03-02-2025

**Testing:**

* The process of identifying the error or bugs or defects.
* In another way, the process of checking whether the project or the product is satisfying the needs/requirements of client or not.

**Software Testing:**

* Software testing is a part of software development life cycle (SDLC).
* It is an activity to detect and identify the defects present in the developed software.
* It checks whether the technical requirements set by design and development are meet.
* The main objective of testing is to release quality product to client.
* And satisfies user requirements efficiently and effectively.

**Need for Software Testing:**

* To deliver a very good quality software.
* To ensure that software is bug free.
* To check whether it is satisfying the needs of the customer or not.
* To reduce the maintenance cost.
* To check whether the developed software is an user friendly or not.
* And to avoid the negative feedback from customer.

**Objectives of Testing:**

* Ensuring the correctness of project or software developed.
* Ensuring the quality of the project or product.
* Ensuring the robustness of the software developed.
* Ensuring the product is bug free before shipment or release.

**Quality:**

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

**Quality Software:**

* Which is completely bug free.
* Delivered on-time.
* User friendly.
* It needs to be maintainable.

**Product:**

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

**Project:**

If the software is developed based on the requirement of single customer then it is known as project.

**Error:**

Incorrect human action that produces a problem in the system is called error.

**Bug/Defect:**

Deviation from the expected behavior to actual behavior of the system is called defect.

**Failure:**

The deviation identified by end-user while using the system is called failure.

**Software Development Life Cycle:**

A software development life cycle model, or SDLC is a structured approach to the development of software. It is used to design, develop and test good-quality software.

It is a methodology that defines the entire process of software development step-by-step.

The goal of the SDLC model is to deliver high quality, maintainable software the meets the user’s requirements.

SDLC is a collection of these six stages:

1. Planning & Requirement analysis
2. Defining requirements
3. Design
4. Development
5. Testing
6. Deployment & Maintenance

04-02-2025

**Software testing:**

The process to test an application that was fully developed. The goal of software tester is to find the bug.

Testing types:

1. Manual Testing
2. Automation Testing

1.**Manual Testing:**

Manual testing is software testing process where testers manually execute test cases without using automation tools.

**Advantages of manual testing:**

1. Cheaper – less in cost
2. It is accurate than automation testing procedure
3. It has quick visual feedback
4. It is ideal for testing minor changes
5. It is suitable for UI testing

**Disadvantages of manual testing:**

1. Time consumption
2. Cost – more expensive for long run
3. We cannot record the process of manual testing
4. Slower execution speed

2.**Automation Testing:**

Automation testing is software testing process where test cases are executed using automated tools and scripts instead of manual effort. It helps speed up testing, reduce human effort and improves test coverage.

Common tool used – Selenium for automation test.

**Advantages of automation testing:**

1. Faster execution - automated tests run faster than manual testing.
2. Higher accuracy - eliminates the human errors and ensure consistent result.
3. Better test coverage – can test multiple scenarios and large datasets efficiently

**Disadvantages of automation testing:**

1. High initial cost – requires investment in tools and script development.
2. Limited to predefined scenarios – cannot adapt like manual testers for unpredictable issues.
3. Maintenance overhead – scripts need frequent updates when the application changes.

**Categories/Techniques of software testing**:

There are classified into two types:

1. Static Testing
2. Dynamic Testing

1.**Static Testing:**

Testing done without executing the code. It involves reviewing code, design, and documentation to find errors early. The purpose is to prevent defects before implementation.

2.**Dynamic Testing:**

Testing done by executing the code to check functionality and behavior. The purpose is to detect defects that appear during runtime.

Dynamic Testing is classified into two types:

1. White box testing
2. Black box testing

1.**White box testing:**

White box testing (is also known as open box or clear box testing) is a software testing technique where the internal code, structure and logic of the application are tested. The tester has full knowledge of the code and verifies its flow, conditions, loops, and functions.

2.**Black box testing:**

Black Box Testing is a software testing technique where the functionality of an application is testedwithout knowing its internal code, structure, or logic. Testers focus only on the inputs and expected outputs without analyzing the actual code.

**Levels of Testing:**

1. **Unit Testing:** In unit testing, individual component of software is tested. The purpose of this testing is that each module is working properly or not.
2. **Integration Testing:** In integration testing individual units are combined and tested as group.
3. **System Testing:** In this testing, we can test whole application.
4. **Acceptance Testing:** A level of software testing in which software is tested for user acceptance. Ensures the software meets business requirements.

