**Day 1**: **14th May 2025**

**Task 1:**

**What is SDLC?**

SDLC stands for Software Development Life Cycle) is a structured process used to develop software in a systematic and efficient way. It includes several phases like requirement gathering, designing, coding, testing, deployment, and maintaining. Each phase has its own purpose and helps ensure the final product is reliable, efficient, and meets the client's needs. SDLC helps teams manage time, cost, and quality effectively during software development.

**Task 2:**

**Why SDLC?**

SDLC is important because it provides clear structure for developing software. It helps teams work in an organized manner, reducing confusion and errors. By following SDLC, we can detect and fix issues early, which saves time and cost. It also ensures the final product meets the client's expectations and is easier to maintain in the long run. Overall, it improves quality, efficiency, and client satisfaction.

**Task 3:**

**Steps involved in SDLC:**

1. Requirement gathering & Analysis - Understand what the client or user needs. Analyze if it's possible to build it.

2. System Design: Plan how the software will work (architecture, UI, databases).

3. Development (Implementation/Coding) – Developers write the actual code based on the design.

4. Testing: Test the software to find and fix bugs and errors to ensure the software works as expected.

5. Deployment - Release the software to users or the production environment.

6. Maintenance: Update, fix, or improve the software after it's released.

**Task 4:**

**SDLC models:**

**Waterfall Model:**

* Sequential, linear approach where each phase must be completed before moving to the next
* Phases include requirements, design, implementation, testing, deployment, and maintenance
* Best suited for projects with clear, unchanging requirements
* Limited flexibility and difficult to accommodate changes once the project begins.  
    
    
    
  

**Agile Model:**

* Iterative approach focusing on continuous development and testing throughout the project
* Emphasizes customer collaboration, adaptive planning, and rapid delivery
* Breaks project into small incremental builds with frequent customer interaction
* Highly flexible and adaptable to changing requirements and market conditions  
    
  

**Spiral Model:**

* Combines iterative development with systematic aspects of waterfall model
* Focus on risk assessment and minimization through prototyping and continuous planning
* Four phases: Planning, Risk Analysis, Engineering, and Evaluation
* Well-suited for large, complex projects with high-risk elements



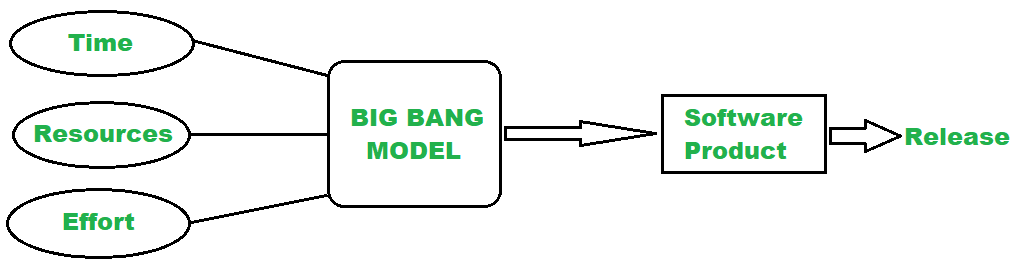
**V-Model (Verification and Validation Model):**

* Extension of waterfall model with emphasis on testing at each development stage
* Each development stage has a corresponding testing phase
* Highly disciplined model with strict validation at each stage
* Best suited for projects requiring high reliability and early testing  
    
  

**Iterative Model:**

* Starts with simple implementation of system requirements
* Progressively gains more complex functionality with each iteration
* Allows for refinement through multiple development cycles
* Good for projects where requirements are clearly defined but complex implementation  
    
  

