Day 1 - 16th May 2025

Your Doc name:

Day 1 - 113312696- Aditya Shubham

SDLC:

Task: 1

What is SDLC?

SDLC is a structured process to design, develop, test, and deploy software. It provides a framework with defined phases to ensure efficiency, cost-effectiveness, and the delivery of a product that meets customer requirements.

5 min 10.10 to 10.15

Task 2:

Why is SDLC ?

1. Ensures Systematic & Organized Development
2. Improves Software Quality
3. Reduces Risks and Costs
4. Enhances Customer Satisfaction
5. Provides Better Project Management
6. Ensures Scalability & Maintainability
7. Compliance & Security
8. Supports Different Development Models

5 min 10.20 to 10.25

Task 3:

What are the stages of SDLC ? write 2 lines about each.

SDLC Phases

1. Planning & Requirement Analysis- Define project scope, objectives, and feasibility
2. System Design- Define architecture and specifications before coding
3. Implementation- Develop the software based on design documents
4. Testing- Identify and fix defects before deployment
5. Deployment- Release he software to users
6. Maintenance & Support- Ensure smooth operation and updates

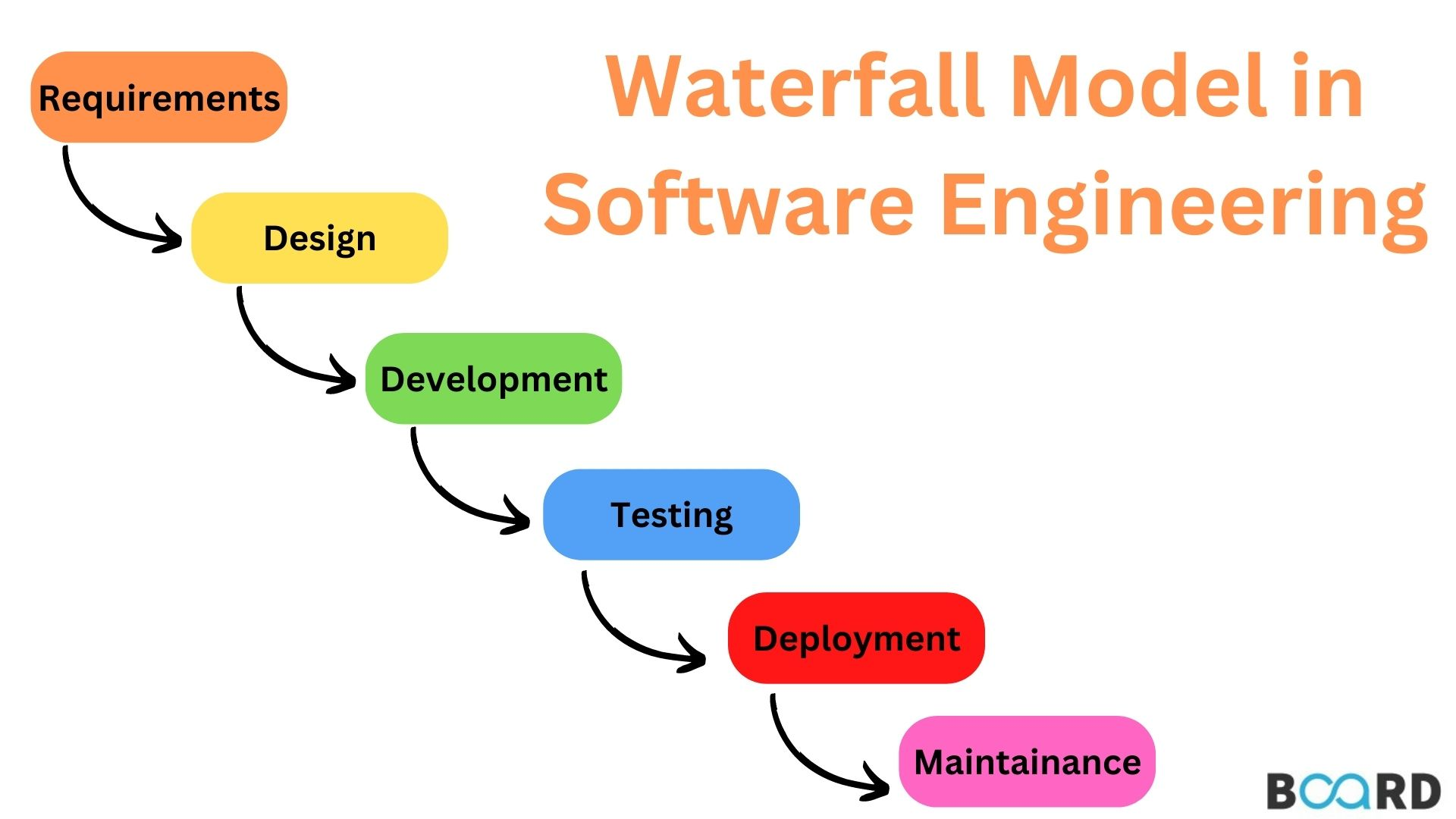
Task 4, 5, 6

SDLC Models:

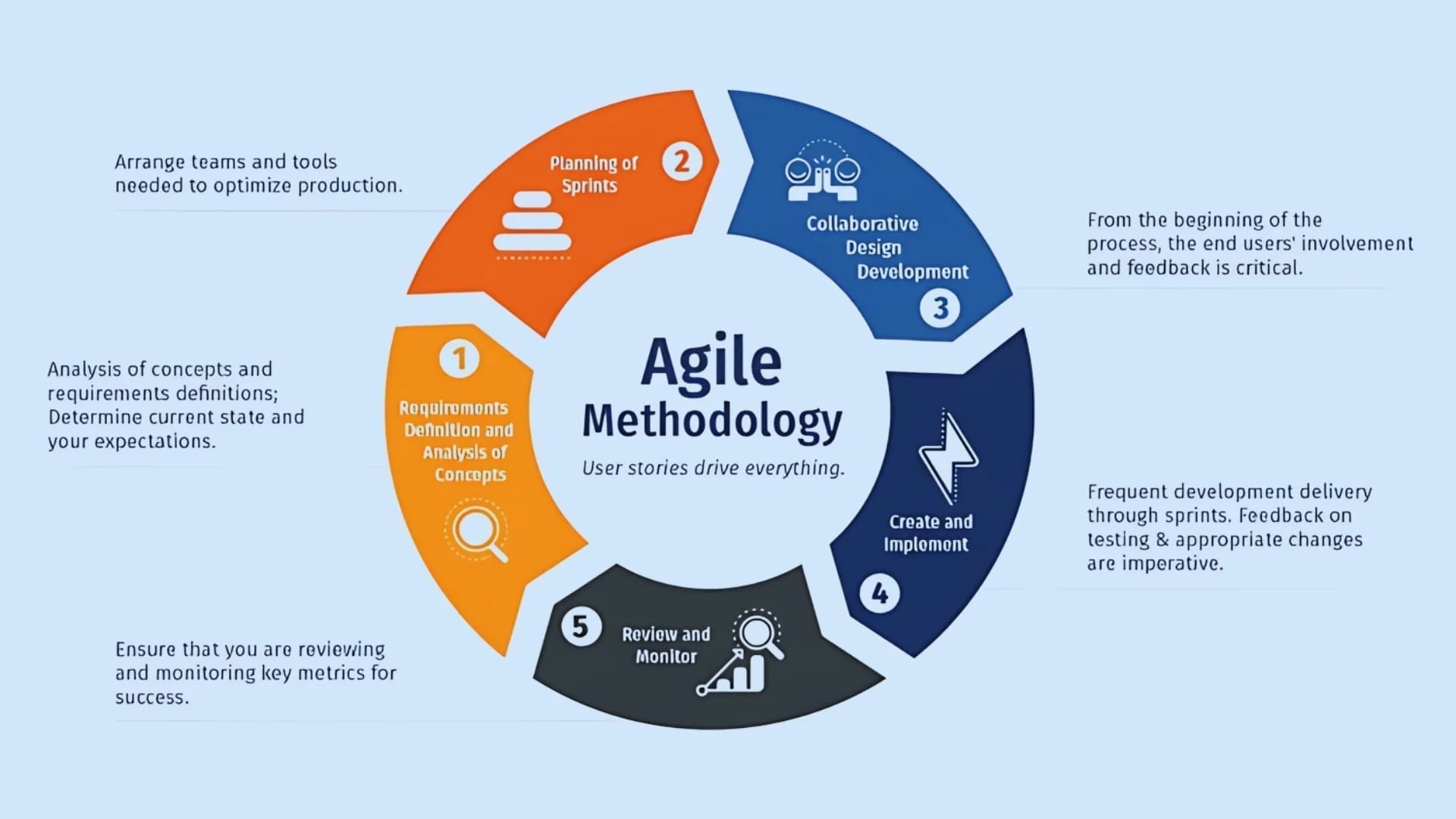
List them , description - 4 lines min and with a image

