**Difference between Error and Failure:**

|  |  |
| --- | --- |
| **Error** | **Failure** |
| **🡪**A mistake in the code or logic made by the developer. | **🡪**If an end-user detects the issue in the product, then it is called failure. |
| **🡪**It is occurred during development | **🡪**It is occurred during use of the software. |
| **🡪**It is identified by the developers. | **🡪**It is identified by the testers or end-users. |

**SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC): 05-02-2025**

The SDLC is a structured process of developing the software applications. It consists of the following phases.

1.Planning

2.Requirement gathering and Analysis

3.Design

4.Coding/Development

5.Testing

6.Deployment and Maintenance

**1.Planning:**

🡪Here we define the project scope, feasibility and objectives.

🡪Planning is done by Project Managers, Product Owners and Stakeholders.

**2.Requirement gathering:**

🡪Here, we gather and document the functional and non-functional requirements.

🡪Requirement gathering is done by Business Analysts, Project Managers and Stakeholders to define the meetings, user interviews and documents.

**3.Analysis:**

🡪After the requirement gathering is done, here we certainly represent and document the software requirements.

🡪This is accomplished through ‘SRS’-Software Requirement Specification document.

**4.Designing:**

🡪Architects and designers create system blueprints, UI/UX, and database models.

🡪With the help of requirements defined in the SRS, multiple designs for the product architecture are present in the Design Document Specification (DDS).

🡪This DDS is assessed by market analysts and stakeholders.

**High level design:** It defines the system architecture.

**Low level design:** Describes how each and every feature in the software product should work.

**5.Development/Coding:**

🡪Developers write the code based on design specifications.

🡪Uses programming languages, frameworks and tools.

**6.Testing:**

🡪Quality Assurance (QA) teams perform unit, integration, system and user acceptance testing.

🡪Identify and fix detects through various testing methods.

🡪Automated and Manual testing are used for testing.

**7.Deployment:**

🡪Here, the software is released into production.

🡪It is handled by DevOps Engineer, System administrators or Release managers to manage CI/CD pipelines and production pipelines.

**8.Maintainance:**

🡪Provide support, updates, and bug fixes after deployment.

🡪It is managed by support engineers, maintenance teams or Development teams.

**Need of SDLC:**

🡪The development team must determine a suitable life cycle model for a particular plan and then observe to it.

🡪Without using the exact life cycle model, the development of a software product would not be in a systematic and disciplined manner.

🡪When a team is developing a software product, there must be clear understanding among team.

🡪A software life cycle model describes entry and exit criteria for each phase. A phase can begin only if its stage-entry criteria have been fulfilled.

🡪So, without a software life cycle model, the entry and exit criteria for a stage cannot be recognized.

**WATERFALL MODEL:**

🡪The Waterfall model is a linear and sequential approach to software development, where each phase must be completed before moving to next.

🡪It is best suited for projects with well-defined requirements and minimal changes.

🡪This classical waterfall model is simple and idealistic.

🡪Here, the testing is done, only when the software was developed completely.

🡪It is suitable for only smaller projects.

🡪Requirements are fixed in waterfall model.

🡪It is budget-friendly model in order to develop a software application.

**Phases of Waterfall Model:**

The following are phases of waterfall model:

1.Requirement gathering

2.System Design

3.Implementation

4.Testing

5.Deployment

6.Maintainence

**1.Requirement gathering:** Here, we gather requirements from stakeholders and analysing them to understand the scope and objectives of the project. By using requirements, develop an SRS document.

**2.System Design:** This involves creating a detailed design document that outlines the software architecture, user interface and system components.

**3.Implementation:** It involves the coding based on the design specifications. This phase also involves unit testing.

**4.Testing:** In the testing phase, the software is tested as a whole to ensure that it meets the specified requirements.

**5.Deployment:** Once, the software is tested and approved, it is deployed into the production to make use of the end-users.

**6.Maintainence:** It involves fixing any issues that arise after the software has been deployed.

**Advantages of Waterfall Model:**

🡪Easy to understand

🡪Individual processing—Phases in this model are processed one at the time.

🡪It is well-fit for small projects.

🡪Sequential execution.

🡪It is budget-friendly model.

**Disadvantages of Waterfall Model:**

🡪The requirements are fixed at the starting stage of the project. Here, we cannot change the requirements.

🡪Testing will be done, only after completion of the project.

🡪Client involvement is less.

🡪No overlapping on phases.

🡪Time consuming, because only one phase is done at a time.

**AGILE METHODOLOGY:**

🡪Agile is flexible, iterative approach to software development that focuses on collaboration, customer feedback and continuous delivery.

🡪It is the latest method of software development.

🡪Requirements of client is taken throughout the process.

