### Day 1- 108415746-Srilekha Bhimavarapu

### What is SDLC?

* Software development life cycle is cost effective and time efficient process that develops the build and design quality.
* There are 5 stages in SDLC which will help us to maintain the quality.
* It defines the entire procedure of software development step by step.
* The risk management is way more potential in this process.

### Why is SDLC?

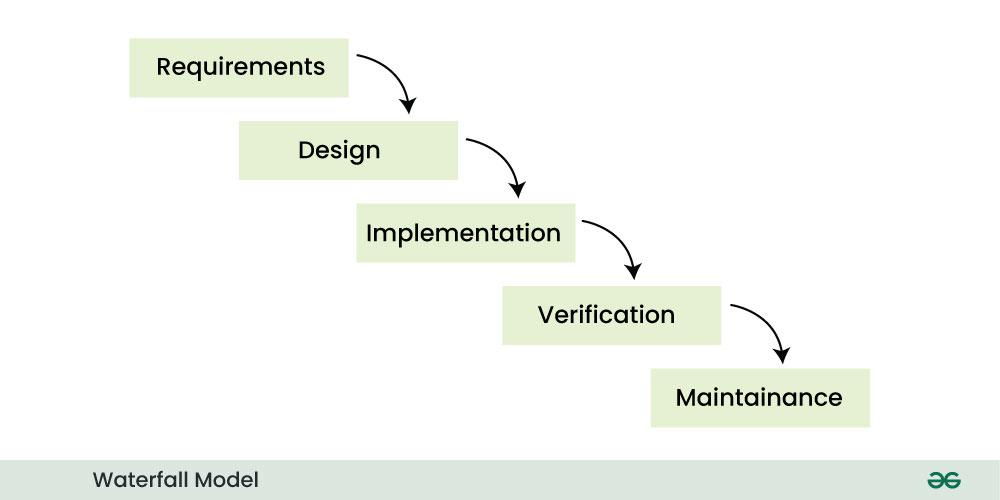
* It helps to maintain project complexities and also improves the quality and ensure the projects are done on time and within the budget.
* It is cost efficient and enhances the risk management.
* It facilitates clear communication between stakeholders and provides documentation for future maintenance.
* Without this there might be a chance of exceeding the timelines and budget also.

### Different stages of SDLC

* There are 6 stages in SDLC.
* **1**. **Planning and Requirement Analysis** – In this stage it defines the project scope and objectives and the resources and also includes the cost benefits analysis.
* **2**. **Defining** – In this stage we will define what the user needs.
* **3**. **Designing** – It includes software architecture and layout of the requirements.
* **4**. **Building** – This stage involves in building the code.
* **5**. **Testing** – It involves in testing the code to get a successful out-come without getting any errors.
* **6**. **Deployment and Maintenance** -- This phase involves releasing the software to the end users in a live environment and the maintenance phase focuses on the improvements of the project after the deployment.
* Each phase contributes to a systematic approach towards the project.

**Models Of SDLC**

**1. Waterfall model**

* 
* a linear, sequential software development process where each must be completed before the next begins.
* We cannot overlap phases in waterfall model
* This model is commonly used for projects with well-defined and stable requirements
* It is not suitable for long and complex projects where flexibility is required and also the testing take places at the end of the complete process.

**2. Agile Model:**



- It is flexible and iterative approach to software development

- Ideal for projects with evolving requirements and need for quick feedback

- The agile model focuses on implementing small changes over a short cycle.

- Agile allows teams to adjust their plans and incorporate feedback at any stage.

- It requires more time and commitment and also less predictable

**3. Spiral Model:**



- Each spiral cycle includes planning, risk analysis, engineering, and evaluation

- Emphasizes risk assessment throughout the development process

- Good for large, complex projects with significant risk factors

- It allows for changes at any stage of the process

- It is complex and requires multiple cycles

**4. V-Model:**



- In this model development process that may be considered an extension of the waterfall model

- In this model verification phase will be on one side and the validation part is on the other side

- Each development stage has an associated testing phase

- Suitable for projects requiring high reliability and extensive testing

- It is highly inflexible, making changes difficult and costly once a phase is completed

**What is scrum?**

It is agile team collaboration framework commonly used in software development and other industries. Scrum prescribes for teams to break work into goals to be completed within time-boxed iterations, called sprints.

