Day 1 - 16th May 2025

Your Doc name: SDLC

Id: hkumarzh (109940031)

Name: Hemant Kumar

Task: 1

**What is SDLC?**

SDLC: Software Development Life Cycle which starts with gathering the requirement till delivering the product which includes the various stages involved in this process.

Gathering information,

Understanding requirement, deciding in which model it will fall such as waterfall, agile etc

Development stage

Various testing methods

And ensuring maintained post-deployement.

Task 2:

**Why is SDLC ?**

Without sdlc, small projects are fine but to understand what we have to do, how it will perform, what are the software/technical requirement, what are the possible way user going to use it, how long and at what level my software will survive in terms of platform updation or other technical changes, How it will be maintain in future for these , SDLC is required.

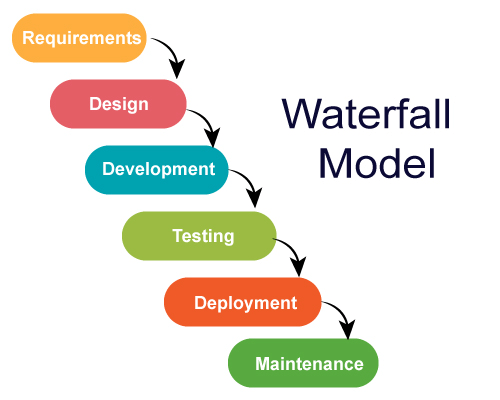
Task 3:

**What are the stages in SDLC?**

1. **Requirement Gathering** – Gathering the complete requirement, BRDs
2. **Analysis and Planning** – Analyzing the requirement, understanding dependencies involved, cost involved etc.
3. **Design** - Creating design document (About architecture, database design, technology stack which we will be using etc.)
4. **Development** – initiating development of the application
5. **Testing** - Testing the application at various level including internal testing and UATs
6. **Deployment** of the application
7. **Maintenance** - Maintaining the application which includes bug fixing, feature upgrades, upscaling etc.

**SDLC Models:**

1. **Waterfall Model:** It is sequential downward flow which basically used when requirement is kind of fixed not changing frequently so that all the SDLC stages are fixed and not overlapping. It goes downward from planning phase till testing phages.



Example Applications:

* Migration of applications from one platform to another

Advantages:

* Easy to manage since requirement are fix
* Easy documentation and understanding of the requirement

Disadvantages:

* As it goes stage wise, testing of the application be done only at the end. Which can be high risk or extra effort from start if anything goes wrong
* No scope of changing requirement

1. **Agile Model:** When requirements are not clear. This divided in phases, basically for sort term projects at each phase, this also require continuous stakeholder involvement to understand and make development basis request change and stakeholder feedback.



Example Applications:

* Web applications such as shopping websites, e-commers websites
* Mobile applications, games etc

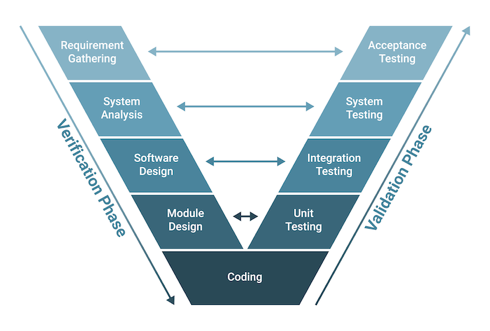
Advantages:

* Flexibility of changes, Stakeholder can get prototype at early stage
* Regular testing

Disadvantages:

* Unclear requirement can involve large development time
* Resources prediction is also difficult as it depends on requirement

1. **V-Model:** Similar to waterfall model however this involves continuous testing of application at functional level at each stage of development.This involves basically critical software’s such as



Example Applications:

* + Aviation systems
  + Banking sofwares

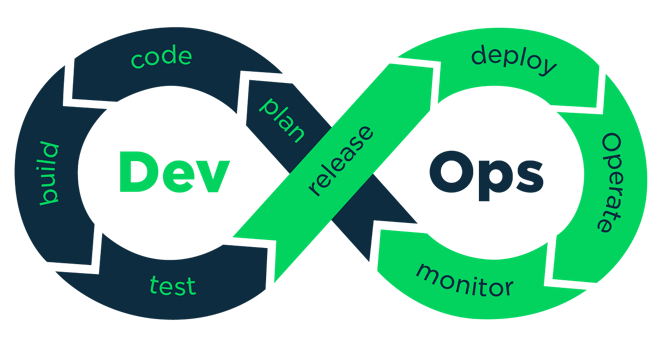
Advantages:

* Testing at every stage ensure correct development of the product
* Bugs can be identified at early stage

Disadvantages:

* Time Consuming
* Prototype will take time to be ready for stakeholders

1. **DevOps:** In this, CI/CD pipelines used, like continuous integration and deployment for frequent changes within the application while maintaining the User level application up and running. This mostly involves cloude applications.



