**day01\_107856406\_dsdipt\_sudipto\_14may2025**

**Employee Code:** 107856406

**Login ID:** dsdipt

**Email :** dsdipt@amazon.com

**Name:** Sudipto Das

**Date:** 14 May 2025 (Day 01)

### ***Task 1: What is SDLC?***

SDLC stands for Software Development Life Cycle. It is a structured process used for developing software efficiently and with quality. It includes several phases such as: **Planning, Analysis, Design, Development, Testing, Deployment, Maintenance**. Each phase helps ensure the software meets user requirements and works reliably.

### ***Task 2: Why is SDLC required?***

**Structured Development** – It provides a clear roadmap for software creation.

**Better Planning** – Helps in managing time, cost, and resources efficiently.

**Risk Reduction** – Identifies and resolves problems early in the process.

**Quality Assurance** – Ensures the final product meets user needs and is bug-free.

**Team Coordination** – Improves communication and role clarity among team members.

**Customer Satisfaction** – Leads to delivering software that fulfills user expectations.

### ***Task 3: What are the different steps of the SDLC?***

**Stage 1: Planning and Requirement Analysis**

This is the foundational phase where the project’s feasibility is evaluated, and its scope is outlined. It involves gathering high-level requirements, understanding business needs, and identifying key stakeholders.

**Key Activities:**

* Initial planning
* Requirement analysis
* DevOps alignment for early infrastructure and pipeline planning

**Stage 2: Define the Requirements**

Detailed documentation of functional and non-functional requirements happens in this stage. The requirements are broken down into tasks and planned across development cycles.

**Key Activities:**

* Define clear and actionable requirements
* Plan sprints: number of sprints and duration based on the overall timeline of the project

**Stage 3: Designing**

In this stage, the system architecture and design specifications are developed. It includes both high-level design (HLD) and low-level design (LLD).

**Key Activities:**

* Create architectural diagrams and data flow
* Assign tasks to developers
* Plan sprints: determine how many are required for design-specific activities

**Stage 4: Development**

This is where actual coding happens. Developers work based on sprint cycles to implement features and functionalities. Agile practices like daily stand-ups are used to track progress.

**Key Activities:**

* Code implementation
* Daily scrum meetings
* Use of JIRA dashboard for task tracking and sprint management
* Maintain a To-Do List (TDL) for each sprint to manage tasks efficiently

**Stage 5: Testing and Integration**

After development, thorough testing is performed to ensure the system works as expected and integrates well. The focus is on correctness and data integrity.

**Key Activities:**

* Unit testing, integration testing, system testing
* Validate correctness of data
* Ensure data integrity and consistency across modules

**Stage 6: Deployment and Maintenance**

The final product is deployed to production. Post-deployment, ongoing maintenance, updates, and issue resolution are managed.

**Key Activities:**

* Product deployment
* Continuous monitoring and maintenance
* Rollout patches and updates as needed

### 

### ***Task 4: What are the different models of the SDLC?***

Different Models of SDLC are structured approaches to software development. Each model defines how the steps of SDLC are carried out. Here are the most common ones:

**Agile Model**

– A flexible, collaborative approach with rapid development in small increments (called sprints).

Best for: Projects that require frequent updates and user feedback.

**Waterfall Model**

– A linear and sequential approach; each phase must be completed before the next begins.

Best for: Simple, well-defined projects.

**V-Model (Validation and Verification Model)**

– An extension of the Waterfall Model where each development stage is directly associated with a testing phase.

Best for: Projects with strict requirements and high reliability.

### ***Task 5: What are the various types of networks in computer networking?***

**PAN (Personal Area Network)** A small network used for personal devices within a short range, typically a few meters.  
 Example: Bluetooth connection between a smartphone and wireless earbuds.

**LAN (Local Area Network)** A network covering a small geographic area like a home, office, or building.  
 Example: Computers connected in a school computer lab.

**WLAN (Wireless Local Area Network)** A type of LAN that uses wireless technology like Wi-Fi instead of cables.  
 Example: Home Wi-Fi network connecting laptops and smartphones.