**Big Bang Model:**

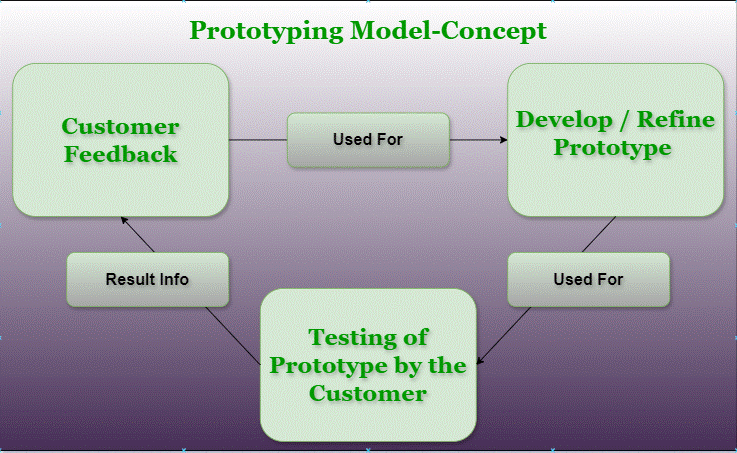
* Very simple model with minimal planning and process
* Resources are focused on development with little formal methodology
* Suitable for small projects with one or two developers
* High risk and uncertainty in final outcome  
    
  

**RAD (Rapid Application Development) Model:**

* Focuses on rapid prototyping and quick feedback rather than long planning
* Emphasizes quick development and delivery of high-quality systems at low cost
* Uses time-boxed, incremental delivery, and reusable components.
* Best for projects requiring quick delivery and where requirements can change  
    
  

**Prototyping Model:**

* Begins with basic prototype creation based on initial requirements
* Involves continuous refinement based on customer feedback
* Helps in understanding exact requirements through working models
* Useful when client requirements are unclear or constantly changing



These models can be adapted or combined based on specific project needs, team size, complexity, and organizational requirements.

**Task 5:**

**What are the different Network types?**

The main types of networks:

PAN (Personal Area Network)

* Coverage area: Within personal space (1-10 meters)
* Examples: Bluetooth connections, wireless keyboards/mouse
* Used for personal devices communication
* Common technologies: Bluetooth, Infrared, NFC

LAN (Local Area Network)

* Coverage area: Limited geographical area (building/campus)
* Connects computers within an organization
* High data transfer rates
* Common technologies: Ethernet, Wi-Fi

MAN (Metropolitan Area Network)

* Coverage area: City-wide network
* Connects multiple LANs across a city
* Used by organizations with multiple locations in a city
* Example: Cable TV networks

WAN (Wide Area Network)

* Coverage area: Countries or continents
* Connects networks over large geographical areas
* Internet is the largest example of WAN
* Uses telecommunication links

CAN (Campus Area Network)

* Coverage area: University/corporate campus
* Connects multiple LANs within campus
* Higher speed than WANs
* Typically owned by organization

SAN (Storage Area Network)

* Dedicated high-speed network for storage
* Connects servers to storage devices
* High-performance data transfer
* Used in data centers

VPN (Virtual Private Network)

* Creates secure connection over public network
* Provides encrypted communication
* Enables remote access to private networks
* Used for secure remote working

WLAN (Wireless Local Area Network)

* Wireless version of LAN
* Uses radio waves for communication
* Common in homes and offices
* Example: Wi-Fi networks

HAN (Home Area Network)

* Network within a home
* Connects home devices and appliances
* Can include smart home devices
* Usually combines wired and wireless

DAN (Desk Area Network)

* Very small network around desk
* Connects peripheral devices
* Similar to PAN but more localized
* Example: Computer and its peripherals

**Task 6:**

**What are the types of servers?**

Main types of servers:

1. Web Servers

* Hosts websites and handles HTTP/HTTPS requests from clients
* Examples: Apache, Nginx, IIS
* Delivers web content to browsers and manages web traffic

2. Application Servers

* Hosts and runs applications and business logic
* Manages application operations between users and backend
* Examples: WebLogic, WebSphere, JBoss

3. Database Servers

* Stores, manages, and retrieves structured data
* Handles database queries and maintains data integrity
* Examples: MySQL, Oracle, SQL Server, PostgreSQL

4. File Servers

* Centralizes file storage and sharing within a network
* Manages file access permissions and backup
* Provides secure and organized file storage solution

