Day 1 - 104608492 - Shirisha Perapagu

SDLC and Networking

**1.What is SDLC?**

SDLC - Software Development Life Cycle

It is a step-by-step process to create or build a high-quality software.

It includes stages or phases from planning and analysis to testing, deployment and maintenance of the software.

**2.Why is SDLC?**

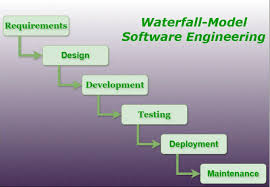
SDLC is important as it helps teams to stay organized and ensure they deliver a high-quality, efficient and cost-effective product that meets user/customer requirements.

**3.Stages of SDLC**

1. **Planning** - understand what the software needs to do and decide if it is worth building. Ex. We need an app to track expenses.
2. **Requirement Analysis** - gather requirements from user and figure out exactly what the software must do. Ex. User should be able to add, edit and view their expenses in the app.
3. **Design** - create a blueprint for how the software will look and work. Ex. sketch the architecture, data structures, algorithms, user interface etc.
4. **Development** - programmers write the actual code based on the design. Ex. Building the app screen by screen.
5. **Testing** - check for bugs or defects and ensure everything works correctly. Ex. Checking the buttons in the app works as it should be.
6. **Deployment** - release the software to the end-users and deploy in the production environment. Ex. App goes on the PlayStore or AppStore.
7. **Maintenance** - the software is monitored for issues, bugs are fixed and updates are made to ensure it continues to function as intended. Ex. getting app updates over time.

**4.SDLC Models**

1. Waterfall Model



A sequential approach where each stage must be completed before the next begins. It is best suited for projects with well-defined requirements and where changes are unlikely.

Applications:

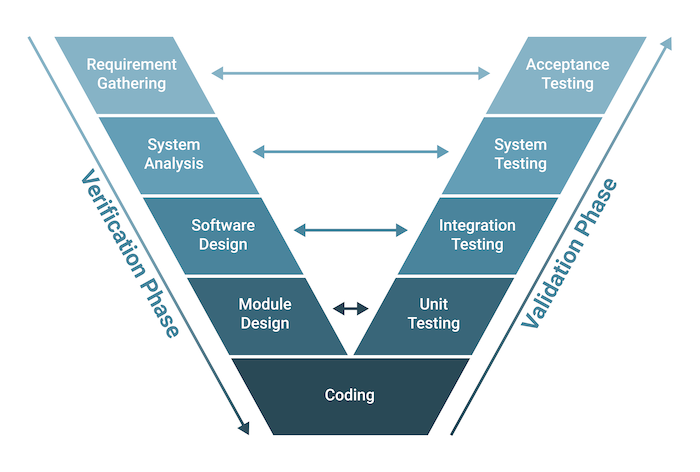
1. Banking systems like ATM or online banking software
2. Payroll systems for calculating employee salaries
3. Airline reservation systems for booking flights

Advantages

1. Clear structure, simple and easy to follow.
2. It is suitable for smaller projects with well-defined requirements.

Disadvantages

1. Once we go to next stage, we can’t move to previous stage and change something.
2. Not suitable for complex projects with sophisticated requirements or frequent changes.
3. V-shaped



V-Shaped model is similar to Waterfall Model, but for every phase on the left (Verification) there is a matching test phase on the right (Validation). It focuses more on testing at every stage.

Applications

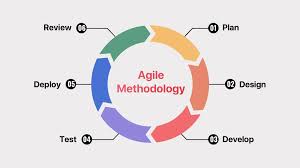
1. Medical Software.
2. Public-sector software projects.
3. Complex systems such as satellite systems

Advantages

1. Testing is done early and often due to which mistakes can be found early.
2. Easy to manage and is good for small to medium projects.

Disadvantages

1. Lack of flexibility as it is hard to change requirements later.
2. Time consuming due to planning and documentation at every stage.
3. Agile



An iterative and incremental approach to build a software in small parts known as sprints or iterations. It allows teams to work closely with users and adapt to changing requirements often.

Applications:

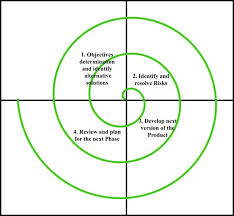
1. Mobile Applications
2. E-commerce sites
3. Social media platforms

Advantages

1. It is very flexible and easy to make changes.
2. Faster delivery of working software allows for quicker feedback and faster time to market.