**MAN (Metropolitan Area Network)** A network that spans a city or large campus, connecting multiple LANs.  
 Example: A university campus network spread across multiple buildings.

**WAN (Wide Area Network)** A large network that connects devices over a broad geographical area, often globally.  
 Example: The Internet is the largest example of a WAN.

**VPN (Virtual Private Network)** A secure, encrypted connection over a public network to access private networks remotely.  
 Example: An employee accessing their company's internal network from home.

### ***Task 6: What are the different types of servers?***

**Web Server:** Hosts websites and delivers web pages to users over the internet using HTTP/HTTPS protocols.  
**Application Server:** Runs and manages software applications, acting as a bridge between users and backend databases.  
**Database Server:** Stores, manages, and processes data in databases; responds to data queries from other servers or clients.

**DNS Server:** Translates domain names into IP addresses, allowing browsers to load websites correctly.  
**Proxy Server:** Acts as an intermediary between a client and the internet, often used for security, filtering, and caching.

**Game Server:** Hosts multiplayer online games, managing player connections, interactions, and game state.  
**Virtual Server:** A server created through virtualization, allowing multiple servers to run on a single physical machine.

### ***Task 7: What is DNS?***

DNS (Domain Name System) is like the **phonebook of the internet** — it translates **human-friendly domain names** (like www.google.com) into **IP addresses** (like 142.250.190.4) that computers use to identify each other on the network.

**Example:**

When you type www.amazon.com into your browser:

* **DNS resolves that name to its actual IP address.**
* **Your browser then connects to that IP to load the Amazon website.**

Without DNS, users would need to remember numeric IP addresses for every website — DNS makes the internet more accessible and user-friendly.

### ***Task 8: What is TCP & UDP?***

**TCP (Transmission Control Protocol)**

TCP is a **connection-oriented protocol**, meaning it first establishes a reliable connection between the sender and receiver before any data is transferred. It ensures that all data packets arrive **accurately**, **completely**, and **in the correct order**.

* ✅ Reliable and error-checked
* 📶 Used in: Web browsing (HTTP/HTTPS), email communication
* 🐢 Slower due to connection setup and error handling

**UDP (User Datagram Protocol)**

UDP is a **connectionless protocol** that sends data without establishing a formal connection. It does **not guarantee delivery**, order, or error checking, making it faster but less reliable than TCP.

* ⚡ Fast and lightweight
* 🎮 Used in: Video/audio streaming, online gaming, VoIP
* 🚫 No guarantee of delivery or packet order

### ***Task 9: What is a MAC address, and how is it different from an IP address?***

A **MAC address** (Media Access Control address) is a **unique identifier assigned to a network interface card (NIC)** for communications on the physical network segment. It’s like a **"serial number"** for a device's network hardware.

**MAC Address:**

* **Usually fixed** and assigned by the device manufacturer. It can be changed (spoofed), but it’s not meant to be altered.
* A MAC address identifies the **device itself**.
* Hardware address (physical)
* Identifies a device **within a local network**

**IP Address:**

* Easily configurable. It can be assigned manually (static IP) or automatically by a DHCP server (dynamic IP).
* An IP address identifies the device’s location on the network.
* Software address (logical)
* **Identifies a device across networks (internet)**

### ***Task 10: What is the OSI model? Explain.***

The OSI Model (Open Systems Interconnection Model) is a conceptual framework used to understand and implement standard protocols in computer networking. It divides the process of network communication into **7 distinct layers**, each with a specific function.

**7 Layers of the OSI Model (from top to bottom):**

**Application Layer (Layer 7)**The interface where end-users interact with network services through applications. Example: Web browsers, email clients (HTTP, SMTP).

**Presentation Layer (Layer 6)**Formats and translates data for the application layer, handling encryption, compression, and data conversion.

*Example:* Data encryption, JPEG compression.

**Session Layer (Layer 5)**Manages communication sessions by establishing, maintaining, and terminating connections between applications.

**Transport Layer (Layer 4)**Ensures reliable or fast data transfer between devices, managing error recovery and flow control.

*Protocols:* TCP (reliable), UDP (fast).

**Network Layer (Layer 3)**Determines the best path to route data packets from the sender to the receiver across networks.

