SDLC

SDLC stands for Software Development Life Cycle. It is Structured process used for Planning, Developing, testing and deploying information systems or software applications.

The objective of SDLC is to implement high Quality Software that meets customer expectations and should be delivered on time.

The phases were: Gathering requirements, planning, designing, Implementation, testing, Deployment and maintenance.

Task-2

-------------------------------

SDLC is a structured approach.

By using SDLC, we can identify the problems in early stage and reduce cost and time to fix them.

Structured process helps to avoid rework and wasted resources.

It ensures reliability and performance. Easier to fix bugs and add features in future with good documentation and structure.

Task-3

Stages of SDLC:

1. Requirement Analysis &planning:

Clients or users gather and analyze what the system should do.

Decide how to build the software. Estimate time, cost and resources.

Identify risks and create a project plan.

2)Design

Define architecture, user interface, database and system flow. Selecting technologies what need to be used.

3)Development

Code the software based on the design.

Develop build modules or components using programming lang and tools,

4)Testing

Check for bugs and error or identifying documenting defects

5)Deployment

Release the software to users, like installing the software,

6)Maintenance

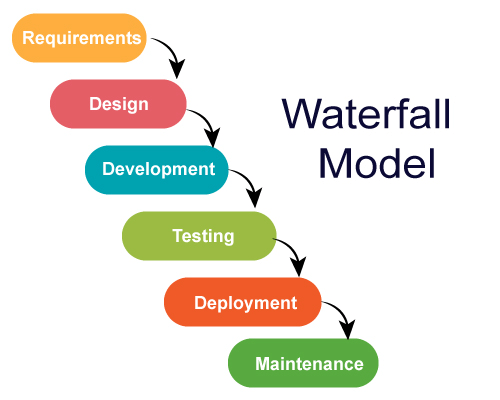
Provide support.

Fix issues, add new features and improve performance

Task : 4

SDLC Models

1. **WaterFall Model**



It uses sequential steps to plan, execute and maintain software.

Because of documentation & planning this is best suited for small applications rather than larger

When we have clear and fixed requirements we use this model.

**Applications**

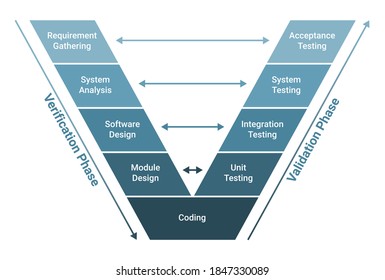
We use this model when we have well-defined, stable and linear.

Ex: Construction building, manufacturing.

**Advantages** : Predictable timelines , Easy documentation and Focus on upfront planning

**DisAdv** : Not ideal for complex projects and Inflexible to change.

**2.Verification&Validation Model**



Verification happens throughout the development phase on the left side of v-shape. It focus on ensuring software built according to specifications and design.

Validation occurs throughout testing phases on the right side of v-shape. It confirms software meets users needs and business goals.

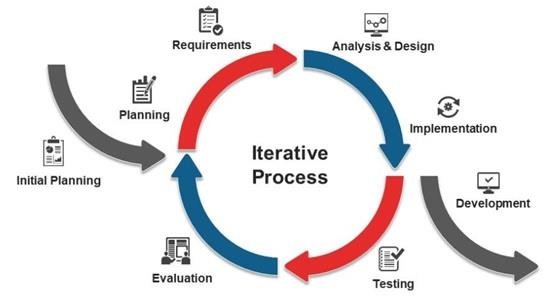
**Applications:**

Medical and Diagnostics software, Space crafts, Chip Designs

**Advantages :** Early defect detection, Improved Quality and Structure approach**.**

**Disadv** : Inflexibility and Limited user input.

**3.Iterative Model**



It will break down the project into smaller components, where each of them is iterated and tested before merging into final product.

It is ideal for flexible applications.

It is can be used for large and complex systems.

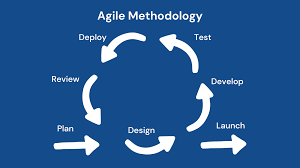
**Advantages** : Reduced risk and Adaptable to change

**Disadv** : Highly skilled resources, Unpredictable timelines.

**Applications**

It is used in Game Development and also in Healthcare

**4. Agile Model**



It uses incremental cycles which are called as sprints.

Software is delivered in small, usable increments.

**Applications :**

It is used in Software development in web& mobile apps.

In Finance like Online banking platforms.

In E-commerce platforms like Amazon, Walmart

**Advantages**: Adapatable to change and Early and continuous feedback

**Disadv** : Requires self organizing teams and Increased management overhead.

**What is Scrum ?**

Scrum is an agile framework that helps teams work together to deliver products step-by-step. Also which is used in Software development.

Scrum breaks work into small chunks called sprints, which are short, focused periods where teams plan, do the work, and review their progress.

It is a flexible way to manage software development projects.

**What is Sprint ?**

Sprint is short and fixed time frame during which a specific set of tasks are meant to be performed.