**What is sprint?**

Sprint is a fixed duration time box within which a development team works to complete a specific set of tasks. It provides structure way to develop and deliver work in short.

**Do’s and Don’ts of sprint**

**Do’s:**

* Involving the entire team may help
* Setting a clear goal
* Keeping the backlog organized

**Don’ts:**

* Dnt extend the sprint duration
* Dnt add unplanned work to the sprint
* Shouldn’t have communication issues
* Shouldn’t be compromised on quality (like skipping the documentation and compromise security etc)

**Backlogs and stories:**

* In scrum backlogs are prioritized lists of work that need to be done to deliver a product or service.
* Independent units of work that represents a user’s need or goal

**Artifacts:**

* In Scrum, artifacts are key pieces of information that document the work being done and help ensure transparency and focus.

**Scrum artifacts:**

* Product Backlog: It 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.
* Sprint Backlog: It is a subset of the Product Backlog, containing the items that the development team plans to work on during a specific Sprint.
* Increment: It is the result of a Sprint, containing all the completed Product Backlog items that meet the Definition of Done.
* Burndown Chart:A visual representation of the progress made on a Sprint or project, helping to track work remaining and identify potential roadblocks.

**What are ports and protocol:**

PORTS:

* Virtual endpoints for communication
* Identified by numbers
* Enable multiple services on same IP address

PROTOCOL:

* Protocols define how data is formatted, transmitted, and received over a network, ensuring effective communication between devices
* They ensure that data is structured in a way that can be understood by both the sender and receiver, regardless of the underlying hardware or software. Common examples include HTTP, HTTPS, FTP, and SMTP.

**What are the different network types:**

types of networks

* LAN (Local Area Network)
* WAN (Wide Area Network)
* MAN (Metropolitan Area Network)
* WLAN (Wireless Local Area Network)
* VPN (Virtual Private Network)
* VLAN (Virtual Local Area Network)

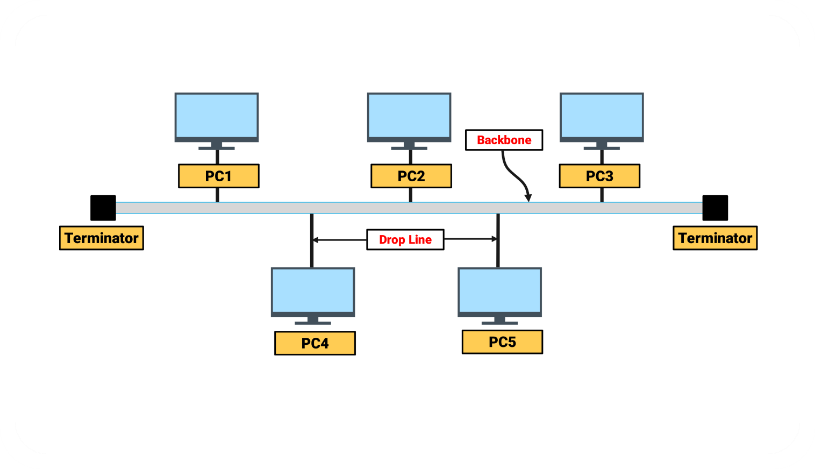
**What are the types of servers**

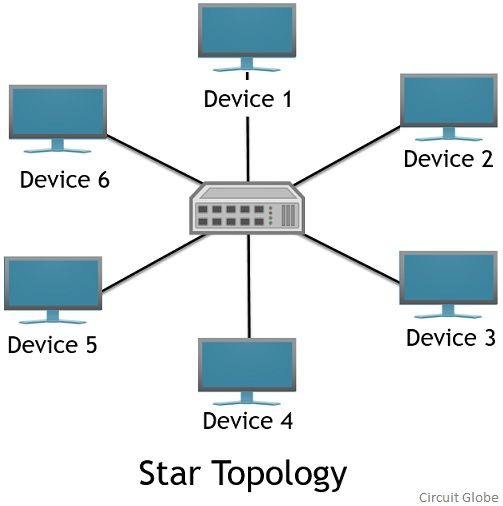
* **Web Servers**: Hosts websites and web applications
* Ex: Apache, Nginx, IIS
* Handles HTTP/HTTPS requests
* **Application Servers**: Runs applications and business logic
* Ex: Tomcat, JBoss, WebLogic
* Manages application resources
* **Database Servers**: Stores and manages data
* Ex: MySQL, PostgreSQL, Oracle
* Handles data queries and storages