Example Application:

* Applications using cloud services (AWS, Azure etc)
* Serverless websites
* Apis, deployed on cloud etc

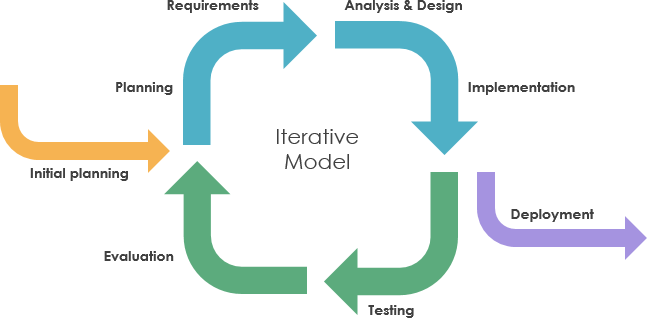
Advantages:

* Features and bugs can be fixed behind the scene without impacting application usages
* Using cloud services can help adjust resources dynamically

Disadvantages:

* Using cloud service can be expensive

1. **Iterative Model:** This model used for large projects, starts with basic functionality, divided in phases and require iteration of planning, requirement gathering, testing etc for each stage.



Example Application:

* Video games which bring new feature updates time to time on existing functionality post release.

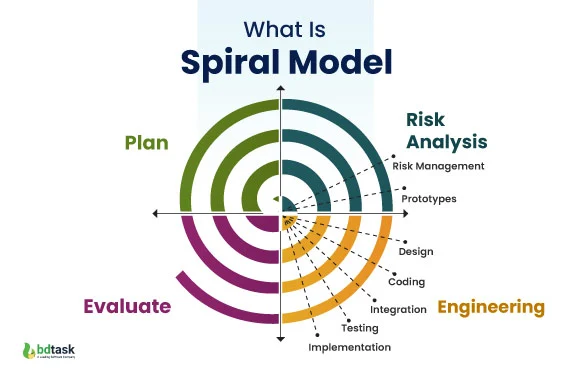
Advantages:

* User version can be available at early stage and features will be added later stages

Disadvantages:

* It can cost more or time consuming since all the stages follows at every next step of development

1. **Spiral Model:** This model used for larger projects where a thorough checking of bugs and problems required before moving to every next stage of development. Basically consist of multiple waterall model we can say.



Example Application:

* Defense system which are critical, and risk and other stage needs to be evaluated at every next stage of integration and updation

Advantages:

* Early risk identification
* Less project failure chances due to evaluation at each stage

Disadvantages:

* Complex structure
* Can take longer time

**What is Scrum?**

Scrum is an Agile framework that helps teams to plan and work on complex projects through fixed time meetings (sprints calls). It includes daily standups, sprint planning, and updates discussion calls or meetings to track progress and address challenges.

**What is Sprint?**

Sprint is basically a fixed time period given to team to complete particular part of the project and at every new sprint team discuss on progress and challenges of the given task.

**Points on Do’s and Don’t while working on Sprint?**

**Do’s:**

* Stick to the end goal
* Update the progress on daily basis
* Highlight blocker if any on daily standup

**Don’ts:**

* Not updating progress of the task
* Not participating in daily standup
* Not highlighting blocker on time
* Daviating from the main assigned task

**What is Backlog?**

Backlog is overall task which needs to be completed for product completion.

**Stories:**

Stories are main task which includes multiple subtasks or breaking main task in small chunks where story is main task and subtask are chunks.

**Product backlog?**

Basically what are the requirements , enhancement and fixes of the main product is considered as product backlog

**Sprint Backlog**

Sprint backlog is what needs to be completed in particular sprint. Basically, a part of product backlog assigned in sprint

**Increment:**

Total number of backlogs of the product which is part of any sprint which is ready to use.

Networking :

**What is the port?**

It is basically a 16 bit number ,which specify the specific service or the process of the computer

**What is protocols?**

Protocols are rules or procedures that specify how data should be transfer or how devices will be communicating to each other.