User Acceptance Testing (UAT) are classified into two types Alpha Testing and Beta Testing that ensure software quality before full release.

**Alpha Testing:**

Testers will test the application software in the presence of the customer. (or) Alpha testing is a pre-release testing phase where the software is tested internally by the development team before being released to real users.  
The purpose is to identify bugs, usability issues, and performance problems early.

**Beta Testing:**

Testing done by customers to check whether software is working properly or not. (or) Beta testing is a real-world testing phase where the software is released to a limited group of externalusers before the final launch.  
The purpose is to collect feedback from real users in different environments.

1. **Smoke Testing:** Basic test to check if the application is stable for further testing.

Difference between Bug and Defect:

|  |  |
| --- | --- |
| **Bug** | **Defect** |
| 1.Identified by the testers. | 1.Identified by the developers |
| 2.Fault in the software. | 2.Variance of the output. |
| 3. Found in development and testing phases. | 3.Found in production or user environment. |

Difference between Error and Failure:

|  |  |
| --- | --- |
| **Error** | **Failure** |
| 1.If a program can’t run or be compiled during development. | 1.If an end-user discovers an issue with the software. |
| 2. A mistake in the code made by a developer. | 2. The incorrect behavior of the software due to an error. |
| 3.Detected by developers during coding or testing. | 3.Detected by testers, users, or customers after deployment. |

**05-02-2025**

**Software Development Life Cycle:**

A software development life cycle or SDLC is a structured approach to the development of software.

It is used to design, develop and test good quality software.

It is a methodology that defines the entire procedure of software development step-by-step.

The goal of the SDLC life cycle model is to deliver high quality, maintainable software that meets the user’s requirements.

Why we use SDLC:

Software Development Life Cycle is used to ensure that software is developed systematically, efficiently and with high quality.

**SDLC Phases:**

1. Requirement Gathering
2. Analysis
3. Designing
4. Development
5. Testing
6. Deployment
7. Maintenance

**Requirement Gathering:**

* It involves collecting, documenting, and managing the requirements that define the features and functionalities of a system or application. The success of a project often depends on the accuracy and completeness of the gathered requirements in software.
* Business Analyst and Project Manager are responsible for requirement gathering. Business analyst is the primary person responsible for collecting, analyzing, and documenting requirements and project manageroversees the process and ensures alignment with business goals.

**Analysis:**

* The Analysis phase in SDLC is where the gathered requirements are thoroughly examined to ensure they are clear, complete, and feasible.
* This phase helps in defining the project scope, identifying potential risks, and preparing for system design.
* In this phases all the gathered requirements are documented as software requirement specifications (SRS).

**Designing:**

* The Design Phase in the SDLC is where the system architecture and technical solutions are planned based on the Software Requirement Specification (SRS).
* System Architects are responsible for designing phase.

**Development:**

* The Development Phase in the SDLC is where the actual coding and implementation of the software take place.
* This phase transforms the design specifications into a functional product by writing code, integrating modules, and building the system as per the defined requirements.
* Software Developers are responsible for development phase.

**Testing:**

* The Testing Phase in the SDLC ensures that the developed software is free of bugs, meets the requirements, and works as expected before deployment.
* It involves executing test cases, identifying defects, and verifying that the software performs correctly under various conditions.
* Software Testers and Quality Assurance team are responsible for testing phase.

**Deployment:**

* The Deployment Phase in the SDLC is when the developed software is released to users. This phase involves making the software available for use, either by installing it on servers or distributing it to end-users.
* DevOps Engineers are primarily responsible for automating deployment processes, managing servers, and maintaining cloud infrastructure.

**Maintenance:**

* The Maintenance Phase in the SDLC involves updating, improving, and fixing the software after it has been deployed to production.
* It ensures that the software remains functional, secure, and up-to-date with evolving requirements over time.
* This phase typically includes bug fixes, performance optimizations, adding new features, and addressing any issues discovered after deployment.

**Waterfall Model:**

* Waterfall model is a linear and sequential software development methodology where each phase must be completed before moving to the next.
* It follows a step-by-step approach.
* Testing will be done only after completely the software was developed.
* It is suitable for only smaller projects.
* The requirements in waterfall model are fixed and works well when requirements are clear from the start.
* It is difficult to measure the progress of the project.
* Client involvement is very less.