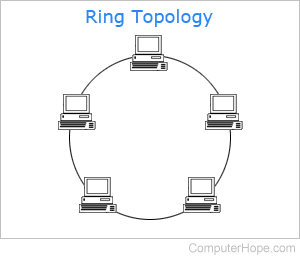
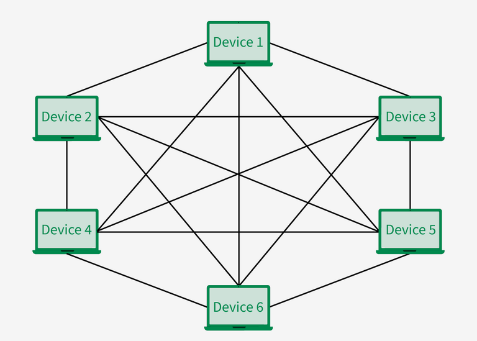
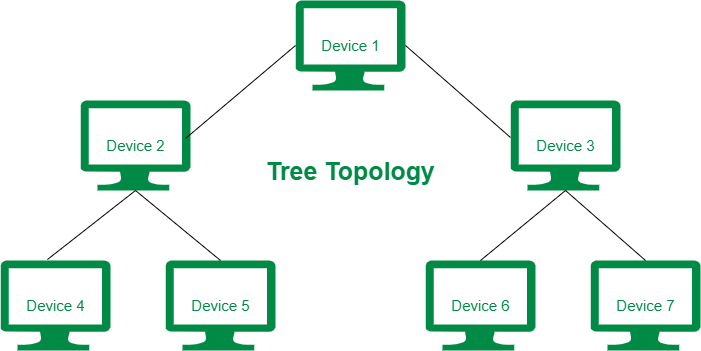
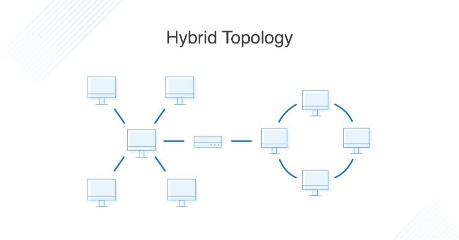
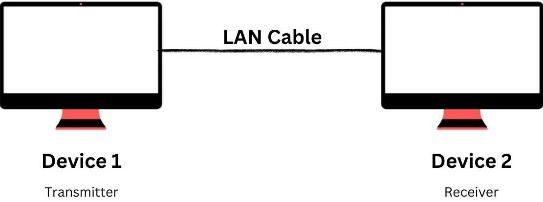
**DNS:**

* A Domain Name System turns domain names into IP addresses, which allow browsers to get to websites and other internet resources.

**Types of topology:**

* Bus Topology: All devices are connect to a single central cable.
* 
* Star Topology: All devices connect to a central hub or switch.



* Ring Topology: Devices connect in a closed loop, with each device linked to adjacent one.
* 
* Mesh Topology: Devices interconnect with multiple direct links to other devices.
* 
* Tree Topology: Hierarchical structure combining elements of bus and star topologies.
* 
* Hybrid Topology: Combination of two or more different topology types.
* 
* Point-to-Point Topology: Direct connection between exactly two network nodes.
* 

**OSI Model:**

Open system interconnection is a conceptual framework that describes how data communication occurs across a network and also divided into 7 layers.

* Physical layer: Deals with the physical transmission of the data across the network
* Data link layer: data transmits between the adjacent nodes and also it detects the error
* Network layer: handles the routing of data packets across different networks
* Transport layer: transmits data using transmission protocols (TCP/UDP)
* Session layer: maintains connection and it is responsible for controlling ports
* Presentation layer: ensures that the data is in a usable format and here the data encryption occurs
* Application layer: provides network services to applications such as http, ftp and email