SDLC Models

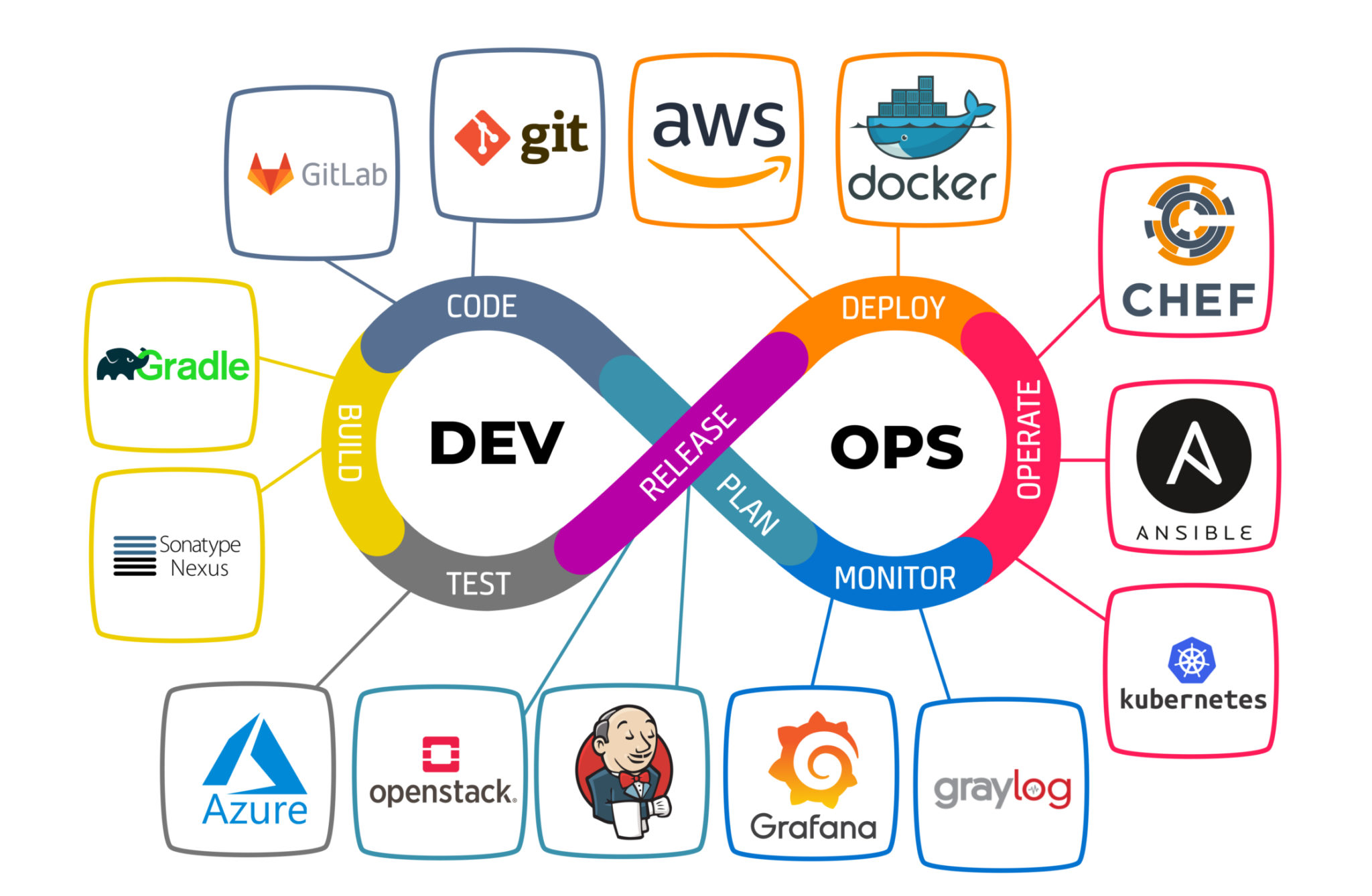
1. Waterfall Model- Sequential linear approach
   1. Each stage must be completed before moving to the next.
   2. Planning & Requirements -> Design -> Implementation -> Testing -> Deployment -> Maintenance
   3. Short, well defined-projects
   4. Simple and easy to manage, good documentation
   5. No flexibility for changes, testing happens late (high risk of failure)
   6. e.g.Building a Banking System (ATM Software, transaction processing)
   7. Requirements are fixed, compliance is critical. E.g. Microsoft early Windows releases