5. Mail Servers

* Handles email sending, receiving, and storage
* Manages email accounts and spam filtering
* Examples: Microsoft Exchange, Postfix, Send Mail

6. DNS Servers

* Translates domain names to IP addresses
* Maintains domain name hierarchy
* Essential for internet navigation and web browsing

7. Proxy Servers

* Acts as intermediary between users and internet
* Provides caching, security, and anonymity
* Can help improve network performance and security

8. FTP Servers

* Manages file transfers between clients and servers
* Provides protocol for uploading and downloading files
* Examples: FileZilla Server, vsftpd

9. Print Servers

* Manages print jobs and printer access
* Connects multiple users to shared printers
* Handles print queues and printer configurations

10. Game Servers

* Hosts multiplayer online games
* Manages game state and player interactions
* Handles real-time game communications

11. DHCP Servers

* Automatically assigns IP addresses to network devices
* Manages network configuration parameters
* Essential for network address management

12. Authentication Servers

* Verifies user identities and access rights
* Manages user credentials and permissions
* Examples: LDAP, Active Directory

13. Backup Servers

* Stores and manages data backups
* Handles automated backup schedules
* Provides disaster recovery capabilities

14. Media Servers

* Stores and streams multimedia content
* Manages digital media libraries
* Examples: Plex, Emby, Netflix servers

15. Virtual Servers

* Runs multiple virtual machines on single hardware
* Provides isolated server environments
* Enables efficient resource utilization

16. Cloud Servers

* Hosted on cloud infrastructure
* Provides scalable computing resources
* Examples: AWS EC2, Google Cloud, Azure VMs

17. NTP Servers

* Provides network time synchronization
* Maintains accurate time across network devices
* Essential for logging and security

18. VoIP Servers

* Manages voice over IP communications
* Handles call routing and management
* Examples: Asterisk, FreePBX

19. Collaboration Servers

* Enables team communication and collaboration
* Manages shared resources and workspaces
* Examples: Microsoft Teams, SharePoint

20. Security Servers

* Manages network security
* Includes firewalls and intrusion detection
* Monitors and protects network resources

**Task 7:**

**What do you know about DNS? Domain Name**

DNS or Domain Name System is basically the internet's phone book that converts website names into IP addresses.

We use DNS every day when we type website names like google.com instead of typing complex IP addresses like 142.250.190.78. It's like using a friend's name in our phone instead of remembering their number.

When we enter a website name, DNS servers work behind the scenes to translate it into an IP address. Think of it like asking for directions - your request goes through multiple DNS servers until it finds the right address.

**Task 8:**

**What is TCP and UDP? What is the difference?**

TCP (Transmission Control Protocol): We use TCP when we need reliable communication, like:

When we're downloading files, sending emails, and browsing websites Think of it like sending a registered letter where we get confirmation of delivery.

UDP (User Datagram Protocol): We use UDP when we need speed over accuracy, such as:

When we're streaming videos, playing online games, and making video calls It's like having a regular conversation where we can miss a word but keep talking."

Key Differences:

* Connection: We can see TCP creates a proper connection first, like a handshake, while UDP just starts sending data without any formal connection.
* Speed: We can find UDP is faster because it doesn't wait for confirmations. That's why we use it for live streaming and gaming where speed matters more than perfect delivery.
* Reliability: When we send data through TCP, we're guaranteed it'll arrive correctly and in order. With UDP, we might lose some packets, but we get faster delivery.

Usage: We typically use:

* + TCP for web browsing, email, file transfers
  + UDP for online gaming, video calls, live streaming

So basically, we choose TCP when we need reliability, and UDP when we need speed. It's like choosing between sending a registered letter versus having a quick phone call.

**Task 9:**

**What is TCP and UDP? What is the difference?**

MAC (Media Access Control) address:

It is a unique physical address permanently assigned to a network device by its manufacturer. It's a 12-digit hexadecimal number that never changes. We use MAC addresses for communication within a local network, while IP addresses are used for communication across networks.   
  