🡪Changes can be made at any stage of the project.

🡪It is mostly used for the large projects.

🡪We can test the project, even the project is completed or not.

🡪The development cost is very less.

Agile has 3 basic principles:

* Customer no need to wait untill the whole software gets ready.
* We can adopt/accept the requirements changes from the customer at any stage of the project.
* Delivering the piece of the project software which contains some functionalities which is developed and tested.

**Advantages of Agile Methodology:**

🡪Requirement changes are allowed at any phase of the project/process.

🡪Frequent delivery ensures user satisfaction.

🡪Continuous collaboration improves software quality.

🡪Easy to adopt

🡪We can test the project at any time of the process.

**Disadvantages of Agile Methodology:**

🡪It is more code-based and produce less documentation.

🡪Face-to-face communication is harder in large-scale organizations.

🡪Only senior programmers are capable of taking the decisions.

🡪Requires active customer involvement.

🡪Less predictable scope and timeline.

🡪Can be challenging for large, complex projects.

🡪Lack of formal documentation leading to potential confusion.

**Agile Scrum:**

Scrum is a popular Agile framework that organizes work into small, iterative cycles called sprints (typically 1-4 weeks). It focuses on delivering incremental value through collaboration and continuous feedback.

**Agile sprint meeting:**

A meeting where the scrum team plans the work for the upcoming sprint.

**Agile review meeting:**

* Here, we have to show the work at the end of the sprint session for feedback.
* If the product still has some non-achievable features, it will be checked in this stage and then passed to the Sprint Retrospective stage.

**Agile retrospective meeting:**

* Here, there will be a discussion about what went well, what not went well and how to improve in next sprint.
* In this stage quality or status of the product is checked.

**Daily Scrum (Stand-up)** – Short meeting to discuss progress and obstacles.

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

**SPIRAL MODEL 06-02-2025**

🡪It is an iterative model.

🡪It is also called as Meta model.

🡪Spiral model provides support for **Risk Handling.**

🡪It is combination of Waterfall model, Iterative model & Prototyping model.

🡪Using, spiral model the software is developed in a series of incremental releases.

🡪Spiral model overcome the drawbacks of the waterfall model.

🡪In every cycle new software is released to the customer.

🡪Software is released in multiple versions. So, it is called version control model.

🡪It is good for large, complex and high-risk projects.

**Phases of the Spiral Model:**

**1.Planning:**

**🡪**There will be communication between customer and project head.

🡪Here, we gather the requirements and define objectives.

🡪Analyses on estimated cost, schedule and required resources.

**2.Risk Analysis:**

🡪In this phase, the risk associated with the project are identified and evaluated.

🡪Design a prototype of model.

**3.Engineering & Execution:**

🡪Designer will design the product as per the final prototype.

🡪Developer will write the code and implement the code.

🡪Tester performs all the testing methods.

🡪Deploy and release the product to the customers.

**4.Evaluation:**

🡪The software is evaluated to determine if it meets the customer’s requirements and if it is of high quality.

🡪Here, the entire software is tested and UAT is also done.

🡪We will take feedback from the customers.

**Advantages of Spiral model:**

🡪Testing is done in every cycle, before going to the next cycle.

🡪Risk Handling

🡪It is good-fit for large projects.

🡪Flexibility in requirements.

🡪Iterative and Incremental approach.

🡪Customer will get use of software for every module.

**Disadvantages of Spiral Model:**

🡪Requirement changes are not allowed in between the cycle

🡪There is no testing in requirement and design phase.

🡪Time consuming.

🡪As it is too expensive.

🡪 It does not work for small projects

🡪Requires strong risk management.

**When to use the Spiral model?**

🡪Spiral model is used for larger projects.

🡪This approach is necessary, when there is a frequent release.

🡪When evaluating risks and costs.

🡪When it is appropriate to create a prototype.

**INCREMENTAL MODEL:**

🡪In incremental model, requirements are divided into multiple modules of the software development life cycle.

🡪Each module goes through the requirements, design, implementation and testing phases.

🡪Each increment adds functionality until the final system is complete.

**Phases of Incremental model:**

**1.Requirement Analysis:**

🡪Business analyst and Project manager collect all the functional and non-functional requirements of customer.

🡪It is crucial role in incremental model.

**2.Design and Development:**

🡪Here, we design the software architecture.

🡪The plan is just made for the next increment, it is easy to change the requirements of the customer.

🡪The programmers write the code as per the requirements given by the customer.

**3.Testing:**

🡪In the testing phase, Tester checks the performance of the existing functionality and the added functionality as per the requirements of the customer.

🡪By using different kinds of testing methods, the testing will be done.

**4.Implementation:**

🡪Implementation phase enables the coding phase of the development system.