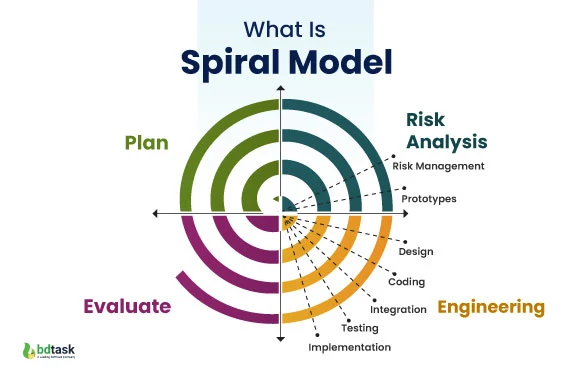
1. Agile Model- Iterative, flexible development
   1. Delivers software in small increments (usually 2-4 weeks sprints)
   2. Scrum (Sprint planning, daily standups, retrospective)
   3. Kanban (Continuous flow, visual task boards)
   4. Dynamic requirements, need for frequent customer feedbacks
   5. Adaptable to change, faster way of working software
   6. Requires high customer involvement, less predictable timeline/budget
   7. E.g. Mobile App Development, frequent updates based on user feedback
   8. Spotify uses Agile

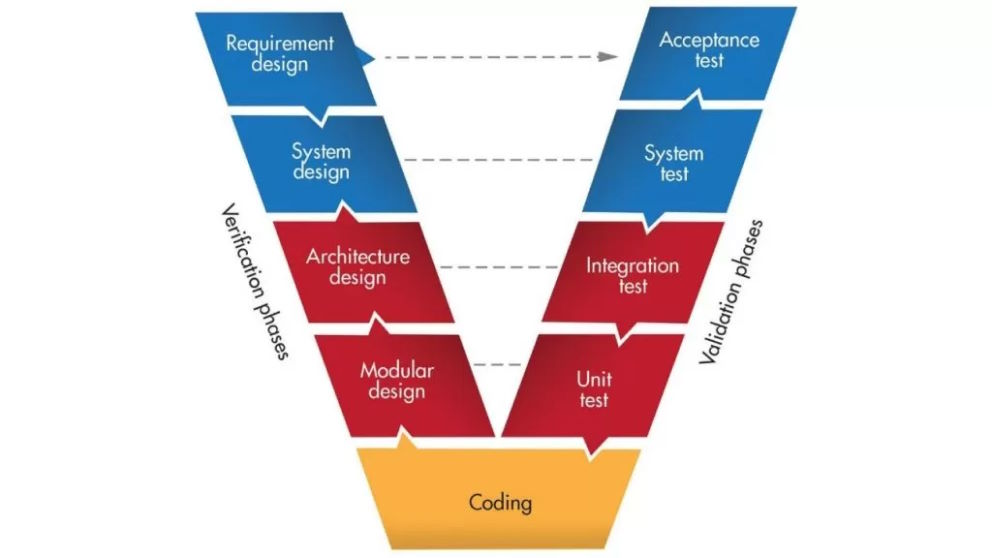


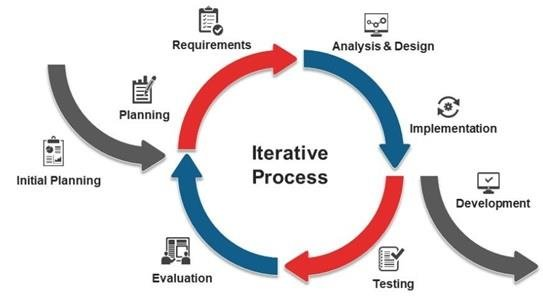
1. DevOps- Combines development and operations for continuous delivery
   1. Continuous Integration (CI) -> Continuous Development (CD) -> Monitoring
   2. Cloud-based scalable application. Frequent updates (e.g. SaaS products)
   3. Faster releases, automations reduces errors
   4. Require cultural shift in teams, high initial setup costs.