The main difference is that MAC addresses are physical and permanent (like a device's serial number), while IP addresses are logical and can change (like a postal address). For example, when we connect to Wi-Fi, our device's MAC address stays the same, but the IP address might change each time. We can think of it this way: MAC addresses identify "who you are" (permanent identity), while IP addresses tell "where you are" (location in the network). This is why we need both - MAC addresses for local network identification and IP addresses for internet routing.

Key differences between MAC Address and IP Address:

1. Nature

* MAC Address: Physical address, permanently burned into network hardware
* IP Address: Logical address, can be changed and assigned dynamically

2. Format

* MAC Address: 12-digit hexadecimal number (XX:XX:XX:XX:XX:XX)
* IP Address: IPv4 uses 32-bit (xxx.xxx.xxx.xxx), IPv6 uses 128-bit format

3. Assignment

* MAC Address: Assigned by device manufacturer, unique worldwide
* IP Address: Assigned by network administrator or DHCP server

4. Purpose

* MAC Address: Used for device identification within local network
* IP Address: Used for routing data across different networks/internet

5. Layer Operation

* MAC Address: Works at Data Link Layer (Layer 2)
* IP Address: Works at Network Layer (Layer 3)

6. Uniqueness

* MAC Address: Globally unique (theoretically)
* IP Address: Unique within network, can be reused in different networks

7. Changeability

* MAC Address: Fixed and cannot be changed (though can be spoofed)
* IP Address: Can be changed easily, dynamic in nature

8. Scope

* MAC Address: Used only within local network
* IP Address: Used for communication across different networks

9. Addressing Type

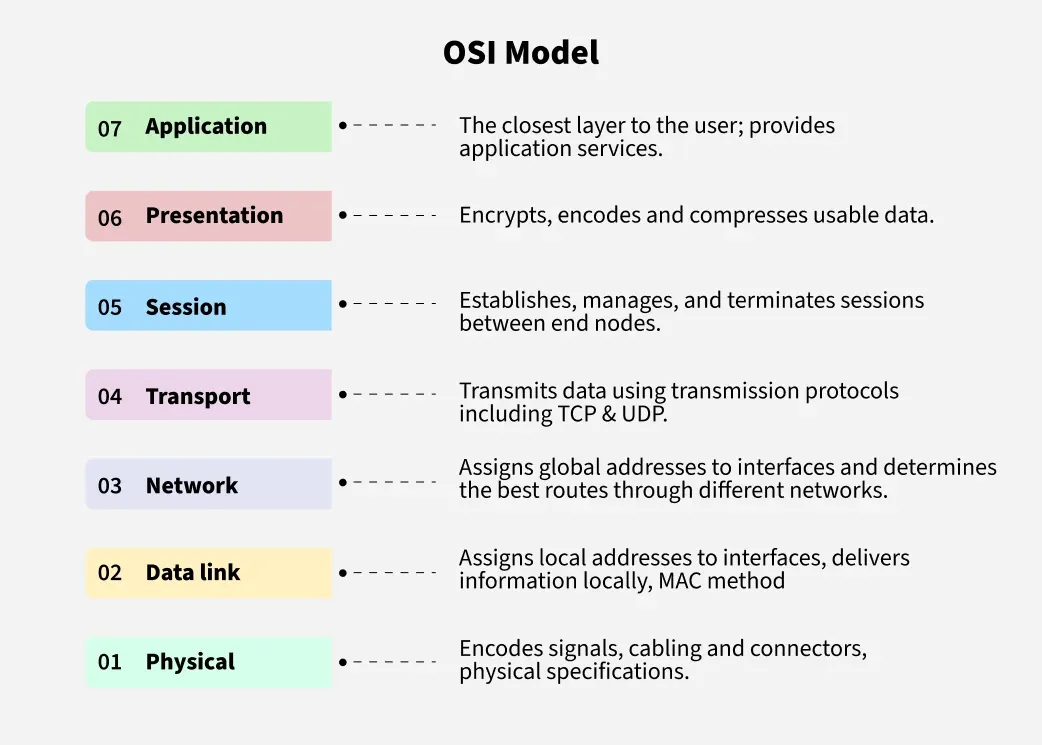
* MAC Address: Flat addressing scheme
* IP Address: Hierarchical addressing scheme