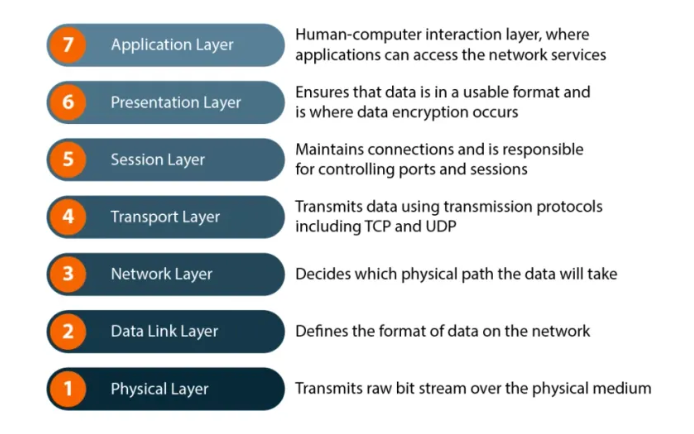
*Protocols:* IP, ICMP.

**Data Link Layer (Layer 2)**Provides error-free transmission of data frames between directly connected nodes and handles physical addressing.

*Example:* MAC addresses, Ethernet.

**Physical Layer (Layer 1)**Deals with the physical transmission of raw bits over cables or wireless media, including electrical signals and hardware devices.

*Example:* Cables, switches, electrical signals.



### ***Task 11: What is an IPv4 address, and what are its different classes?***

An **IPv4 address** is a 32-bit numerical identifier assigned to devices on a network. It helps to locate and communicate with devices over the Internet. IPv4 addresses are written in four decimal numbers separated by dots, for example, 192.168.1.1.

**Different Classes of IPv4 Addresses**

IPv4 addresses are divided into five classes based on their starting bits and purpose. **Class A** addresses range from 1.0.0.0 to 126.255.255.255 and are used for very large networks, such as major organizations. **Class B** addresses cover from 128.0.0.0 to 191.255.255.255 and are intended for medium-sized networks. **Class C** addresses, ranging from 192.0.0.0 to 223.255.255.255, are used for smaller networks like small businesses or home networks.

Classes **D** and **E** serve special purposes: Class D (from 224.0.0.0 to 239.255.255.255) is used for multicast groups, allowing one-to-many communication, while Class E (from 240.0.0.0 to 255.255.255.255) is reserved for experimental or research use and is not typically assigned to devices.

**Summary:.**

* IPv4 addresses help identify devices on a network.
* Classes A, B, and C are used for different network sizes.
* Classes D and E have special purposes (multicast and experimental).

### ***Task 12: Advantages of using VPN***

**Protection –** Hides your IP address and browsing activity from ISPs and trackers.

**Data Encryption –** Secures your data over public Wi-Fi or unsecured networks.

**Bypass Geo-Restrictions –** Access region-locked content (e.g., Netflix, websites).

**Avoid Censorship –** Bypass government or network restrictions in restricted regions.

**Improved Anonymity –** Helps mask your identity online.

**Protects from Hackers –** Reduces risk of cyberattacks, especially on public Wi-Fi.

### ***Task 13: Types of VPN***

**Access VPN** – Connects individual remote users to a private network securely.  
**Site-to-Site VPN** – Connects entire networks at different locations over the internet.

**Remote Access VPN** – Flexible, good for remote workers.  
**Intranet VPN** – Links multiple offices of the same organization using a shared network.  
**Extranet VPN** – Securely connects different organizations for shared business activities.

### ***Task 14: How would you define nodes and links in networking?***

A **node** is any device or point in a network where data can be created, received, or transmitted. Examples include computers, printers, routers, or switches. Nodes are like the “stops” or “junctions” in a communication system.

A **link** is the communication pathway or connection between two nodes. It allows data to travel from one node to another. Links can be physical cables (like Ethernet or fiber optics) or wireless connections (like Wi-Fi or Bluetooth).

**Networking** is the practice of connecting multiple nodes together using links to share data and resources. It allows devices to communicate, access shared information, and work together efficiently, whether in a small home setup or a large global system like the internet.