1. Spiral Model- Combines waterfall + Iterative Development
   1. Planning -> Risk analysis -> Development -> Evaluation (repeats in cycles)
   2. High-risk, complex projects, long-term projects with evolving requirements
   3. Expensive (requires frequent evaluation)



1. V-Model (Verification & Validation Model): Extension of Waterfall with testing each stage
   1. Each development phase has a corresponding testing phase.
   2. Requirements -> System testing
   3. Design-> Integration testing
   4. Coding -> Unit Testing
   5. Mission-critical systems (e.g. medical, aerospace). High-reliability requirements.
   6. Early defect detection, structured and disciplined.
   7. Rigid (hard to accommodate changes, time-consuming)
2. Iterative Model: Builds Software in Repeated Cycles
   1. Start with a simple version -> Enhance in iterations
   2. Large projects where requirements evolve. Early user feedback is needed
   3. Reduces initial delivery risk, flexible for changes
   4. Can be costly if iterations are uncontrolled



Task 7:

What is a Scrum?

Think of Scrum like a recipe for managing software projects in bite-sized chunks (called "sprints" that typically last 2-4 weeks), where a team works together like a rugby team (which is actually where the term "scrum" comes from) to get things done efficiently. Instead of trying to build everything at once, the team picks the most important tasks from a prioritized list (called a "backlog"), has quick daily check-ins (called "standups"), shows their progress to stakeholders in demo meetings (called "sprint reviews"), and then reflects on what went well or badly (in "retrospectives") - all guided by a Scrum Master who removes obstacles and helps the team stay on track, while the Product Owner makes sure they're building the right things that customers actually want. In essence, Scrum is a practical framework that helps teams deliver working software regularly while being flexible enough to handle changes, much like how a GPS recalculates your route when you take a wrong turn.

Task 8:

What is a sprint?

A short time for which all the product development process is done

Sprint are block of subtasks that are done have got its specific time limit

Task 9: Do's and Don'ts of Sprint (3 points each)

Do's:

1. Follow sprint planning and stick to committed tasks

2. Attend all sprint ceremonies (standups, reviews, retrospectives)

3. Maintain transparency and regular communication

Don'ts:

1. Change sprint scope mid-sprint or add new requirements

2. Skip important meetings or work in isolation

3. Overcommit or take more than capacity

Task 10: Stories and Backlogs in Scrum

User Stories:

- Brief descriptions of functionality from user's perspective

- Follow format: "As a [user], I want [goal] so that [benefit]"

- Should be independent, negotiable, valuable, estimable, small, and testable

Backlogs:

- Prioritized list of all required work

- Contains user stories, bugs, technical tasks

- Dynamic and regularly refined

- Owned by Product Owner

- Constantly evolving based on business needs

Task 11: Scrum Artifacts

1. Product Backlog:

- Complete list of all desired product features

- Prioritized by Product Owner

- Dynamic and continuously updated

- Contains epics, stories, bugs, technical debt

2. Sprint Backlog:

- Subset of Product Backlog items selected for current sprint

- Detailed plan for delivering increment

- Owned by Development Team

- Updated daily during sprint

3. Burn-Down Chart:

- Visual representation of work remaining vs time

- Shows sprint progress

- Updated daily

- Helps identify if team is on track

4. Increment:

- Sum of all completed Product Backlog items

- Must be "Done" as per Definition of Done

- Should be potentially releasable

- Represents value delivered to stakeholders

Task 10: Stories and Backlogs in Scrum

User Stories:

* Brief descriptions of functionality from user's perspective
* Follow format: "As a [user], I want [goal] so that [benefit]"
* Should be independent, negotiable, valuable, estimable, small, and testable

Backlogs:

* Prioritized list of all required work
* Contains user stories, bugs, technical tasks
* Dynamic and regularly refined
* Owned by Product Owner
* Constantly evolving based on business needs

Task 11: Scrum Artifacts

1. Product Backlog:

* Complete list of all desired product features
* Prioritized by Product Owner
* Dynamic and continuously updated
* Contains epics, stories, bugs, technical debt

1. Sprint Backlog:

* Subset of Product Backlog items selected for current sprint
* Detailed plan for delivering increment
* Owned by Development Team
* Updated daily during sprint

1. Burn-Down Chart:

* Visual representation of work remaining vs time
* Shows sprint progress
* Updated daily
* Helps identify if team is on track

1. Increment:

* Sum of all completed Product Backlog items
* Must be "Done" as per Definition of Done
* Should be potentially releasable
* Represents value delivered to stakeholders

Task 12: Ports and Protocols

PORTS:

- Virtual endpoints for network communications

- Numbers ranging from 0 to 65535

- Divided into categories:

\* Well-known ports (0-1023)

\* Registered ports (1024-49151)

\* Dynamic ports (49152-65535)

Common Well-Known Ports:

1. HTTP: Port 80

2. HTTPS: Port 443

3. FTP: Port 21

4. SSH: Port 22

5. SMTP: Port 25

6. DNS: Port 53

7. Telnet: Port 23

PROTOCOLS:

1. TCP (Transmission Control Protocol)

- Connection-oriented

- Reliable data delivery

- Used for: Email, web browsing, file transfer

2. UDP (User Datagram Protocol)

- Connectionless

- Faster but less reliable

- Used for: Streaming, gaming, DNS

3. HTTP/HTTPS

- Web communication

- HTTPS is secure version with encryption

4. FTP (File Transfer Protocol)

- File transfer between client and server

5. SSH (Secure Shell)

- Secure remote access

- Encrypted communications

6. SMTP (Simple Mail Transfer Protocol)

- Email transmission

7. DNS (Domain Name System)

- Domain name resolution to IP addresses

Remember: Port + Protocol combinations enable specific types of network communications and services.

Task 13: Different Network Types

1. PAN (Personal Area Network)

- Smallest network type

- Range: Few meters

- Examples: Bluetooth devices, wireless headphones

- Used for personal device connections

2. LAN (Local Area Network)

- Limited geographical area (building/campus)

- High data transfer speeds

- Common in offices, schools, homes

- Usually connected via Ethernet or Wi-Fi

3. MAN (Metropolitan Area Network)

- Covers a city or large campus

- Larger than LAN, smaller than WAN

- Examples: Cable TV networks, city-wide networks

- Connects multiple LANs

4. WAN (Wide Area Network)

- Spans large geographical areas

- Connects multiple LANs/MANs

- Example: Internet

- Uses telecommunication lines

5. WLAN (Wireless Local Area Network)

- Wireless version of LAN

- Uses Wi-Fi technology

- No physical cable connections

- Common in homes and offices

6. CAN (Campus Area Network)

- University/corporate campus networks

- Connects multiple buildings

- Combination of various network types

7. SAN (Storage Area Network)

- Dedicated network for storage devices

- High-speed data transfer

- Used in data centers

- Specialized for data storage/backup

8. VPN (Virtual Private Network)

- Creates secure connection over public network

- Encrypts data transmission

- Provides remote access

- Ensures privacy and security

Each network type serves specific purposes and has distinct characteristics in terms of:

- Geographic scope

- Speed

- Security

- Purpose

- Infrastructure requirements

Task 14: Types of Servers

1. Web Server

- Hosts websites and web applications

- Handles HTTP/HTTPS requests (Apache, Nginx)

- Delivers web content to browsers

2. Database Server

- Stores and manages databases

- Handles data queries and updates

- Examples: MySQL, PostgreSQL, MongoDB

3. File Server

- Centralizes file storage and sharing

- Manages access permissions

- Common in corporate environments

4. Mail Server

- Handles email sending/receiving

- Manages email accounts and routing

- Examples: Microsoft Exchange, Postfix

5. Application Server

- Hosts applications and business logic

- Connects web and database servers

- Examples: Tomcat, JBoss, WebLogic

6. DNS Server

- Resolves domain names to IP addresses

- Maintains DNS records

- Critical for internet navigation

7. Proxy Server

- Acts as intermediary between users and internet

- Provides caching and filtering

- Enhances security and performance

8. FTP Server

- Manages file transfers

- Provides file upload/download services

- Controls file access permissions

Task 15: DNS (Domain Name System)

1. Purpose:

- Translates domain names to IP addresses

- Makes internet navigation user-friendly

- Works like a phone book for the internet

2. DNS Hierarchy:

- Root Domain (.)