10. Example Usage

* MAC Address: Like a device's serial number
* IP Address: Like a postal address that can change

**Task 10:**

**What is OSI model?**

The **OSI (Open Systems Interconnection)** Model is a set of rules that explains how different computer systems communicate over a network. OSI Model was developed by the **International Organization for Standardization (ISO)**. The OSI Model consists of 7 layers and each layer has specific functions and responsibilities. This layered approach makes it easier for different devices and technologies to work together. OSI Model provides a clear structure for data transmission and managing network issues. The OSI Model is widely used as a reference to understand how network systems function.  


Physical Layer: Transmits raw bits through physical medium like cables and handles electrical/optical signals.

Data Link Layer: Provides node-to-node data delivery and handles physical addressing and error detection.

Network Layer: Manages logical addressing and routing of data packets between different networks.

Transport Layer: Ensures end-to-end delivery, reliability, and flow control between applications.

Session Layer: Establishes, maintains, and terminates connections between applications.

Presentation Layer: Handles data formatting, encryption, and compression between different systems.

Application Layer: Provides network services directly to end-users and applications.

**Task 11:**

**What is an IPv4 address? What are the different classes of IPv4?**

An IPv4 address is a 32-bit numerical address used to identify devices on a network. These addresses are typically represented in a dotted decimal format, like 192.168.1.1, where each number represents a group of 8 bits (an octet). IPv4 addresses are divided into five classes (A, B, C, D, and E) based on their first octet, with each class having a different default subnet mask and intended usage.

Classes of IPv4 Addresses:

* Class A:

0-127.0.0.0 to 127.255.255.255 (Large networks, 8-bit network ID, 24-bit host ID)

* Class B:

128-191.0.0.0 to 128.255.255.255 (Medium-sized networks, 16-bit network ID, 16-bit host ID)

* Class C:

192-223.0.0.0 to 223.255.255.255 (Small networks, 24-bit network ID, 8-bit host ID)

* Class D:

224-239.0.0.0 to 239.255.255.255 (Multicast addressing, used for sending data to multiple hosts simultaneously)

**Task 12:**

**Why we use VPN and advantages of VPN**

A VPN (Virtual Private Network) creates a secure, encrypted connection between your device and the internet, hiding your IP address and protecting your online activities from hackers, ISPs, and other third parties. It allows users to bypass geographical restrictions, access region-locked content, and browse the internet anonymously. VPNs are particularly useful when using public Wi-Fi networks, accessing sensitive information, or working remotely, as they provide an additional layer of security and privacy.  
  
**Task 13:**

**Types of VPN:**

Remote Access VPN: Allows individual users to securely connect to a private network from a remote location using the internet.

Site-to-Site VPN: Connects multiple office locations by creating a secure tunnel between different sites of an organization over the internet.

Intranet VPN: Connects multiple internal sites of the same organization, allowing them to share resources and data securely within the company network.

Extranet VPN: Connects an organization with its partners, suppliers, or customers, allowing controlled access to specific resources while maintaining security.

Task 16:

**Different types of network topology:**

Types of Network Topology:

1. Bus Topology

* All devices connected to a single central cable (backbone)
* Simple, but if backbone fails, entire network fails

**A----B----C----D----E (All devices connected to a single line)**

2. Star Topology

* All devices connected to a central hub/switch
* Easy to manage, but if hub fails, network fails

**B | C---H---D | E (H is hub/switch in center, all devices connected to it)**

3. Ring Topology

* Each device connected to exactly two other devices
* Data travels in one direction in a circle
* If one device fails, network can fail