### ***Task 15: What is network topology?***

**Network topology** refers to the physical or logical arrangement of nodes and links in a computer network. It defines how devices (nodes) are connected and how data flows between them.

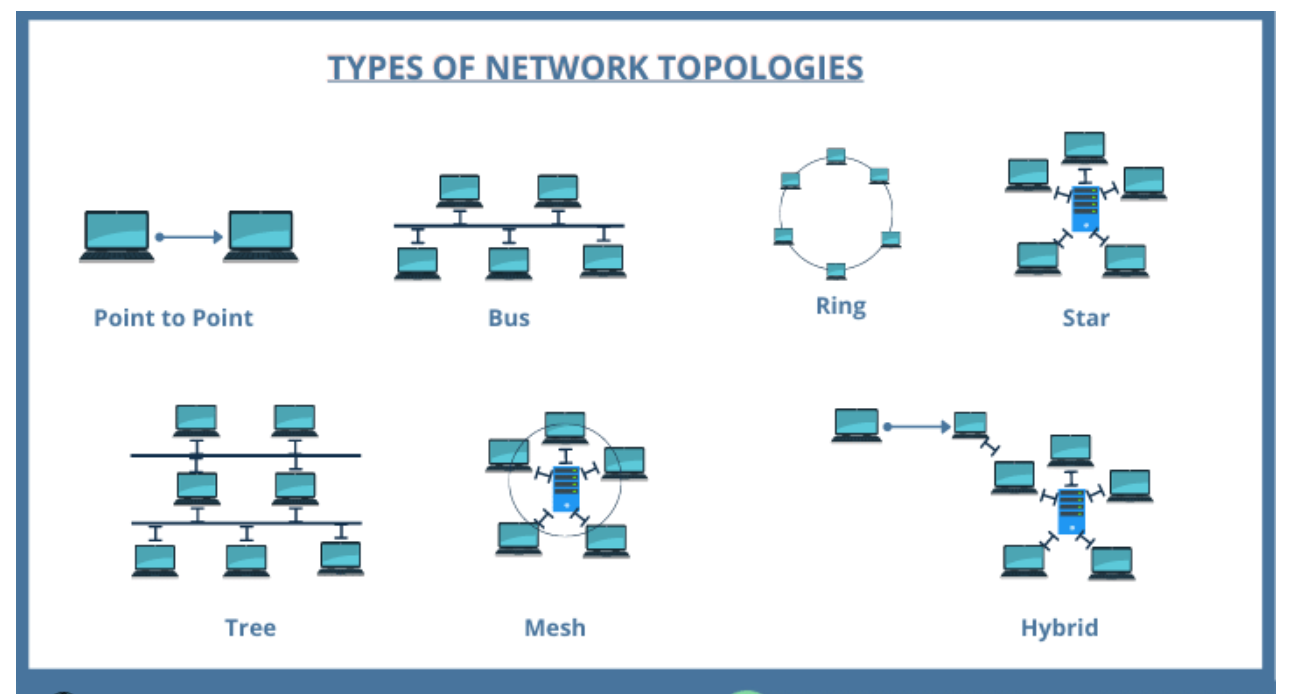
### ***Task 16: What are the different types of network topology?***

**Bus Topology** – All devices share a single communication line (like a bus line).  
 *Example:* Small office networks.

**Star Topology** – All devices are connected to a central hub or switch.  
 *Example:* Home Wi-Fi networks.

**Ring Topology** – Devices are connected in a circular loop.  
 *Example:* Older LAN setups.  
**Mesh Topology** – Every device connects to every other device (fully or partially).  
 *Example:* Military or critical networks.  
**Tree Topology** – Hierarchical mix of bus and star topology.  
 *Example:* Large enterprise networks.

**Hybrid Topology** – Combination of two or more topologies.



### ***Task 17: What is extended bus topology?***

Extended Bus Topology is a variation of the standard bus topology where additional bus lines are branched off the main central bus. It connects multiple nodes to the main bus and may include repeaters to extend the network range.

**Key Features:**

* Builds on the simple bus topology by adding branches (or "spurs").
* Uses terminators at both ends of the main bus to prevent signal reflection.
* Often includes repeaters or hubs to maintain signal strength over longer distances.

