*TASK 1*

**What do you understand about SDLC?**

SDLC is a step-by-step process used to develop software or websites. It includes stages like planning, analysis, design, development, testing, deployment, and maintenance. It helps in building software efficiently, with better quality and fewer errors. SDLC ensures that the final product meets the user’s needs and expectations.

*TASK 2*

**Why do we use SDLC?**

We use SDLC to make sure software is built the right way, step by step. It helps everyone on the team stay clear about what to do and when to do it. This process avoids confusion, saves time, and makes fewer mistakes. It also helps us build something that works well and does what the user needs. In short, SDLC keeps the whole software-making journey smooth and successful.

*TASK 3*

**What are the different stages of SDLC?**

i. **Planning** – We decide what software or webpage to design.

ii. **Analysis** – We find out what the customers/ the users want.

iii. **Design** – Planning on how the software will look or work.

iv. **Development** – Starting to write the code to build the software.

v. **Testing** – Check for bugs and fix errors.

vi. **Deployment** – Release the software to users.

vii. **Maintenance** – Keep fixing and improving or updating the software after it is released.

*TASK 4, 5 & 6*

**What are the Models of SDLC?**

1. Waterfall Model:

This model follows a step-by-step process. Each phase must be completed before moving to the next. You can’t go back to make changes easily. It’s best when requirements are clear from the start.

Used in ATM Softwares, Pay role systems.

Example: Building a railway booking system with fixed requirements.

Advantage: Since each phase (like requirements, design, development) is completed fully before the next begins, it's easy to manage and track progress.

Disadvantage: The rigid phase-by-phase approach makes it hard to go back and fix earlier mistakes during testing.



1. Agile Model:

Work is done in small parts called sprints. You get feedback after each sprint and make changes. It’s flexible and good for evolving projects. Teamwork and quick updates are key.

Used in software startups and mobile app development, where customer needs change often.

Example: Building a food delivery app where new features are added regularly.

Advantage: The use of sprints allows developers to release usable software quickly and respond to user feedback in real time.

Disadvantage: Because planning, coding, and testing happen continuously, it needs constant communication and might lack proper documentation.



1. Iterative Model:

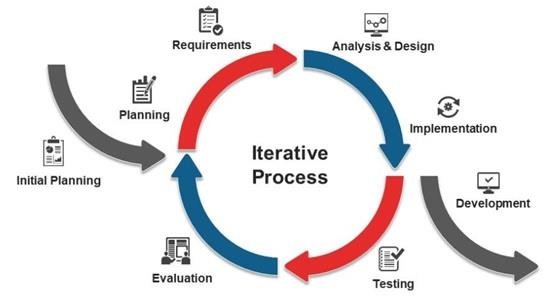
You build a basic version first. Then you improve it in steps through multiple cycles. Each version adds more features or fixes. It’s useful when the full plan isn’t clear upfront.

Best for products that need early versions quickly, and improvements can come later.

Example: Creating a learning platform where basic features go live first, then more tools are added.

Advantage: The early delivery of a basic prototype allows teams to gather feedback before entering the next iteration, improving the final product.

Disadvantage: Frequent changes after each iteration can create confusion and increase integration issues if not well-documented.



1. Spiral Model:

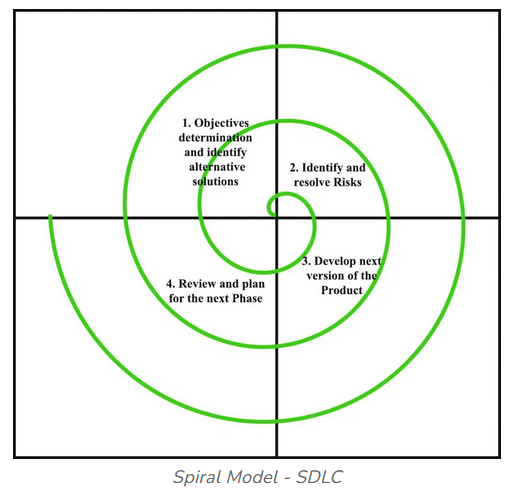
Work is done in loops, like a spiral. Each loop includes planning, building, testing, and reviewing. It focuses on managing risks early. Great for big, risky projects with changing needs.

