

Assignment-1

- What is SDLC?

Ans: SDLC Means Software Development Life Cycle. The software development life cycle (SDLC) is the process of planning, writing, and modifying software. There are the steps of SDLC software development life cycle

- 1] Requirements Gathering
- 2] Requirements Analysis
- 3] Design
- 4] Implementation
- 5] Testing
- 6] Maintenance

- What is software testing?

Ans: The Process Which You Can Correct The Application And Complete The Process And Check The Quality. That's call Software Testing.

- What is agile methodology?

Ans: Agile Is Combination Of iterative And Incremental Process Model Agile. working Rapidly

- What is SRS?

SRS Means Software Requirements Specification(SRS).

Software Requirements Specification(SRS) Is Complete Description Of The Behavior Of The System To Be Developed.

- What is oops?

Ans: The Basic Concept Of OOPS IS To Create Objects Re-Use Them Throughout The Program And Manipulate These Objects To Get Results

Object oriented Programming Is About Data And Methods

- Write Basic Concepts of oops

Ans : Objects

Class

Abstraction

Inheritance

Polymorphism

Encapsulation

- What is class

Ans: When You Define a Class You Define a Blueprint For An Object

This Doesn't Actually Any Data But It Does Define What The Class Name Means That is What An Object Of The Class Will Consist Of And What Operation Can Be Performed On Such An Object

- What is encapsulation

Ans: Encapsulation Is The Process Of Including In An Object Every Thing Needs Hidden From Other Objects The Internal State Is Usually Not Accessible By Other Objects.

Encapsulation Enables Data Hiding ,Hiding Irrelevant Information From The User Of a Class And Exposing Only Relevant Details Required By The User.

- What is inheritance

Ans: Inheritance Means One Class Inherits Characteristics OF another Class

This is very Useful Concept of OOPS Since This Feature Reduce The Code Size
Code Reusability Can Be Achieved Though This Concept.

- What is polymorphism

Ans: There are tow types of polymorphism

1) OVERLOADING

2) OVERRIDING

- Write SDLC phases with basic introduction

Ans: SDLC Means Software Development Life Cycle.

Thare are 6 Phases of Software Development Life Cycle (SDLC).

1)Requirement Gathering

2) Analysis Phase

3) Design Phase

4) Implementation Phase

5) Testing Phase

6) Maintenance Phase

1) Requirement Gathering: First Phase Of SDLC model is Collect Requirement From Client.

Requirement Gathering Stage Need Teams To Get Detailed And Precise Requirement

2) Analysis Phase: Analysis Phase Defines The Requirements Of The System Independent Of How These Requirements Will Be Accomplished.

Details On Computer Programming Languages And Environments Machines ,Packages, Application Architecture, Distributed Architecture Layering ,Memory Size Platform Algorithms Data Structures Global Type Definitions ,Interfaces And Many Other Engineering Details Are Established.

3) Design Phase: Design Architecture Document.

Implementation Plan

Critical Priority Analysis

Performance Analysis

Test Plan

The Design Team Can Now Expnd Upon Information Established In The Requirement Document

4) Implementation Phase: Implementation Start Once The Developer Gets The Design Document The Software Design Is Translated Into Source Code All The Component Of The Software Are Implemented In This Phase.

5) Testing Phase: Once The Software Is Complete And It Is Deployed In The Testing Environment The Testing Team Starts Testing Functionality Of The Entire System

This Is Done To Verify That The Entire Application Works According To The Customer Requirement.

6) Maintenance Phase: Maintenance Is The Process Of Changing A System After It Has Been Deployed.

Corrective Maintenance : Identifying And Repairing Defects

Adaptive Maintenance : Adapting The Existing Solution To The New Platforms

Perfective Maintenance : Implementing The New

- Explain Phases of the waterfall model:

Ans: **Requirement Gathering Stage:**

During this phase detail requirements of the software system to be developed are gathered from client. And Requirements are fixed there are small projects.

Design Stage:

Plan the programming language eg JAVA, PHP, .NET
Or database like Oracle, MySQL, etc.
Or other high level technical details of the projects.

Built Stage:

After design stage it is built stage, that is nothing but coding the software.

Test Phase:

In this phase you test the software to verify that it is built as per the specifications given by the client.

Deployment Phase:

Deploy the system in the respective environment.

Maintenance Phase:

Once your system is ready to use you may later require change the code as per customer request.

- Write phases of spiral model

Ans: **Planning**

It includes estimating the cost, schedule and resources for the iteration. It also involves understanding the system requirements for continuous communication between the system analyst and the customer.

Risk Analysis:

Identification of potential risk is done while risk mitigation strategy is planned and finalized.

Engineering:

It includes testing, coding and deploying software at the customer site.

Evaluation:

Evaluation of software by customer also includes identifying and monitoring risks such as schedule slippage and cost overrun.

- Explain working methodology of agile model and also write pros and cons.

Ans: Agile SDLC Model Is Combination Of Iterative And Incremental Process Model Which Focus On Process Adaptability And Customer Satisfaction By Rapid Delivery Of Working Software Product.

At The End Of The Iteration A Working Product Is Displayed To The Customer And Important Stakeholders

Pros OF Agile Model:

It Is Very Realistic Approach To Software Development

Functionality Can Be Developed Rapidly And Demonstrated

Resource Requirements Are Minimum

Suitable For Fixed and Changing Requirements

Minimal Rules ,Documentation Easily Employed

Little Or No Planning Required Easy To Manage Gives Flexibility To Developers.

Cons of Agile Model:

Not Suitable For Handling Complex Dependencies

More Risk Of Sustainability, Maintainability ,Extensibility

There Is Very High Individual Dependency, Since There Is Minimum Documentation Generated

Transfer Of Technology To New Team Member May Be Quite Challenging Due To Lack Of Documentation