**Advantages:**

* More flexible than a basic bus topology.
* Can connect more devices over a wider area.
* Relatively easy to expand.

**Disadvantages:**

* If the main bus fails, the entire network goes down.
* More complex than a basic bus.
* Performance can still degrade as more devices are added.

### 

### ***Task 18: What is the use of a router and how is it different from a gateway?***

**Router**

Definition: A router connects multiple networks and directs data between them, usually between a LAN and the internet.

**Main Uses:**

* Routes traffic between networks
* Assigns IPs via DHCP
* Provides firewall and NAT
* Supports wired and wireless connections

**Example:**  
 At home, your router assigns your device an IP and handles data between it and the internet.

**Gateway**

Definition: A gateway serves as an entry/exit point between networks, often translating between different protocols.

**Main Uses:**

* Connects networks using different protocols
* Translates data formats (e.g., TCP/IP to MPLS)
* Acts as a default route for network devices

**Example:**  
 In home networks, the router often acts as a gateway between your LAN and the internet.

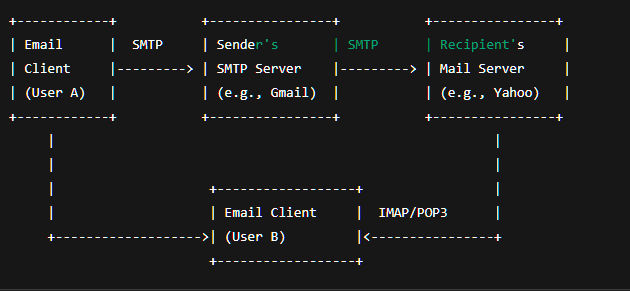
### ***Task 19: Can you explain the SMTP protocol with the help of a diagram?***

**Definition:**

SMTP stands for Simple Mail Transfer Protocol. It is a standard protocol used to send emails from a client (like Outlook or Gmail) to a mail server and between mail servers. It works over TCP port 25 (or 587 for encrypted connections).

**How SMTP Works (Step-by-Step):**

* User sends an email using an email client (e.g., Outlook, Gmail).
* The client connects to the SMTP server (like smtp.gmail.com).
* SMTP sends the email to the recipient's mail server.
* The recipient retrieves the email using IMAP or POP3.

****

### ***Task 20: What is the difference between the OSI and TCP/IP models?***

**OSI Model (Open Systems Interconnection):**

* Has 7 layers: Application, Presentation, Session, Transport, Network, Data Link, Physical.
* Theoretical/conceptual model used for teaching and standardization.
* Layers are clearly separated with distinct functions.
* Protocol-independent model.
* Less commonly used in practical networking.

**TCP/IP Model (Transmission Control Protocol/Internet Protocol):**

* Has 4 layers: Application, Transport, Internet, Network Access.
* Practical/implementation model used in real-world networks (like the internet).
* Combines some OSI layer functions (e.g., Presentation and Session layers are part of Application layer).
* Protocol-dependent — built around TCP, IP, and other protocols.
* Forms the basis of modern internet communication.

**Key Differences:**

* **Layers:** OSI has 7, TCP/IP has 4.
* **Use:** OSI is for reference/design, TCP/IP is used in practice.
* **Structure:** OSI separates layers strictly, TCP/IP combines some.
* **Development:** OSI by ISO; TCP/IP by U.S. DoD.
* **Protocols:** OSI is general; TCP/IP is protocol-specific.

### ***Task 21: What is the difference between HTTP and HTTPS?***

**HTTP (HyperText Transfer Protocol)** and **HTTPS (HyperText Transfer Protocol Secure)** are both protocols used for transferring data over the web, but they differ mainly in terms of **security**.

HTTPS is a **more secure** version of HTTP, encrypting the data sent between your browser and the server. It's essential for websites that handle **sensitive information**.