**Phases of waterfall model:**

1. Requirement Gathering
2. System design
3. Implementation
4. Testing
5. Deployment
6. Maintenance

**Requirement Gathering:**

In requirement gathering phase, all requirements needed for the software are collected from the stakeholders. And detailed documentation is created (SRS).

**System Design:**

Based on the gathered requirements, a high- level design for the software is created and the planning is done by architects.

**Implementation:**

In this phase, developers write the actual code according to the design documents.

**Testing:**

After coding, the software is tested to identify any defects or issues. Testers will perform different types of testing and produces relevant reports.

**Deployment:**

The software is deployed to the production environment and made available to users. It is released after thorough testing and approval.

**Maintenance:**

After deployment, any issues, bugs, or necessary updates are addressed. Maintenance continues for the life of the software, ensuring that it remains functional and meets user needs.

**Advantages of Waterfall model:**

1. Easy to manage - since the process is linear, it’s easier to manage and predict progress.
2. Ideal for small projects -works well for projects with clear, unchanging requirements.
3. Well-defined requirements-all requirements are gathered upfront, making it easier to define and track progress.

**Disadvantages of Waterfall model:**

1. Inflexible to changes - once a phase is completed, you cannot go back easily to make changes.
2. Not Ideal for Complex or Long-Term Projects - it is not flexible enough for projects with uncertain or changing requirements, making it unsuitable for many modern projects.
3. Late Testing - testing occurs only after development is complete, which means defects are discovered later in the process, potentially increasing costs.

**Agile Methodology:**

* The Agile model is a flexible and iterative approach to software development, where the project is divided into small parts called sprints (1-4 weeks long).
* Unlike the waterfall model, agile allows for continuous feedback, quick changes and incremental improvements.
* It is the latest method of software development.
* Mostly used to develop large projects.
* Requirements of client is taken throughout the process.
* Agile methodology is delivering the piece of software which was develop.
* Testing is done on piece of software developed.
* Deliver the projects in early/less time.
* Development cost is very less.

**Agile has three basic principles:**

1. Customer should not need to wait till the whole software is developed.
2. Delivering the piece of software which contains some functionalities which is developed and tested.
3. We can adopt/accept the requirement changes from the customer at any point of process.

**Advantages of Agile Method:**

1. Requirement changes are allowed at any phase of process
2. Releases will be very fast (weekly).
3. Customer need not to wait for longer time.
4. Easy to adopt.

**Disadvantages of Agile Method:**

1. Lack of documentation.
2. Complex for large projects.
3. Requires strong customer involvement.
4. Difficult to measure progress.

**Agile scrum:**

* Scrum is a specific framework in the Agile methodology that provides a structured approach to help teams work together to develop and deliver high-quality software.
* It is widely used in software development and other industries to manage projects in a more flexible, iterative, and incremental way.
* Scrum organizes work into manageable chunks and emphasizes collaboration, continuous improvement, and fast, incremental delivery.

**Agile sprint meeting:**

* An Agile sprint meeting refers to a series of key meetings held during a sprint in Scrum.
* These meetings help the team stay aligned, focused, and ensure continuous improvement during the development process.

**Agile review meeting:**

* In review meeting, we have to show the work at the end of the sprint session for feedback.
* It’s a key event where the development team presents the work they've accomplished, and stakeholders provide feedback, which helps guide future work.

**Agile retrospective meeting:**

* In this meeting, there will be 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.

06-02-2025

**Spiral Model:**

* The spiral model is a risk – driven software development process that combines iterative development with systematic aspects of the waterfall model.
* It is particularly useful for large, complex and high-risk projects.
* The spiral model overcomes the drawbacks of the waterfall model by introducing iterative development, risk management and flexibility.
* In every cycle new software will be released to customer.
* Spiral model is iterative model.

**Phases of spiral model:**

1. Planning
2. Risk analysis
3. Engineering
4. Evaluation

**Planning:**

* Gathering requirements from the stakeholders.
* Defining the objectives, scope and risks.
* Estimating cost and time.

**Risk analysis:**

* Identifying and analyzing the potential risks.
* Developing prototypes, means creating a simplified version of a software to explore design ideas, test functionality or evaluate feasibility before full development.
* Make decisions on feasibility and alternative solutions.