**Do's and dont's while working in sprints?**

**Do’s**

1. **Define Clear Sprint Goals -** Ensure the team understands the why behind what they’re building.
2. **Prioritize Collaboration -** Encourage open communication among developers, testers, designers, and stakeholders.
3. **Focus on Quality -** Include testing and code reviews within the sprint

**Don’t**

1. **Don't Skip Planning -** Rushing into work without clear goals or definitions leads to confusion and rework.
2. **Don’t Change Sprint Goals Midway -** Changes should be managed through backlog grooming, not during active sprints (unless critical).
3. **Don’t Delay Feedback -** Waiting until the end of the sprint (or later) to review work risks costly changes.

**what do we mean by backlogs and stories ?**

A **backlog** is a **prioritized list of work** for the team that provides visibility into what needs to be done. There are two main types:

1. **Product Backlog**:
   * Owned by the **Product Owner**.
   * Contains all the features, enhancements, bug fixes, technical tasks, and improvements that might be delivered in the product.
   * Continuously refined (this is called **backlog grooming** or **refinement**).
2. **Sprint Backlog**:
   * A subset of the product backlog items that the team commits to during a sprint.
   * Owned by the **development team**.
   * Includes tasks necessary to deliver the sprint goal.

**story** is a feature or requirement from the perspective of the **end user** or customer. It's the most common item in a backlog.

 Purpose: Captures what the user wants to achieve, helping the team understand the value and requirements.

**Scrum Artifacts**

1. **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.An ordered list of everything that is known to be needed in the product.
2. **Sprint Backlog -** The Sprint Backlog is a subset of the Product Backlog, containing the items that the development team plans to work on during a specific Sprint (a fixed time period for completing a defined amount of work)
3. **Increment -** The Increment is the sum of all the Product Backlog items completed during a Sprint and all previous Sprints.

At the end of a Sprint, the new Increment must be “Done,” which means:

* It must meet the Scrum Team’s Definition of “Done.”
* It must be in usable condition regardless of whether the Product Owner decides to actually release it.

**Networking**

**------------------------------**

**what do you understand by ports and protocols in networking?**

A **protocol** is a set of **rules and standards** that governs how data is transmitted and received over a network.

A port is a virtual communication channel, within a computer that's used to identify a specific process or service. Ports act as endpoints for network connections, allowing applications to send and receive data.  A **port** is a **logical endpoint** for communication, used to differentiate between multiple services or applications on a single device.

**Different Networking types**

**1.**Personal Area Network (PAN)

**2.**Local Area Network (LAN)

**3**.Metropolitan Area Network (MAN)

**4**. Wide Area Network (WAN)

**What are the types of servers ?**

A **server** is designed to handle requests from clients and respond with the appropriate data or service. Servers can be physical machines, virtual machines, or even software applications.

**Web Servers:** These servers host websites and deliver web pages to users. They include software like Apache, IIS, or Nginx, according to MilesWeb.

 **Mail Servers:** These servers manage email communication, handling sending, receiving, and storing emails, says Indeed.com.

 **Database Servers:** These servers manage and store large amounts of data, often using database software like MySQL or PostgreSQL.

 **Application Servers:** These servers host and manage web applications, allowing users to interact with them through web browsers.

 **File Servers:** These servers store and share files over a network, enabling users to access and manage them centrally, notes Indeed.com.

 **Print Servers:** These servers manage printing services for computers on a network.

 **Proxy Servers:** These servers act as intermediaries between clients and other servers, often filtering requests and responses.

 **DNS Servers:** These servers translate domain names (like google.com) into IP addresses, [states Cloudflare](https://www.cloudflare.com/learning/dns/dns-server-types/).

 **FTP Servers:** These servers facilitate file transfer between computers over a network.

**DNS**

Domain Name System (DNS) is a hierarchical and distributed naming system that translates human-readable domain names (like google.com) into machine-readable IP addresses (like 142.250.190.14). This translation is crucial for internet communication, allowing users to interact with websites and services using easily memorable names instead of complex numerical addresses.

**Different types network topologies**

A **topology** refers to the **arrangement of devices and connections** in a network—how they’re physically or logically connected and communicate with each other.

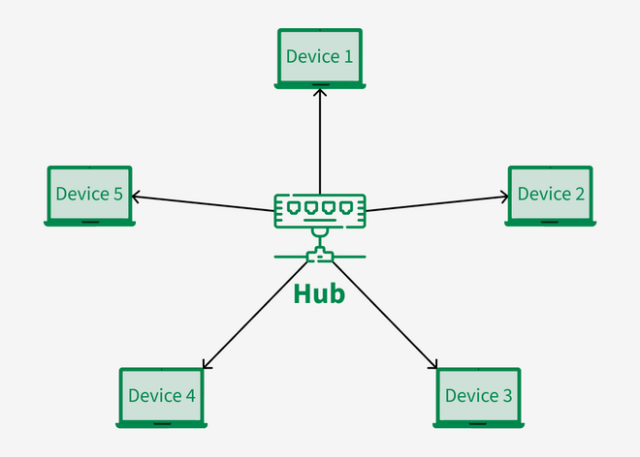
### ****Bus Topology****

* **Structure**: All devices are connected to a single central cable (the bus).
* **Communication**: Data travels in both directions along the cable.