Used in big-budget, high-risk projects like aerospace or banking software.

Example: Developing software for space missions or defense systems where risk must be handled carefully.

Advantage: Each loop of the spiral includes risk analysis, which helps identify potential problems early—especially in high-risk projects.

Disadvantage: The repeated risk assessment and detailed documentation required in every spiral phase makes it time-consuming and costly.



1. V Model (Validation and Verification):

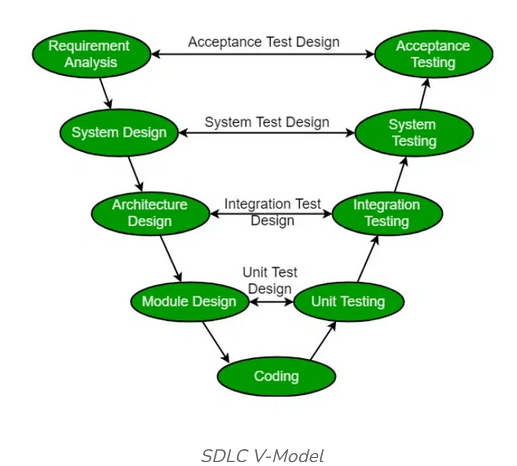
Development and testing go side by side. Each stage has a testing phase linked to it. It ensures high quality and early bug fixing. But like Waterfall, it’s not good for late changes.

Ideal for medical, automotive, or safety-critical systems where testing is crucial.

Example: Software for hospital machines, where any error can risk lives.

Advantage: Every development stage (like requirement analysis or design) is directly linked to a corresponding testing stage, ensuring thorough validation.

Disadvantage: Since testing is tightly coupled with early-stage requirements, any change in requirements later on can affect the entire structure.



*TASK 7*

What do you understand from Scrum?

Scrum is a way of working in teams to build products step by step. It breaks the work into short time frames called sprints, where the teams plan, works, and reviews what’s done. Everyone in the team has a role – like the Product Owner who decides what to build, and the Scrum Master who supports the team and removes roadblocks. Scrum helps teams stay focused, work faster and improve based on feedback after each sprint. It’s very useful when project needs keep changing.

*TASK 8*

What is SPRINT?

Sprints are short, fixed time period in Scrum – usually 2 to 4 weeks – where the teams works on a set of tasks to deliver a small, usable part of the products.

At the start of the sprint, the team selects what they can complete from the backlog. During the sprint, they review what’s done, take feedback, and plan the next sprint.

It’s like a mini project inside the bigger project – helping teams stay focused, deliver faster, and improve regularly.

*TASK 9*

Do’s and Don’ts of Sprints:

Do’s:

1. Break tasks into smaller, manageable pieces for easier tracking
2. Review progress frequently to catch issues early.
3. Celebrate small wins to keep the team motivated, be open to feedback and improve each sprint.

Don’ts:

1. Don’t procrastinate on difficult tasks—tackle them early and don’t add new tasks in the middle of a sprint without agreement.
2. Don’t ignore problems or blockers – raise them early.
3. Don’t resist change—be flexible to adapt if priorities shift.

*TASK 10*

What are stories and backlogs?

User Stories are short simple descriptions of a feature or task written from the user’s point of view.

Each story represents a small piece of work the team can complete during a sprint.

A backlog is a lost of all the tasks featured, bug fixes, and improvements needed for the project.

There are 2 main types:

Product Backlogs: Maintained by the Product owner, it holds all ideas and requirements for the product.

Sprint Backlog: A small list created from the product Backlog with only theitems selected for the current sprint.

*TASK 11*

In Scrum, artifacts are the important tools or documents that help the team track progress, stay aligned, and deliver value. They bring transparency to the process so everyone knows what’s happening.

There are 3 main Scrum Artifacts:

1. Product Backlog – This is the master list of all the features, changes, bug fixes, and ideas for the product. It’s created and maintained by the Product Owner. Items in the backlog are called user stories, and they are continuously refined, added, or removed based on priorities. It answers the question: "What needs to be built?"