**A---B---C | | F---E---D (Devices connected in circular manner)**

4. Mesh Topology

* Every device connected to every other device
* Highly reliable but expensive to implement
* Two types: Full mesh and Partial mesh

**A---B |\ /| | X | |/ | C---D (Each device connected to every other device)**

5. Tree Topology

* Hierarchical structure like a tree
* Combination of bus and star topologies
* Good for large networks

**A | B-----C | | D--E F--G (Hierarchical structure)**

6. Hybrid Topology

* Combination of two or more different topologies
* Flexible and scalable
* More complex to manage

**A---HUB1 HUB2---B**

**/|\ /|\**

**/ | \ / | \**

**C D E F G H**

**|**

**|**

**I----J----K----L (Bus Section)**

This represents:

* Two Star topologies (at top), One Bus topology (at bottom), Connected together to form Hybrid

Each topology has its advantages and disadvantages in terms of cost, reliability, scalability, and maintenance.

**Task 18:**

**Difference b/w router and gateway:**

A router is a network device that forwards data packets between different networks and determines the best path for data transmission, while a gateway is a network node that serves as an entry/exit point between two different networks with different protocols. In simple terms, all gateways can work as routers, but not all routers can work as gateways since gateways can translate between different network protocols while routers primarily focus on routing traffic within the same protocol.

Key Differences in a Nutshell:

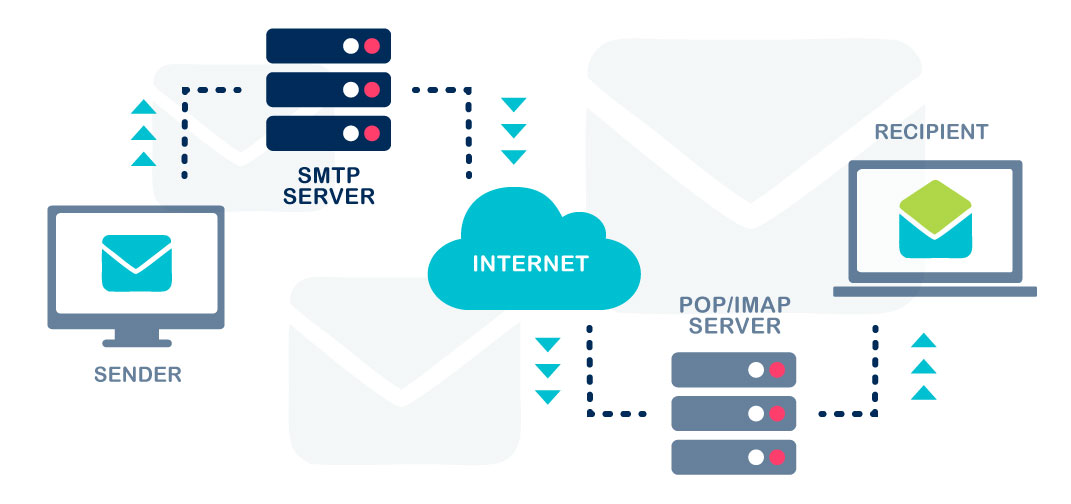
* Network Type: Routers work within networks, while gateways connect networks.
* Protocol Conversion: Routers generally don't need to convert protocols, but gateways often do.
* Security: Both can have security features, but gateways often have more extensive security capabilities.
* Functionality: Routers focus on forwarding data, while gateways focus on connecting and sometimes converting data between different systems.

In summary, a router is a specialized network device for routing data packets within a network, while a gateway is a more versatile device that connects different networks, potentially performing protocol conversion and providing security features.

**Task 19:**

**What is SMTP:**

Behind every email sent, there’s a powerful protocol working tirelessly to ensure your message reaches its destination: SMTP, or Simple Mail Transfer Protocol.  
  
SMTP is a set of rules and guidelines that email servers follow to ensure that emails are delivered from the sender to the recipient accurately and securely.