Disadvantages

1. Difficulty in planning, hard to predict cost and time exactly.
2. Lack of documentation and focus on working software can make it difficult for new team members to onboard and understand project context.
3. Spiral Model



Spiral model is a mix of Waterfall and Agile models. It builds software in repeated cycles known as spirals and focuses on risk analysis i.e. finding and solving problems early.

It is well-suited for projects with complex requirements and high levels of uncertainty.

Applications:

1. Large, high-risk projects
2. Game development projects
3. Military or defense software

**Advantages**

1. **Risk Management as it focuses on early problem solving.**
2. **Suitable for large and complex projects and is flexible**

**Disadvantages**

1. **Expensive and time-consuming projects.**
2. **Need experienced people to handle risks.**

**5.What is Scrum?**

**Scrum is a simple agile way for teams to work together to build software.**

**A small group of employees work together which includes developers, a Scrum Master (a person in charge) and a Product Owner (who knows what needs to be built).**

**Every day the team has a short meeting known as Daily Scrum to check the status, issues and progress of the work. Work is done in short cycles known as sprints. At the end of each sprint, the team reflects on what went well and what to improve.**

**6.What is Sprint?**

A sprint is a short, fixed time duration within which a team works to complete a specific set of tasks which is a small part of project.

Sprint typically lasts between 1 and 4 weeks. Each sprint focuses on what team can complete from to-do list called the product backlog. It is like a focused work session with a clear goal.

**7.What are the Do’s and Don’ts in Sprint?**

Do’s

1. Do plan well. First understand what to build.
2. Do talk daily. Have Daily Scrum meetings to stay in sync.
3. Do finish tasks. Complete all started sprint tasks.
4. Do review and improve at the end of sprint.

Don’ts

1. Don't add new work suddenly. Avoid changing sprint goals mid-way.
2. Don’t work alone. It must be team work, discuss and share problems and solutions.
3. Don’t skip meetings. Meetings would align everyone on same page.
4. Don’t ignore valuable feedback. It helps team to improve and learn and build the right thing.
5. **What are backlogs and what are stories?**
6. User Stories

A user story is a concise, simple description of a specific feature or requirement from the end-user's perspective.­­­

They serve as a way to communicate the requirements to the development team, emphasizing the "what" and "why" rather than the "how".

1. Backlog

A backlog is a prioritized list of all the work that needs to be completed to deliver a product or feature.

It includes user stories, bugs, tasks, and other work items that contribute to the overall product development.

In short

Backlog - to-do list

Stories - the items on that list, describing what needs to be built and why.

1. **What are Scrum artifacts?**

Scrum artifacts are key pieces of information used by the team and stakeholders to track progress, manage work and ensure transparency throughout the development process.

The three main Scrum artifacts are the Product Backlog, Sprint Backlog, and Increment.

Product Backlog

The Product Backlog is a prioritized list of all the work that the team needs to do to deliver the product. It's a living document, constantly being updated with new ideas, features, and bug fixes.

Ex. A list of features to be added to a mobile app, prioritized by user needs.

Sprint Backlog

The Sprint Backlog is a subset of the Product Backlog. It has the items that the development team plans to work on during a specific Sprint.

Ex. A list of tasks to implement a specific feature in a mobile app during a Sprint.

Increment

The Increment is the result of a Sprint, containing all the completed Product Backlog items.

Ex. A version of the mobile app with the completed features from the Sprint.

**10. Ports and Protocols in networking**:

Ports and protocols work together to facilitate communication between devices.

Ports:

Ports are numeric identifiers that lets data go in and out of a device depending on what service or app is being used.

They help ensure that data destined for a specific application or service gets routed to the correct destination within the network.

Example: when we visit a website, our browser uses port 80 (for HTTP) or 443 (for HTTPS) to communicate with the web server.

Protocols

Protocols are sets of rules that define how data is formatted, transmitted, and received across a network.

They ensure that data is exchanged reliably and correctly between devices.

Example:

HTTP (Hypertext Transfer Protocol): Used for web browsing, primarily on port 80.