2. Sprint Backlog – This is a smaller list taken from the Product Backlog. It contains the tasks the team commits to completing in the current sprint. The Development Team selects these items during Sprint Planning. It is a dynamic list that the team updates daily as work progresses. It answers the question: "What are we working on right now?”

3. Increment – This is the final output or working product delivered at the end of a sprint. It includes all completed Product Backlog items from that sprint. The increment must be “Done”—meaning it is tested, functional, and potentially ready to release. It answers: "What did we finish and can deliver?"

Burndown Chart is not officially one of the three main Scrum Artifacts, but it is still a very important Scrum tool.

A Burndown Chart is a visual tool used in Scrum to show how much work is left in a sprint and how fast the team is completing it.

It helps track progress daily and ensures the team is on schedule to meet their sprint goal.

*TASK 12*

What are Ports and Protocols?

Think of your computer like a big building. Now, every time data comes in—like opening a website or sending an email—it needs to go to the right room. That’s where ports come in.

Ports are like room numbers in your computer that guide the data to the right place.

For example:Port 80 is the room for normal websites, Port 443 is for secure websites, Port 25 is for sending emails.

Now, for the data to travel and understand how to behave, it needs rules—those are called protocols.

Protocols are like the language or set of instructions the data follows. So HTTP is the rulebook for websites, FTP is for file transfers, and TCP/UDP tells how data should travel—whether it needs to be safe or fast.

Ports are like doors where data enters, and protocols are the rules it follows to talk properly once it’s inside.

*Task 13*

What are the different Network types?

1. PAN (Personal Area Network)

This is the smallest type of network, used around a person.

Example: Your phone connected to your smartwatch or wireless headphones via Bluetooth.

2. LAN (Local Area Network)

This network connects computers within a small area, like a home, office, or school.

Example: All computers in a school lab sharing files and printers.

3. MAN (Metropolitan Area Network)

Covers a larger area than LAN—like a city or a big campus.

Example: A city-wide Wi-Fi system or university networks spread across campuses.

4. WAN (Wide Area Network)

This connects computers over large distances—even across countries.

Example: The internet is the biggest WAN—it connects the whole world.

5. WLAN (Wireless LAN)

Same as LAN, but without wires—it uses Wi-Fi instead.

Example: A wireless router connecting your laptop and phone at home.

*TASK 14*

What are the types of servers?

A server is a powerful computer that provides services, data, or resources to other computers (called clients) over a network.

Now here are the main types of servers:

1. **Web Server**

It stores websites and sends them to your browser when you type a URL.

Example: When you visit Google, a web server sends the page to you.

2. **File Server**

Stores and manages files, allowing users to upload, download, and share files on a network.

Used in offices to share documents between employees.

3. **Database Server**

Stores and organizes large amounts of data that other apps can request and use.

Example: When you log into Amazon, the server fetches your order history from a database.

4. **Mail Server**

Handles sending and receiving emails.

Example: Gmail uses mail servers to deliver and store your emails.

5. **Application Server**

Runs specific software applications and serves data to users or other servers.

Example: Banking apps connect to an application server to show your balance.

6. **DNS Server (Domain Name System)**

Translates website names (like google.com) into IP addresses that computers understand.

7. **Proxy Server**

Acts like a middleman between your device and the internet—used for security, filtering, or faster access.

*Task 15*

What do you know about DNS? Domain Name System

The Domain Name System (DNS) is a system that translates human-friendly website names (like www.google.com) into computer-friendly IP addresses (like 142.250.182.14).

Every website on the internet has an IP address, but remembering these numbers is hard. DNS makes it easier by letting us use names instead.

For example, when you type a website name into your browser, DNS quickly finds the right IP address behind the scenes and connects you to the correct website.

Without DNS, we would have to remember the numeric IP address of every website we visit.

------

VPN stands for Virtual Private Network.

It’s a technology that helps us browse the internet securely by creating a private tunnel between our device and the internet.

This means our data is encrypted, and no one—not even hackers or internet providers—can see what we’re doing online.

It also hides our real IP address, so we can access blocked websites or content safely, even on public Wi-Fi.

Now, there are a few types of VPNs we should know:

1. Remote Access VPN – This is used by people who want to securely access their company’s network from anywhere, like work-from-home employees.