**Engineering:**

* Design, develop and test the software.
* Implement feedback from previous iterations.
* Develop new features based on refined requirements.
* In this phase, unit testing, integration testing and system testing are involved.

**Evaluation:**

* Here entire software is tested and UAT also done.
* Decide whether to continue, refine or end process.

**Advantages of Spiral model:**

1. Risk management – the spiral model includes risk analysis at every phase, reducing project failures.
2. Requirement changes are allowed after every cycle before going to the next cycle.
3. Testing is done in every cycle.
4. Early detection of issues.

**Disadvantages of spiral model:**

1. Requirement changes are not allowed in between the cycle.
2. Every cycle of spiral model looks like waterfall model.
3. There is no testing in requirement and design phase.

**Incremental model:**

* Incremental model is a process of software development where requirements will be divided into multiple modules of the SDLC.
* In this model, each module goes through the requirements, design, development, testing, and maintenance phases.
* Every subsequent release of the module adds function to previous release.
* The process continues until the complete system achieved.

**Phases of the Incremental Model:**

* 1. Requirement Analysis:
* The overall system requirements are divided into smaller, independent modules.

1. Design & Development:

* Each module is designed, developed, and tested separately.

1. Implementation:

* The first increment (basic system) is delivered. Further increments enhance the system.

1. Testing:

* Each increment is tested individually and as part of the system.

1. Deployment and feedback:

* Customers provide feedback, and the system is refined in later increments.

**Advantages of incremental model:**

1. We can use this for lengthy developed projects.
2. When the requirements are changing frequently.

**Disadvantages of incremental model:**

1. No proper planning
2. Cost is high
3. Not suitable for rapid development – it takes time to build the full system.

**Prototype model:**

The prototype model is a software development approach where a basic working version (prototype) of the system is built first to gather feedback before developing the final product.

Phases of prototype model:

1. Requirement gathering and analysis:

* Initial requirements are collected, focusing on key functionalities.

1. Design:

* A simple design is created for the prototype with only essential features.

1. Prototype development:

* A working model (prototype) is developed, usually with limited functionality.

1. Evaluation and feedback:

* Stakeholders review the prototype and provide feedback for improvements.

1. Refinement and iteration:

* Based on feedback, the prototype is modified, improved or redesigned.

1. Final product and development:

* Once the prototype is approved, full – scale development begins, incorporating final features.

**Advantages of Prototype Model:**

1. Flexibility in design.
2. Missing functionalities can be easily figured out.
3. Errors can be detected earlier.
4. New requirements can be easily accommodated as there is scope for refinement.

**Disadvantages of Prototype Model:**

1. It is a time-consuming process.
2. Prototyping tools are expensive.
3. Special tools & techniques are required to build a prototype.
4. Costly with respect to money and time.
5. It is slow process because it takes more time for development.

**Cloud:**

Cloud refers to a network of remote servers that store, manage, and process data instead of using a local computer or physical storage. It allows users to access services and resources over the internet anytime, anywhere.

**Cloud computing:**

Cloud computing is the on-demand delivery of computing resources (like servers, storage, databases, networking, software, and analytics) over the internet.

Instead of maintaining physical hardware or software, users can access these services remotely through the cloud.

**Types of cloud computing:**

1. Service models
2. Deployment models

**Deployment models:**

1. **Public cloud:**

* A public cloud is cloud computing in which the infrastructure and services are owned and operated by a third-party provider and made available to the public over the internet.
* The public can access and use shared resources, such as servers, storage, and applications and the main thing is you pay for what you used.
* Eg: AWS, Microsoft Azure.

1. **Private cloud:**

* A private cloud is a cloud computing environment in which the infrastructure and services are owned and operated by a single organization, for example, a company or government, and it is accessed by only authorized users within that organization.
* Private cloud organizations have their own data center.
* Private cloud provides a higher level of security.

1. **Hybrid cloud:**

* A hybrid cloud is a combination of both public and private cloud environments that allows organizations to take advantage of the benefits of both types of clouds.
* It manages traffic levels during peak usage periods. It can provide greater flexibility, scalability, and cost-effectiveness than using a single cloud environment.

1. **Community cloud:**

* A community cloud is shared among multiple organizations with commonrequirements (e.g., security, compliance, industry regulations).
* It can be managed by one of the organizations or a third-party provider.