🡪It involves the final coding that design in the designing and development phase and tests the functionality in the testing phase.

🡪After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product.

**When to use the incremental model?**

🡪When Software team are not very well skilled or trained

🡪When the customer demands a quick release of the product.

🡪A project has long development schedule.

🡪Lowers initial delivery cost.

🡪Requirements are clearly specified and understood.

🡪Requires good planning and design.

🡪 You can develop prioritized requirements first.

**Advantages of Incremental Model:**

🡪More flexible.

🡪Easier to test and debug.

🡪Changes are easy to implement.

🡪Identification of errors are simple.

🡪Provides risk handling support, because of iterations.

**Disadvantages of Incremental Model:**

🡪Need good planning.

🡪Total cost is high.

🡪Integration challenges with multiple increments.

🡪It takes a lot of time and effort to fix an issue in one unit if it needs to be corrected in all the units.

🡪Well defined module interfaces are needed.

**PROTOTYPE MODEL:**

🡪Prototype is not a complete product, it is just a toy representation of your idea, software.

🡪It is applied when customer do not know the exact project requirements.

🡪It generates before actual development of software.

🡪Here, the project head, designer and development team take a quick decision to build a prototype.

🡪It is also known as trial, iterative and error method.

**Phases of Prototype Model:**

**1.Requirement gathering and Analysis:** In this phase, users are asked about what they except or what they want from the system.

**2.Quick Design:** This model covers the basic design of the requirement through a quick overview.

**3.Build a Prototype:** This step helps in building an actual prototype from the knowledge gained from prototype design.

**4.Initial user evaluation:** Here, the customer will tell the strengths and weakness of the design, which was sent to the developer.

**5.Refining Prototype:** If any feedback is given by the user, then by improving the client’s feedback and suggestions until the system gets approved.

**6.Implement product and maintain:** Prototyping model where the final system is tested and distributed to production.

**Advantages of Prototype Model:**

🡪Flexibility in design.

🡪Missing functionalities can be easily figured out.

🡪Errors can be detected earlier.

🡪Early feedback from the customers and stakeholders.

🡪 New requirements can be easily accommodated as there is scope for refinement.

**Disadvantages of Prototype Model:**

🡪It is a time-consuming process.

🡪Prototyping tools are expensive.

🡪 Special tools & techniques are required to build a prototype.

🡪Costly with respect to money and time.

🡪It is slow process because it takes more time for development.

**CLOUD:**

🡪’The cloud’ refers to the servers that are accessed over the internet. Cloud servers are located in data centers all over the world.

🡪The cloud is a vast online storage space where people and businesses store the files and access the files and applications from anywhere through the internet connection.

🡪The cloud also offers the services, databases, networking and software applications.

🡪The main purpose of the cloud is to provide on-demand access to computing resources and services over the internet.

**Cloud Computing:**

🡪Cloud computing means storing and accessing the data and programs on the remote servers that are hosted on the internet instead of local server.

🡪Cloud computing is also referred to as Internet-based computing, it is a technology where the resource is provided as a service through the internet to the user.

🡪There are some operations that are performed with cloud computing:

* Storage, back-up and recovery of data.
* Delivery of software on demand.
* Development of new applications and services.
* Streaming videos and audio.

**Types of Cloud Computing:**

There are 2 types:

1.Service Models

2.Deployment Models

**Deployment Models:**

There are 4 deployment models:

1.Public Cloud

2.Private Cloud

3.Hybrid Cloud

4.Community Cloud

**1.Public Cloud:**

🡪Services are shared across multiple organizations and managed by third-party providers.

🡪It comes with offering a pay-as-you-go principle for scalability and accessibility of cloud resources for numerous users.

🡪It ensures cost-effectiveness by providing enterprise-needed services.

**2.Private Cloud:**

🡪Dedicated to a single organization for more security and control (e.g., On-premise VMware, OpenStack).

🡪Offer the limited services to the limited number of people due to security reasons.

🡪It is perfect for companies which looking for security and compliance needs.

**3.Hybrid Cloud:**

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

🡪It offers flexibility in optimizing resources such as sensitive data in private clouds and important scalable applications in the public cloud.

🡪An organization may use their private cloud for some services and their public cloud for others, or they may use the public cloud as backup for their private cloud.

**4.Community Cloud:**

🡪Shared by multiple organizations with common needs (e.g., government or healthcare clouds).

🡪It can be managed internally or by a third-party provider.

🡪The infrastructure of the community could be shared between the organization which has shared concerns or tasks.

**Service Models:**

There are 4 service models:

1.Saas-Software as a Service

2.Paas-Platform as a Service

3.Iaas-Infrastructure as a Service

4.Faas-Fubction as a Service