SMTP protocol, which stands for Simple Mail Transfer Protocol. SMTP is mainly used to send emails over the internet. Whenever we send an email, SMTP is the protocol working in the background to transfer our message from the sender’s email client to the recipient’s email server. It uses TCP/IP to ensure reliable delivery and usually works on port 25. We can think of SMTP like a digital postman—it collects our message, verifies the sender and receiver, and hands it over to the correct mail server. SMTP only handles outgoing emails, so for receiving emails, we usually combine it with protocols like POP3 or IMAP. During the transfer, it follows a step-by-step process with commands like HELO, MAIL FROM, RCPT TO, and DATA. These helps identify both parties and send the actual content. We can say SMTP plays a vital role in email communication, and without it, email systems wouldn't function as smoothly as they do today.

**Task 20:**

**Difference between OSI and TCP/IP:**

Differences between OSI and TCP/IP models:

1. Number of Layers:

* OSI: 7 layers (Physical, Data Link, Network, Transport, Session, Presentation, Application)
* TCP/IP: 4 layers (Network Access, Internet, Transport, Application)

2. Development & Origin:

* OSI: Developed by ISO as a theoretical model
* TCP/IP: Developed by DARPA as a practical model for internet communication

3. Approach:

* OSI: Theoretical and conceptual model that defines networking functions
* TCP/IP: Practical model based on standard protocols in actual use

4. Implementation:

* OSI: More complex and difficult to implement due to its rigid structure
* TCP/IP: Simpler and widely implemented due to its flexibility

5. Protocol Dependency:

* OSI: Protocol-independent (protocols can be replaced as technology changes)
* TCP/IP: Protocol-dependent (protocols are fixed part of the model)

6. Market Adoption:

* OSI: Limited practical implementation, mainly used as reference
* TCP/IP: Widely adopted and is the de facto standard for internet communication

**Task 21:**

**What is High level design and low-level design in SDLC?**

High-Level Design (HLD):

HLD provides a broad system architecture and overview without going into implementation details. It focuses on the "WHAT" aspect of the system, describing major components, modules, and their interactions. It includes system architecture, database design, and interface definitions at a higher abstraction level. HLD is easily understood by stakeholders and serves as a blueprint for the overall system structure. It's like looking at a building's architectural plan from a bird's eye view.

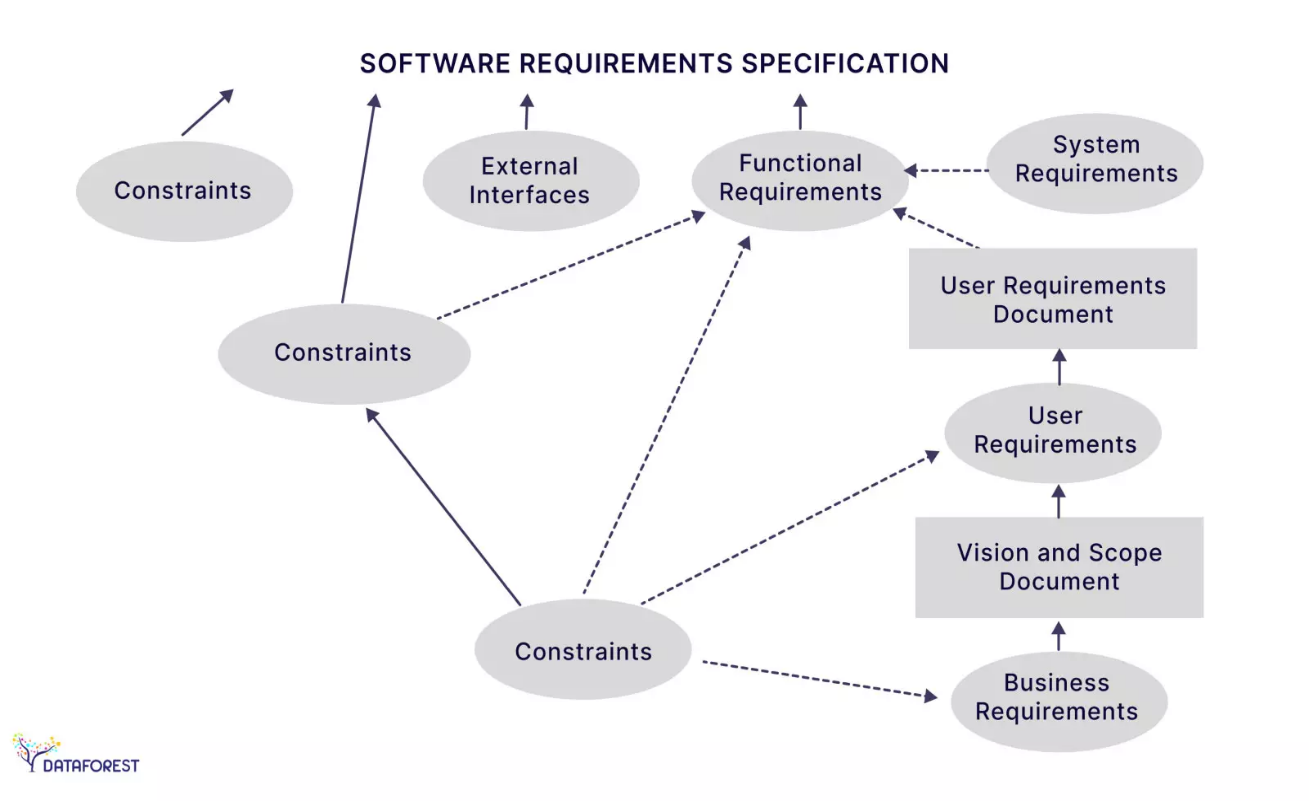
Low-Level Design (LLD):