### ***Task 22: Low Level Design (LLD) and High Level Design (HLD)?***

**HLD (High-Level Design):**

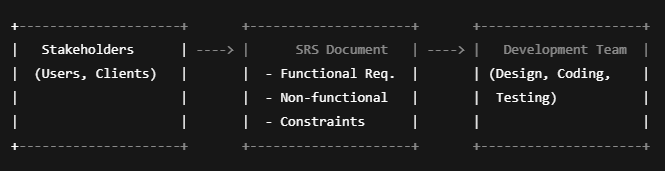
* Overview of system architecture
* Focus on modules and their interaction
* Used by architects and project leads
* Includes tech stack, data flow, and interfaces
* Less technical, more abstract

**LLD (Low-Level Design):**

* Detailed design of each module
* Focus on logic, classes, functions, and DB structure
* Used by developers
* Includes class diagrams, pseudocode, APIs
* More technical and implementation-ready

### ***Task 23: What is SRS (Software Requirements Specification)?***

SRS is a detailed document that defines the software’s functional and non-functional requirements. It serves as a blueprint for developers, testers, and stakeholders. SRS outlines what the system should do, how it should perform, and any constraints. It helps ensure everyone has a clear and common understanding of the project requirements.



### 

### ***Task 24: SDLC MCQ***

**A feasibility study using the SDLC model is conducted to**

* determine whether or not the project is technically possible
* determine whether the proposal is financially viable
* ~~Both a and b~~
* None of the above

**A well-documented life cycle model aids in the detection of what during the development phase?**

* Inconsistencies
* Redundancies
* Omission
* ~~All of the above~~

**How many lines of code does the Build & Fix Model suit for programming exercises?**

* ~~100-200~~
* 300-400
* 600-700
* Above 800+

**In which life cycle does regression testing play a significant role?**

* Waterfall model
* V model
* Iterative model
* ~~All of the above~~

**What determines if the project should go forward?**

* ~~feasibility assessment~~
* opportunity identification
* system evaluation
* program specification

**What is the most significant disadvantage of employing the RAD Model?**

* ~~Developers/designers that are highly specialized and skilled are required.~~
* Component reusability is improved.
* Encourages client/customer input.
* Increases component reusability.

**Which of the following developmental models is incremental?**

* Prototyping, V model, Agile
* ~~Prototyping, RAD, Agile, RUP~~
* Prototyping, V model, RAD, Agile, RUP
* All of the above

**Which of the following is an Agile development characteristic?**

* Shared code ownership
* Test-Driven Development
* Implement the simplest solution to meet today's problem
* Continual feedback from customer
* ~~All of the above~~

**Which of the following steps in the SDLC framework are valid?**

* Requirement Gathering
* Software Design
* System Analysis
* ~~All of the above~~

**Who is in charge of system development, staffing, budgeting, and reporting, as well as ensuring that deadlines are met?**

* ~~Project managers~~
* Network engineers
* Graphic designers
* Systems analysts

### ***Task 25: Miscellaneous***

**List Top SDLC tools**

**Jira**: This software is intended to make workflow management easier for a wide range of groups. Jira was created with the intention of being a simple system for recording tasks and errors. However, it has since matured into a robust workflow management solution.

**Git** is a distributed version management system that is open-source. Developers aiming to examine changes and contributions to the overall code might considerably benefit from a version control system or VCS. This software customisation management tool is an important part of the SDLC.

**Confluence**: During this stage, Confluence is a wonderful tool for developing product research docs and sharing design assets.

**Asana**: From daily activities to larger projects, Asana assists teams in orchestrating their work. Teams are more confident, move faster, and accomplish more with less when they use Asana, regardless of where they are based.

**What is Software Configuration Management, and how does it work?**

The process of tracking and regulating changes that occur during the software development life cycle is known as software configuration management. Any modification made during the development of software must be tracked using a well-defined and controlled process. Any modifications performed during software development are regulated through a well-defined process, thanks to configuration management. Revision control and the establishment of baselines are two SCM procedures.

**What do a Software Project Manager's responsibilities entail?**

The Software Project Manager is in charge of seeing the project through to completion. The Software Project Manager is responsible for ensuring that the entire team follows a methodical and well-defined approach to software development. They also handle project planning, tracking project status, resource management, and risk management.

