**SDLC:**

Task 1:

**What is SDLC?**

SDLC stands for Software Development Life Cycle.

SDLC ensures a software is developed with continuous testing and making sure the end product is created as per the expectations of the end customer.

SDLC has various stages:

First we check the scope of the software being created.

Then we create a design for the software about how it’s going to work.

We test the code and then deploy and maintain.

There are various models in SDLC such as

Waterfall model, Iterative Model, Agile Model.

Task 2:

**Why is SDLC**

SDLC is important to identify defects and bugs in the software before it is handed over to the end customers.

It ensures software works as expected and performs well under different conditions.

SDLC helps in identifying defects in the early stage of development, this ensures minimal costs in fixing defects.

It overall improves the customer satisfaction because software goes through rigorous stages of testing and feedback.

Task 3:

**What are the stages of SDLC**

1. Scope

Before jumping on to creating software, first we need to identify the need of the software. What problem is it going to solve and what service it is going to provide and what are the resources software will require.

1. Design

After figuring out the scope of the software, developers work on creating the blueprint of software about how it is going to work. What features it will have.

1. Code

This part of the process is coding and actually working on the software creation using schema or blueprint of the software.

1. Test

Code written by developers has to be tested using various methods to ensure it is bugs free and works as intended.

1. Deploy

After making modifications, software is deployed so that end users can start working on it.

1. Documentation

Once software is complete, developers have to create documents on how the software works. It describes the functionality of the software.

1. Maintenance

A software once complete goes through continuous improvement and continuous development stage where developers can make changes to the software to add new features to it or enhance it.

Task 4:

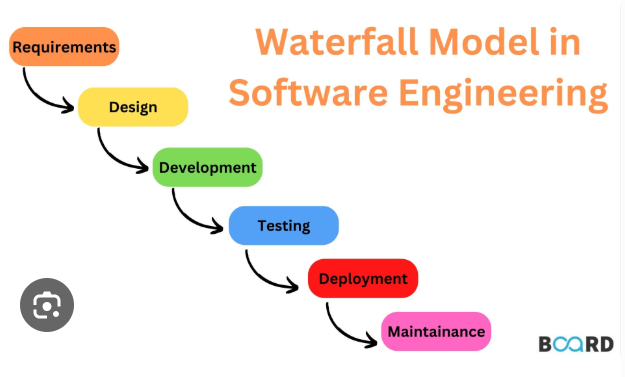
**SDLC Models**

There are 5 such SDLC models which ensure the product is developed as per user expectations and goes through testing under different stages of development of a software.

**Waterfall Model:**

This model uses a step by step approach.This model is suitable for small applications where requirements are clear and fixed. This model is easy to understand and follow.

It focuses on upfront planning. Frequent changes can disrupt the testing stage.



**Application**:

NASA has used the Waterfall Model for some of its space mission software (like space shuttle control systems) because requirements are fixed, and there’s no room for error or last-minute changes.

**Advantages**:

This model has a clear and defined process. It is suitable for small projects so timelines are also predictable.

**Disadvantages**:

Since testing happens at a later stage of the development, changes in the application can be costly and time consuming.

Once a phase is complete, we cannot go back to previous phases.

**Verification and Validation Model:**

Under this model, testing is done in stages to identify and fix bugs before it becomes a bigger issue. Each development stage has testing on the right side.

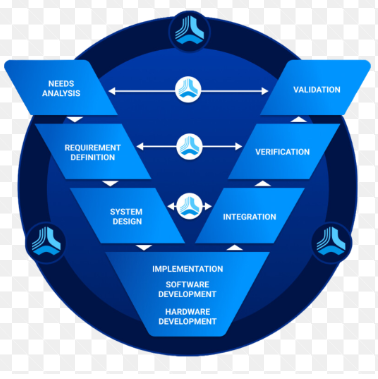
It is also known as V methodology. It is an extension of the waterfall model.