LLD deals with the detailed internal design and component-level specifications, focusing on the "HOW" aspect of the system. It includes specific class diagrams, database schemas, interface definitions, and algorithm details. LLD is highly technical and used primarily by developers for actual implementation. It's like looking at the detailed construction plan of each room in a building, including electrical wiring, plumbing, and other specific details.

**Task 22:**

**Software Requirement Specification (SRS):**

SRS is a comprehensive document that describes the complete functionality and behavior of a software system to be developed. It serves as a contract between the client and development team, containing detailed requirements including functional requirements (what the system should do), non-functional requirements (performance, security, etc.), and constraints. The document follows IEEE standards and includes system features, external interfaces, and performance requirements. It acts as a foundation for the entire software development process, helping in planning, design, testing, and final delivery of the product.



**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.

**Add ons-**

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

The process of tracking and regulating changes that occur during the software development lifecycle 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.

**8. 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.

**9. 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.

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

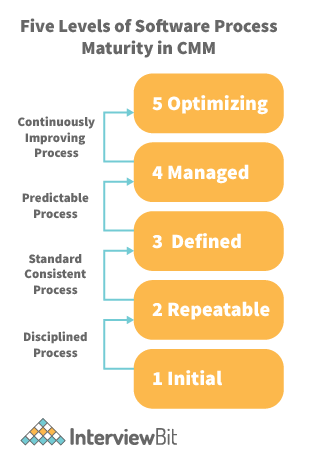
Scrum is an agile development approach based on iterative and incremental procedures that are used in the creation of software. It's an agile structure that's adaptable, rapid, flexible, and excellent at delivering value to customers throughout the project's development. Companies of all sizes employ 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.

**11. 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.

**12. What is 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.

**13. 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.

**14. 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.

**15. What are different types of prototype model?**

There are four types of Prototyping models:

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

**16. What is 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.

**17. 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.

**18. What is the use of 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

**SDLC MCQ**

1.

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

2.

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

Inconsistencies

Redundancies

Omission

**All of the above**

3.

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

**100-200**

300-400

600-700

Above 800+

4.

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

Waterfall model

**V model**

Iterative model

All of the above

5.

What determines if the project should go forward?

**feasibility assessment**

opportunity identification

system evaluation

program specification

6.

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.

7.

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

8.

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**

9.

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

Requirement Gathering

Software Design

System Analysis

**All of the above**

10.

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