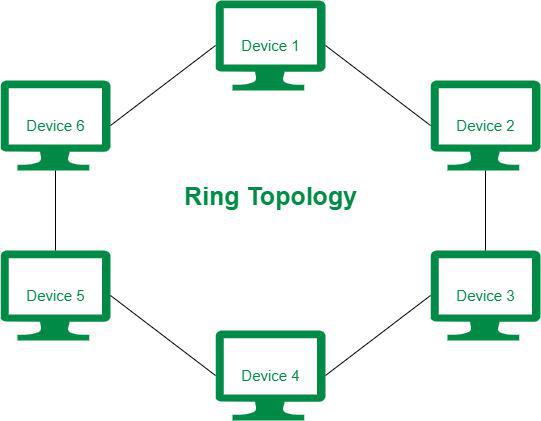
### 2. ****Star Topology****

* **Structure**: All devices are connected to a central hub or switch.
* **Communication**: Data passes through the hub before reaching its destination.

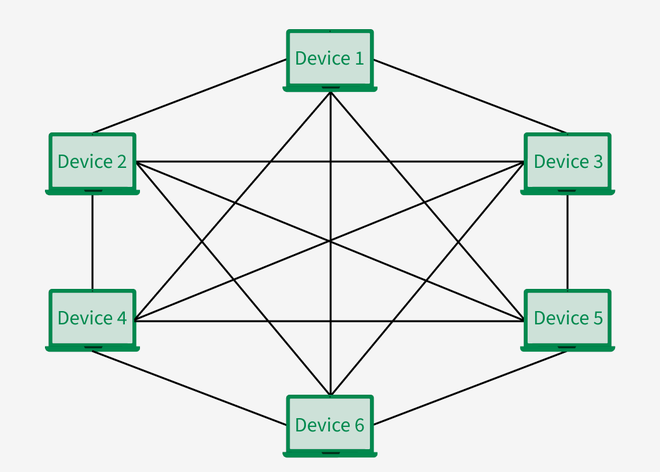


### 3. ****Ring Topology****

* **Structure**: Each device is connected to two others, forming a closed loop.
* **Communication**: Data travels in one direction (or both in a dual-ring).

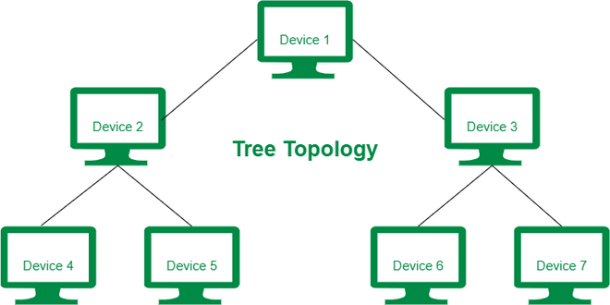


### 4. ****Mesh Topology****

* **Structure**: Every device is connected to every other device.
* **Types**:
  + **Full Mesh**: Every node is connected to every other node.
  + **Partial Mesh**: Only some nodes are fully connected.
* 

### 5. ****Tree Topology (Hierarchical)****

* **Structure**: A combination of star and bus topologies, with groups of star-configured networks connected to a linear bus backbone.



**OSI Model**

It describes how different networking systems communicate. It is used to understand and standardize how data travels from one device to another over a network.

There are 7 layers in OSI model

1.**Physical Layer -** handling the physical transmission of data over a network medium. Transmits raw bits (0s and 1s) over a physical medium.

2. **Data Link Layer -** The data link layer is responsible for the node-to-node delivery of the message. Ensures error-free data transfer between adjacent nodes. This layer provides data transmission over a single network link. It uses protocols like Ethernet and MAC addresses.

The Data Link Layer is divided into two sublayers: LLC and MAC

3. **Network Layer -** The network layer works for the transmission of data from one host to the other located in different networks. It also takes care of packet routing. This layer deals with routing and addressing. It ensures data packets are delivered to the correct destination network.

**4. Transport Layer -** This layer provides reliable end-to-end data transmission. It uses protocols like TCP and UDP to ensure data is delivered correctly.

5. **Session Layer -** This layer manages connections between devices, handling session initiation, maintenance, and termination. It allows applications to establish, manage, and terminate connections.

6. **Presentation Layer -** This layer handles data representation and formatting. It ensures that data transmitted between different systems is understandable. Examples include encryption, decryption, and data compression.

7. **Application Layer -** This is the highest layer, directly interacting with end-user applications. It provides network services like email, file transfer, and web browsing. Examples include HTTP, SMTP, and DNS.