First verification is done on the basis of requirements and system design.

On the basis of this, coding is done to develop the application.

Once the coding is complete, under validation phase testing is done.

Each development stage has a testing stage on the right side.



**Application:**

In aviation, software controlling things like autopilot or navigation must be 100% safe and reliable. So, companies like Boeing use the V&V model to first verify that each part of the software is built correctly (checking against requirements), and then validate that the entire system works as intended in real flight conditions.

**Advantages:**

Clear and straightforward with defined phases

Easy to manage and test at each stage.

**Disadvantages:**

Simple projects can be very complex due to feedback at every stage.

Limited flexibility for changes late in development.

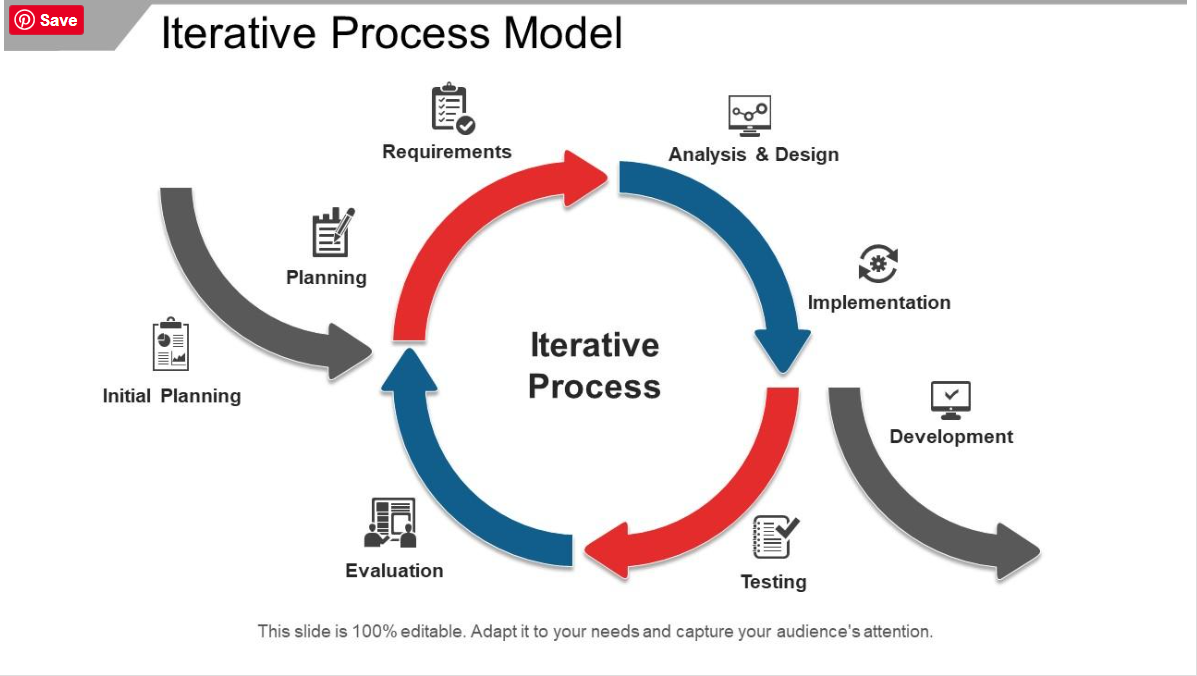
Not suitable for complex projects.

**Iterative Model:**

The cycles for each component mimic a full development cycle, from planning and design to testing and deployment.

It is a data driven model and test results are used for the next iteration created. It is used for flexible applications where scalability is an important factor.

It ensures the final product aligns with the user's needs and expectations.



**Application:**

It is used where applications require tailoring that evolve with specific business needs.

Used by Social Media websites like Facebook, instagram.