HTTPS (Hypertext Transfer Protocol Secure): A secure version of HTTP, using port 443.

SMTP (Simple Mail Transfer Protocol): Used for sending email, typically on port 25.

**11. Network Types:**

1. Personal Area Network (PAN) - small network connecting devices owned by a single person over a short distance. Ex. Phone connected to any Bluetooth device or smartwatch.
2. Local Area Network (LAN) - connects devices within a small location such as a home, office, or school. Ex. Wifi network connecting to phone, TV, laptop.
3. Campus Area Network (CAN) - a LAN which connects nearby buildings. Ex. All departments in a university connects in one n/w.
4. Metropolitan Area Network (MAN) - connects devices in a larger geographical area such as a city or metropolitan area, often using fiber optic cables. Ex. A university with several buildings across city connected in one n/w.
5. Wide Area Network (WAN) - connect devices over large are like countries or continents, often using the internet or other public networks. Ex. Companies connected in different countries using internet.
6. Virtual Private Network (VPN) - creates a secure network that hides IP address and encrypts data. It allows users to connect to a private network even when they are in public network. Ex. Connecting our laptop from home to our office network.

**12. What are the types of servers ?**

1. Web Servers

Stores and delivers websites to our browser over internet.

Ex. When we open [www.google.com](http://www.google.com), the browser talks to Google’s Web Server to get the webpage.

1. Database Servers

Stores and manages data in a database, allowing clients to access and manipulate the data.

Ex. To store customer info, orders info etc. from an online store, it uses a database server.

1. Mail Servers

Stores, sends and receives email.

Ex. When we send email using Gmail, it goes through Google’s mail server.

1. File Servers

Stores and shares files in a network.

Ex. Employees access shared files in an office using file server.

1. Application Servers

Runs apps or software for users online.

Ex. A banking app connects to application server to check A/c transactions.

1. Game Servers

Manages multiplayer online games, allowing players to connect and interact with each other.

Ex. PUBG game uses Game server

1. Proxy Servers

Act as mediators between clients (our device) and internet.

Ex. Schools blocking YouTube site with a proxy

1. Virtual Servers

Software based server running on top of a physical server, providing flexibility and scalability.

Ex. Cloud services like AWS, Azure, Google cloud use virtual servers called VMs.

**13. Domain Name System(DNS)**

It acts as a phonebook for the internet, translating human-readable domain names (like www.example.com) into the numerical IP addresses that computers use to communicate.

This allows users to access websites using names they can easily remember rather than having to memorize complex IP address strings.

**14. Network Topologies**

Network topologies define how devices are connected in a network. The main types include bus, star, ring, mesh, tree, and hybrid topologies.

Network Topologies Types:

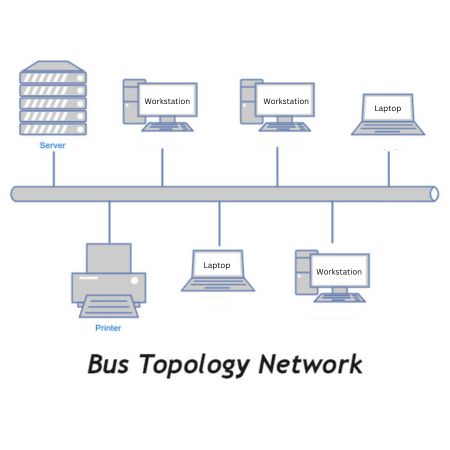
Bus Topology:

All devices share a single long cable(bus) to communicate with each other.

Data travels in both directions along the cable until it reaches the right device.

If the main cable fails, the whole network stops working.

Ex: small offices with few systems.



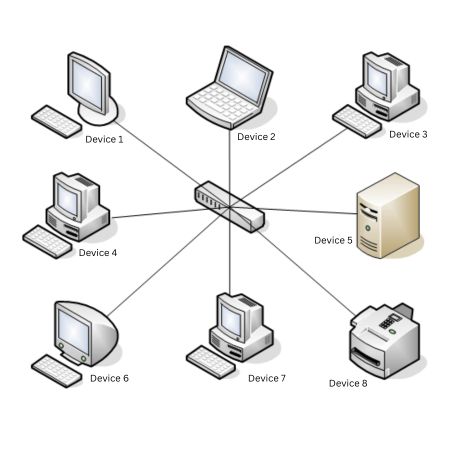
Star Topology:

All devices are connected to a central device like a hub or switch.