2. Site-to-Site VPN – This connects entire networks together. For example, a company’s branch office in another city can securely connect to the head office network.

3. Client-based VPN – This is when users install a VPN app on their device to connect securely. It’s what we mostly use for personal VPNs.

4. Network-based VPN – This is handled by organizations at the network level, protecting communications without needing installation on each device.

So in short, a VPN helps keep your internet activity safe, secure, and private, no matter where you are.

------

*TASK 16*

What are the different Network Topologies? Name the types.

Network topology is just a fancy way of saying how devices like computers and printers are connected in a network.

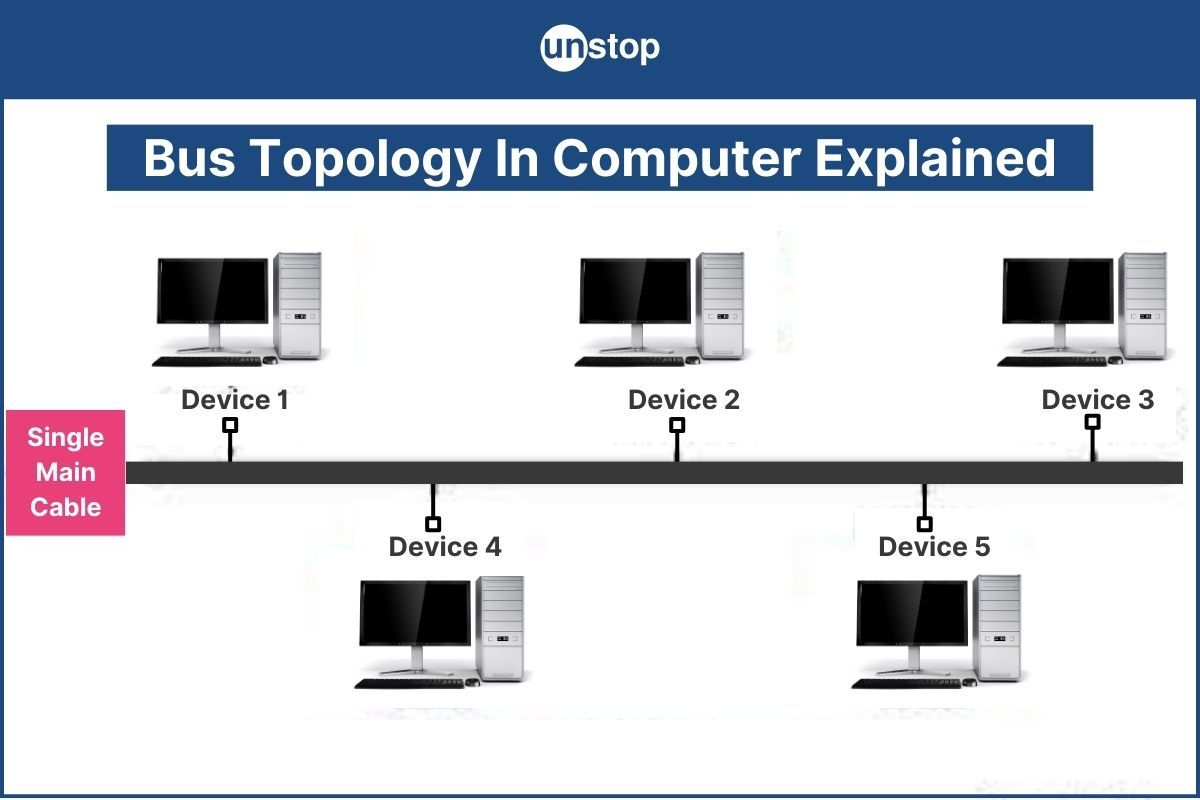
It’s the layout or structure of how everything talks to each other.

Each layout has its own strengths and weaknesses, depending on how the network is used.