**Advantages:**

Adapt to change

Early user feedback ensures Improved quality

Focuses on improving the product with each version

Usually cheaper and simpler

**Disadvantages:**

Unpredictable timelines can be a challenge

Increased management cost

Need for high skilled resources

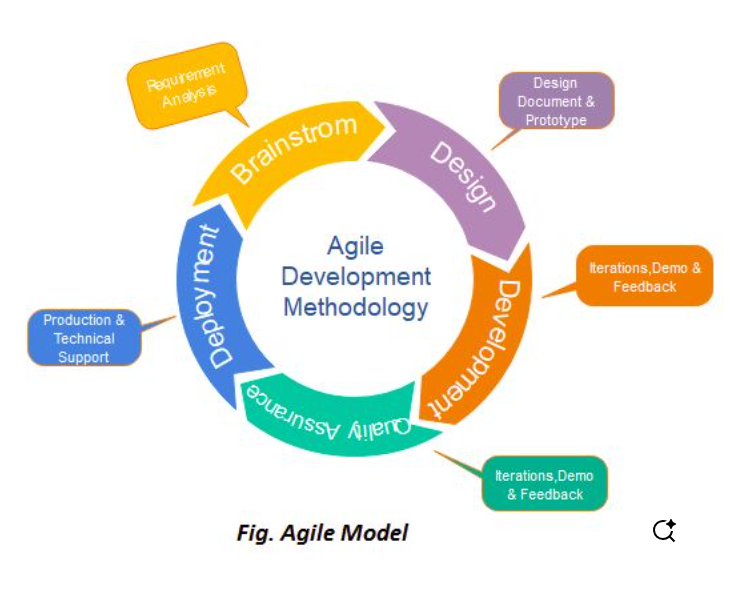
**Agile Model:**

Agile allows stages to overlap and includes frequent communication with stakeholders to continually refactor the requirements and other important factors.

The model is used where requirements are not fully defined.

**Application:**

This model is used where applications are large and rapid feedback is required to make the changes in the project. Complex projects can be created using agile models. Used in web apps, and startups.



**Advantage:**

Faster delivery of usable software because a working software is created in the beginning

Flexibility to adapt to changes

Disadvantage:

Less documentation can lead to confusion

It can be challenging to predict the timeline

**Extreme Programming model:**

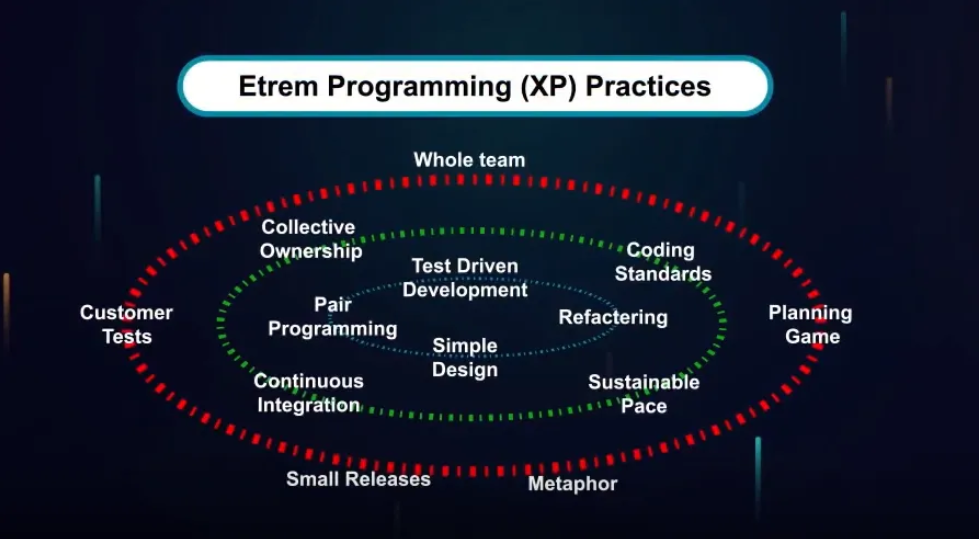
Standard Agile principle taken to the extreme

XP works when small and collaborative teams.