Data goes from a device to the central hub, then to the target device.

If the central hub fails, no devices can communicate.

Ex: School computer labs connected to central switch.



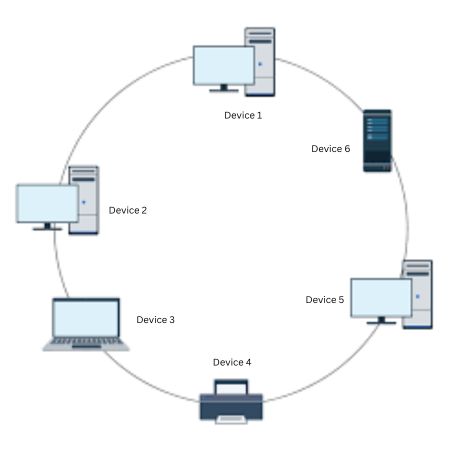
Ring Topology:

Each device is connected to two other devices in a circle/ring shape.

Data circulates around the ring in one direction, passing through each device.

If one device or cable breaks, the whole network goes down.

Ex: Token Ring LANs used in IBM networks.



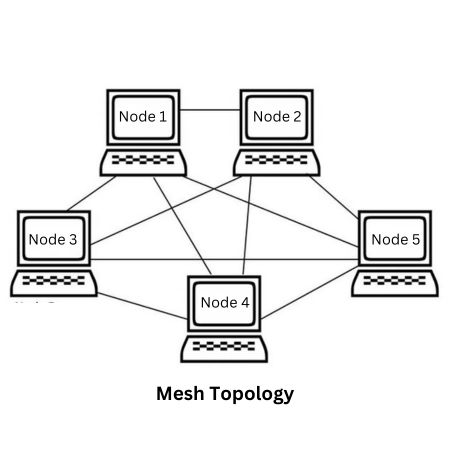
Mesh Topology:

Each device has a direct connection to every other device.

This creates multiple paths for data to travel in the network to reach its destination.

Even if one connection fails, data can still travel in another path. It offers high reliability but is expensive.

Ex. Military communication systems.



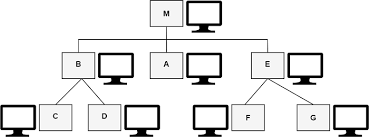
Tree Topology:

It connects multiple star topologies in a hierarchical or layered way just like branches of tree.

Data flows from the top/root node to middle-level nodes and then to lower-level devices.

Easy to manage and expand but if root fails, entire sections go offline.

Ex: University campuses or corporate office networks.

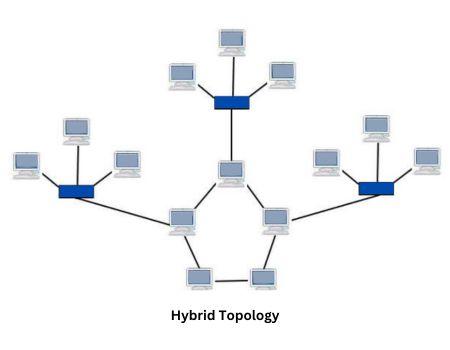


Hybrid Topology:

Combines two or more of the other topologies. Ex. Star + ring, star + bus

Data flows according to the rules of each included topology. It can be flexible.

Ex: A company might use a star topology for office computers and a ring topology for secure servers and connect them together which is hybrid.



**15. Open Systems Interconnection (OSI) model**

The Open Systems Interconnection (OSI) model describes how computer systems communicate over a network.

It divides the process into 7 layers, each with a specific task just like a team working together to send and receive data.

The 7 layers of the OSI Model:­

1. Physical Layer - Sends the raw data as electrical signals or light like wires, cables, wi-fi signals.
2. Data Link Layer - ensures the data is sent without errors between devices on the same network.
3. Network Layer - selects best path to send data to another network like the internet.
4. Transport Layer - breaks down data into small pieces called packets and ensures they arrive correctly.
5. Session Layer - manages connections between two devices from start, usage and until end.
6. Presentation Layer - prepares data to be used by apps like turning it into text, images or encryption.
7. Application Layer - this is where we see and use like apps, websites and emails.