**Types of Network Topologies**:

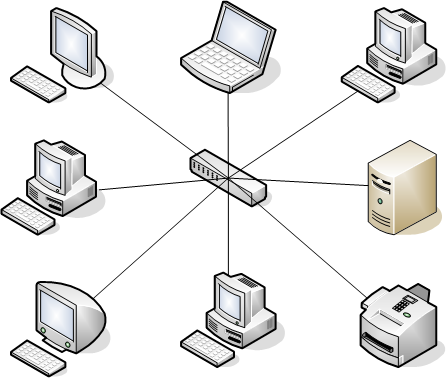
1. Bus Topology:

All devices are connected to a single cable. Think of it like a one-way street where only one car (data) can pass at a time. It’s simple and cheap, but if the cable breaks, everything stops working.



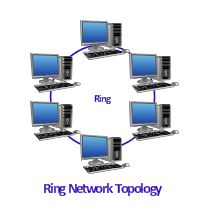
2. Star Topology:

Every device is connected to a central hub. If one device fails, the others are fine—but if the hub goes down, the whole network crashes. This is common and easy to manage.



3. Ring Topology:

Devices are connected in a circle, like a ring. Data moves around the ring until it reaches the right device. It works well, but if one link breaks, it can disrupt the whole loop.



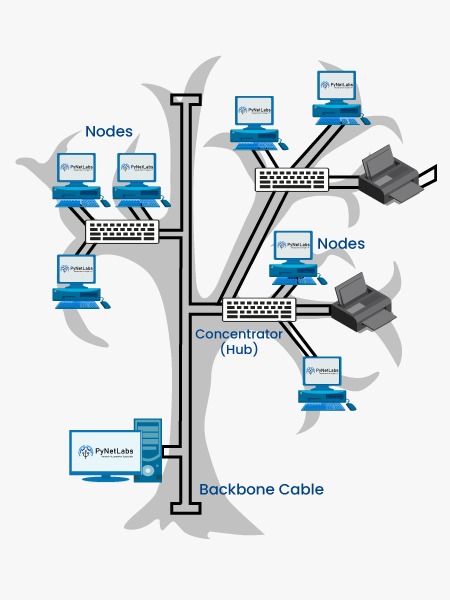
4. Mesh Topology:

Every device is connected to every other device. It’s super reliable because data has many paths to travel. But it’s expensive and complex to set up, used in high-security places.



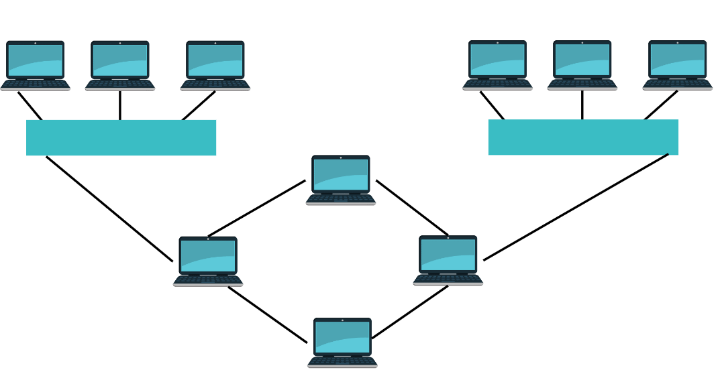
5. Tree Topology:

This looks like a tree, with a main line and branches. It’s good for large networks with groups of devices. If the main trunk fails, the whole branch can be affected.



6. Hybrid Topology:

A mix of two or more topologies based on what the network needs. It gives flexibility and can be customized easily. Used in big companies with many departments and network styles.



*TASK 17*

What is OSI Model ? Describe the 7 layers with description.

The OSI Model (Open Systems Interconnection) is like a step-by-step guide that explains how data moves from one computer to another over a network. It has 7 layers, each doing a specific job in the journey of data.

1. Physical Layer – This is the actual hardware: cables, switches, signals—where bits travel physically.

2. Data Link Layer – Makes sure data is sent to the right device on a local network using MAC addresses.

3. Network Layer – Handles the delivery of data between networks using IP addresses (like a GPS).

4. Transport Layer – Ensures data is delivered correctly and in order, using TCP or UDP.

5. Session Layer – Manages the start, control, and end of conversations (sessions) between devices.

6. Presentation Layer – Translates data into a format both sender and receiver understand (like encryption or file types).

7. Application Layer – This is where the user interacts—think of browsers, email apps, etc.