Pair programming is used where one programmer writes the code and other observes

**Application:**

Facebook used XP-like practices like quick releases, constant user feedback, pair programming, and continuous integration to rapidly improve its platform.



**Advantages:**

High quality code because one observes and other works

Adaptable to change

Reduced risk

**Disadvantages**

High resource requirements

Steep learning curves and limited documentation.

**Spiral Model:**

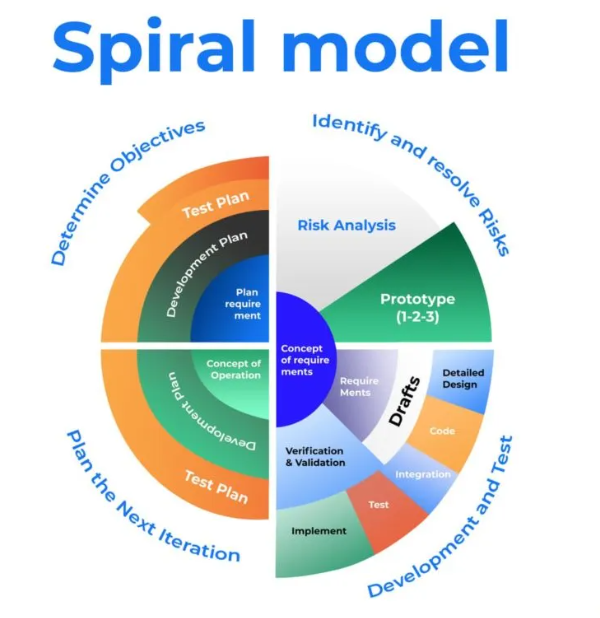
The Spiral Model is like building Application step-by-step in circles.

Each circle means planning, building, and checking the work to catch problems early.

It repeats these steps again and again until the final product is ready.

**Application:**

Used in large and complex software projects like banking systems, aerospace software, or government projects where regular testing and risk checks are important.



**Advantages:**

Catches risks early through regular testing and reviews

Flexible to changes at any stage of the project

Focuses on risk management in each cycle

**Disadvantages:**

Can be expensive because of repeated steps

Not ideal for small or simple projects

**What is Scrum in Agile**

Scrum is a way for a team to work together to build a software step by step. It helps the team stay organized, work faster and make improvements in the software. Work is done in short periods called sprints. At the end of each sprint team has a small part of the software ready.

**What is Spring**

Sprint is a fixed time period between 1-4 weeks. Planning is done keeping in mind the goal deliverable at the end of each period.

**Do’s**

Focus on the sprint goals. It helps keep the team aligned

Work collaboratively. support each other to remove blockers.

Use feedback from sprint reviews to improve the product and how the team works.

**Don’ts**

Do not add new work mid way

Do not try to solve everything on your own.

Do not hide the problem. Bring up blockers early during the Daily Scrum.

**What are stories and backlogs**

**Stories**

User Stories are short, simple descriptions of a feature or task from the user’s perspective.

They help the team understand what needs to be built and why.

As a customer, I want to reset my password so that I can log in if I forget it.

**Backlog**

A backlog is a to-do list for the team. It is unfinished work that’s yet to be done.

It contains user stories, bugs, tasks, and features that need to be worked on.

**Scrum Artifacts**

**Product backlog:**

It is a master to do list for the product.

It has all the features, bug fixes and feature improvement that the team will work on in the future.

**Sprint backlog:**

It is a smaller list taken from the product backlog.

It is owned by the development team.

**Increment:**

The working product or part of the product that is done by the end of the sprint.

It’s fully tested and ready to use or release.

**Burn-Down Chart:**

It is a simple graph that shows how much work is left in the sprint.

It helps the team to track the progress and see if they’re on schedule.