- Top-Level Domains (com, org, net)

- Second-Level Domains (google, amazon)

- Subdomains (mail, www)

3. Key Components:

- DNS Servers

- DNS Records (A, MX, CNAME, etc.)

- DNS Cache

- DNS Resolvers

4. DNS Resolution Process:

- User enters domain name

- Local DNS cache check

- Query to DNS resolver

- Recursive search through DNS hierarchy

- Returns IP address to user

5. Common DNS Record Types:

- A Record (IPv4 address)

- AAAA Record (IPv6 address)

- MX Record (mail servers)

- CNAME (aliases)

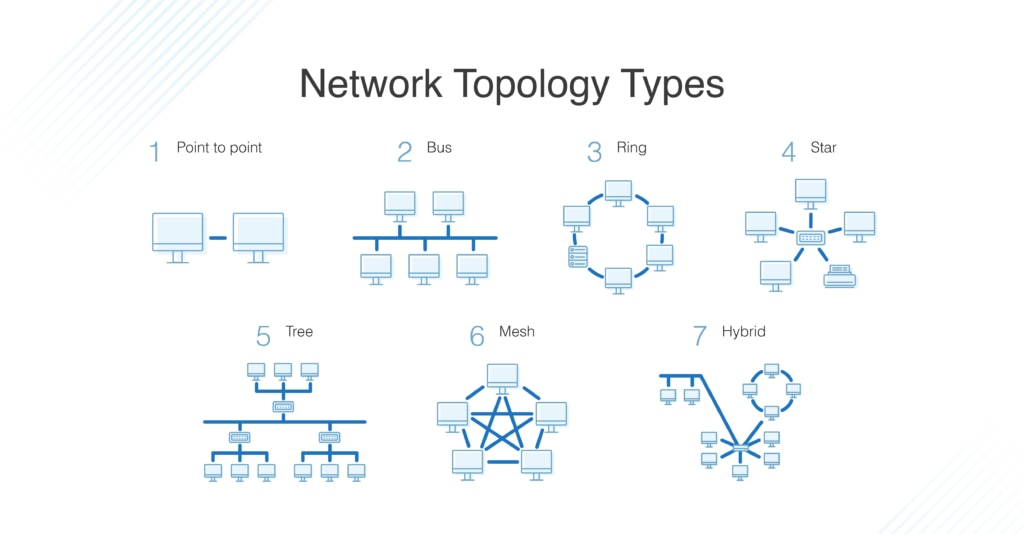
- NS Record (nameservers)

Task 16:

What are the different Network Topologies?

Network Topology is the physical or logical arrangement of network devices and their interconnections in a computer network. It defines how different nodes (computers, servers, switches, etc.) are connected and how they communicate with each other.

Types include Bus, Star, Ring, Mesh, Tree, Hybrid, and Point-to-Point, each having unique characteristics and use cases suitable for different network requirements.



What is OSI Model ?

Describe the 7 layers with description.

OSI Model: A conceptual framework that standardizes the functions of a telecommunication/computing system into seven abstraction layers for network communication.

The 7 Layers (Top to Bottom):

7. Application Layer

- Directly interacts with user applications

- Examples: HTTP, FTP, SMTP, DNS

6. Presentation Layer

- Data formatting and encryption

- Handles data translation and encryption

5. Session Layer

- Manages connections between applications

- Controls authentication and session restoration

4. Transport Layer

- End-to-end delivery and error checking

- TCP/UDP protocols operate here

3. Network Layer

- Routes data packets between networks

- IP addressing and path determination

2. Data Link Layer

- Node-to-node delivery and error detection

- MAC addressing and frame synchronization

1. Physical Layer

- Raw bit transmission over physical medium

- Handles cables, switches, and physical topology