**What do you know about Scrum impediments?**

Obstacles or challenges that the scrum team faces slow down their work speed are referred to as impediments. An obstacle is anything that tries to prevent the scrum team from getting work "Done." Impediments can take many different forms. Some of the roadblocks include resource shortages or sick team members, technical, operational, and organisational issues, a lack of management support systems, and business issues.

**Briefly explain Scrum methodology in the Agile model.**

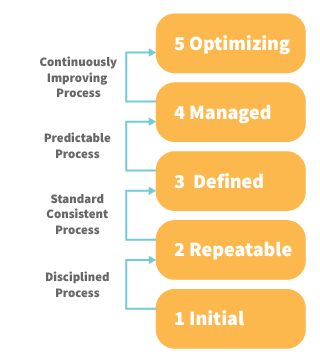
Scrum is an agile development approach based on iterative and incremental procedures that are used in the creation of software. It's an agile structure that's adaptable, rapid, flexible, and excellent at delivering value to customers throughout the project's development. Companies of all sizes employ the Agile Scrum technique because of its ability to provide high-end cooperation and efficiency for project-based work. Scrum is a sort of agile approach that breaks projects down into manageable parts known as "sprints." The Agile Scrum methodology is ideal for companies who need to complete projects fast.

**What are Capability Maturity Model(CMM) levels?**

**Following are the five Capability Maturity Model Levels:**

* Initial: The first step is to create an unstable process environment. The software development process is considered haphazard and even chaotic at times. There are few methods that have been specified, and success is based on individual effort and heroism.
* Repeatable: Work is planned and monitored, making it repeatable. To track cost, schedule, and functionality, basic project management techniques are implemented.
* Defined: This level encompasses written and defined standards that evolve over time and support consistent performance. The work is well defined at this point.
* Managed: Extensive data on the software development process and product quality are gathered. Both the software development process and the end products are quantified and managed.
* Optimized: Work is based on continuous improvement (optimization). The focus on continuously improving process performance is a significant feature of this level.

**What is the Capability Maturity Model?**



The Capability Maturity Model (CMM) is a cross-discipline and technical paradigm for facilitating and refining software development processes and system improvement. This methodology is at the heart of most management systems that aim to improve the quality of all product and service development and delivery.

**What is Level-0 DFD?**

Context Diagram is another name for DFD Level 0. It's a high-level overview of the entire system or process that's being studied or modelled. It's meant to be a quick peek into the system, displaying it as a single high-level process with its connections to external entities. Stakeholders, business analysts, data analysts, and developers should all be able to understand it readily.

**How can DDLC and SDLC work together?**

The DDLC (Documentation Development Life Cycle) is a software documentation development life cycle used by technical documenters to prepare software documentation. The life cycle is followed in tandem with the SDLC, as testers and developers work on the programme at the same time. Because the documentation requires input and feedback from the various phases of the SDLC, the DDLC has stages that are comparable to the SDLC.

**What are different types of prototype models?**

There are four types of Prototyping models:

* Rapid Throwaway prototypes.
* Evolutionary prototype.
* Incremental prototype.
* Extreme prototype.

**What is the FRS document?**

This document captures the user's voice from the outside, or the end user's perspective. A Business System Analyst creates it (BSA). This paper demonstrates how a system will react when a user interacts with it in order to meet the BRD and SRD standards. The key area of interest for software experts is the Functional Requirement Specification (FRS). An FRS is useful for software testers to learn the situations in which the product is intended to be tested, just as it is for developers to understand what product they are planning to produce. An FRS's ultimate purpose is to meet all of the requirements outlined in the SRS and BRS regulations.

**What is the Software release process?**

The Software Development Life Cycle (SDLC) release phase is historically connected with production, deployment, and post-production operations, which generally include software maintenance and support. So, release management is the process of managing, planning, scheduling, and controlling a full software development at every stage and environment, including testing and releasing software releases.

**What is the use of a JAD session?**

JAD is a strategy for defining business system requirements that are commonly utilised in the early phases of a systems development project. JAD's goal is to bring MIS and end-users together in a structured workshop setting in order to extract outcome system needs. It allows clients and developers to swiftly agree on a project's fundamental scope, objectives, and specifications