**What are Ports and Protocols?**

**Port**

A port is like a door on your computer or server that allows different kinds of data to come in or go out. They make sure the right application gets the right data.

Each type of service like websites, email, games uses a different port number.

Port 80 for HTTP

Port 443 for HTTPS

Port 25 for SMTP

Protocol:

A protocol is a set of rules that computers follow to talk to each other.

It decides how data is sent, received, and understood. Without protocols, devices don't know how to talk to each other.

HTTP : for loading web pages

FTP : for file transfers

TCP/IP : for internet communication

What are different types of networks?

PAN - Personal Area Network

Very small network for personal use. It connects devices like phones, laptops, headphones via Bluetooth or USB.

LAN - Local Area Network

Covers a small area like a home, office, or school

It connects computers, printers, and devices with the help of cables or Wi-Fi

WAN - Wide Area network

Covers a very large area, even across countries or continents. Connects multiple LANs.

The internet is the biggest example of a WAN

WLAN - Wireless LAN

A wireless version of LAN using Wi-Fi

No physical cables needed

What are different types of Servers

Web Server

Hosts websites and delivers web pages to users.

Uses protocols like HTTP or HTTPS.

Example: Apache, Nginx

Mail Server

Sends, receives, and stores emails.  
Works with protocols like SMTP, IMAP, and POP3.

Example: Microsoft Exchange, Postfix

Application Server

It Runs business applications and handles logic between the front end and database. It Used for complex systems like ERPs or mobile apps.

Database Server

Stores and manages data in a structured way. Responds to data requests from other software or apps. MySQL, Oracle, Microsoft SQL Server

DNS Server

Translates domain names into IP addresses

It’s like the internet’s phone book.

File Server

It Stores and shares files over a network. Users working on the same network can upload, download, and edit files.

What do you know about DNS?

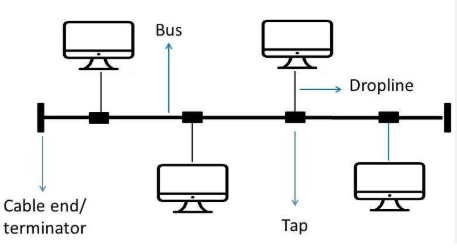
It’s like a phonebook for the internet. Since computers do not understand names, it needs an IP address to reach the server and process the request. DNS finds that number.

**What are different types of Network Topologies?**

**Bus Topology**

All devices share a single cable to connect.

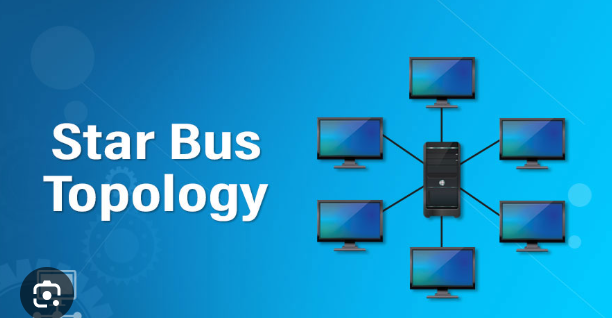
It is easy to set up and cheap however if cable disconnects, all systems will disconnect.



**Star Topology**

All devices are connected to a single hub.

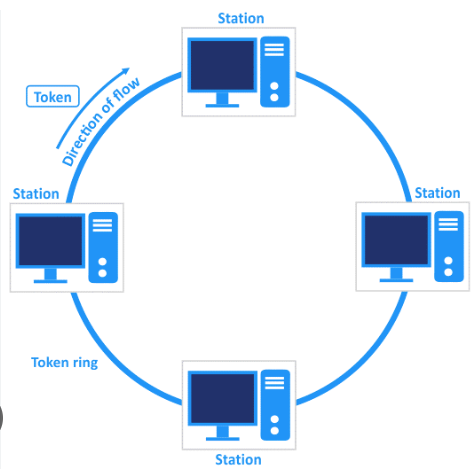
If the central hub fails, all device connections are lost.



**Ring Topology**

Each device is connected to its two neighbours.

Data moves in one direction.



**Mesh Topology**

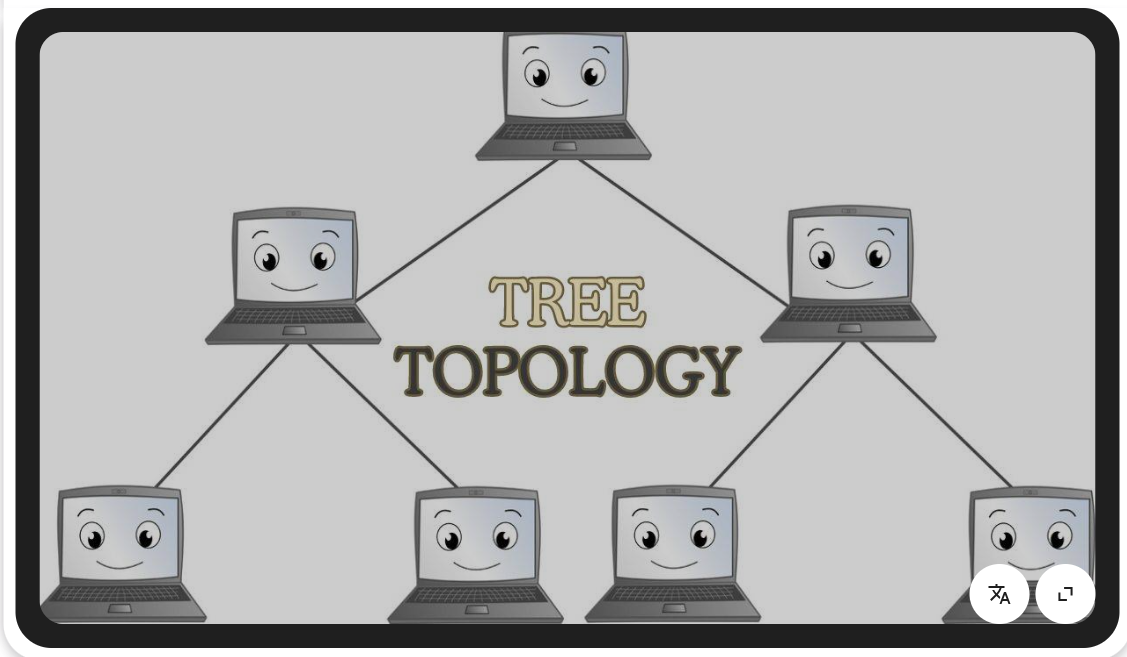
Every device is connected to the other devices

It is very reliable however very expensive to set up



**Tree Topology**

It is a combination of bus topology and star topology. It is easy to expand and it maintains a hierarchical structure.



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

The OSI Model is like a big rule book that explains how different computers and devices talk to each other over a network. It has 7 layers.

1. **Application Layer:**

This is the first layer of the OSI model. Here, the user interacts with web applications, email or apps. HTTP requests are handled on this layer

1. **Presentation Layer**

It translates the data into language a device understands. Encryption and compression is done on this layer.

1. **Session Layer**

It manages the conversation between two devices, it creates a session using API and sockets between two devices.

1. **Transport Layer**

It makes sure the data gets delivered correctly. It splits big messages into smaller parts, then reassembles them at the other end. TCP/UDP are used here.

1. **Network Layer**

This layer decides where the data should go. It uses addresses to send the data to the correct destination. Data is transferred in the form of small packets.

1. **Data link Layer**

It takes the raw bits from Layer 1 and organizes them into small packages called frames.

It also handles error checking to make sure the data isn’t messed up during transmission. It is responsible for MAC addressing.

1. **Physical Layer**

This layer is the actual hardware and cables. It handles the raw movement of data bits.