**What are the different network types?**

* LAN
* WAN
* VPN
* Client-Server Network
* PAN
* Wired network
* Wireless Network

**What are the types of servers ?**

* **Web server**

A web server delivers web pages to users' browsers. For example when we type website address, the web server sends back the requested pages.

* **Database Server**

Stores and manages data for applications. It's like a digital filing cabinet that organizes, stores, and retrieves data when needed. Examples include MySQL and Oracle, which handle everything from simple data storage to complex transactions.

* DNS server

Works like the internet's phone book. When you enter a website name (like www.google.com), the DNS server converts it to an IP address that computers can understand. Without DNS servers, we'd need to remember IP addresses for every website.

* DHSCP server

Automatically assigns IP addresses to devices on a network. When your device joins a network, the DHCP server gives it an IP address and other network settings. It's like an automated receptionist assigning room numbers.

* FTP server

Specialized for file transfers over networks. It allows users to upload, download, and manage files between computers. Think of it as a digital courier service for files.

* Proxy server

Acts as an intermediary between users and the internet. It can filter content, improve security, and cache frequently requested data. Like a security guard and assistant that manages internet traffic.

* Cloude server

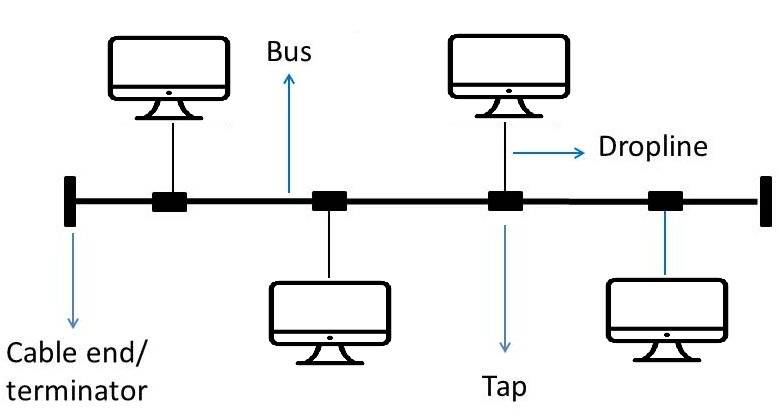
Virtual servers hosted in cloud infrastructure like AWS or Azure. They provide scalable resources accessible over the internet. Think of them as rental computers that can grow or shrink based on needs.

**What do you know about DNS? Domain Name System**

The Domain Name System is a hierarchical naming system for computers, services, or other resources connected to the Internet or a private network.

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

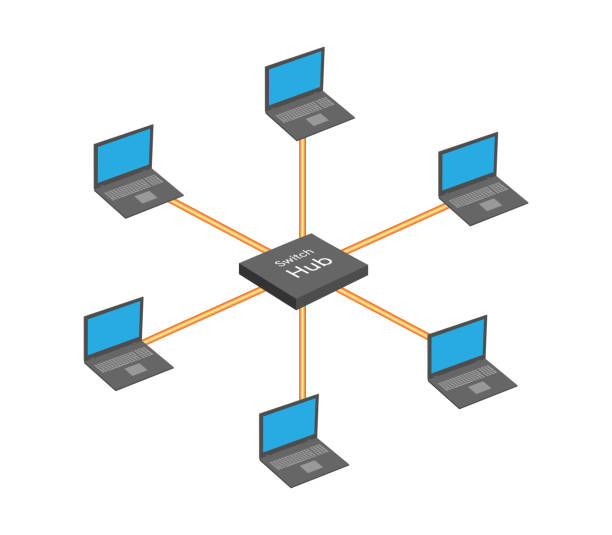
1. **BUS Topology:**



All devices connected to single host or say central cable.

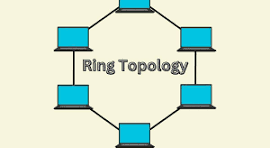
And if main central cable fails, all the connected network fails.

1. **STAR Topology:**



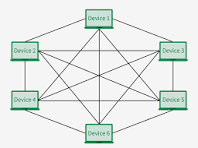
All devices connected to a hub. Here easily devices can be connected or disconnected from the hum without impacting other connected devices.

1. **Ring Topology**



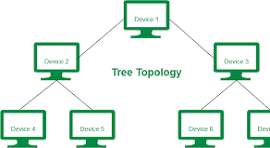
Each device will be connected to each other and forming a ring shape.

1. **Mesh Topology**



Each device connects to every other device available in that network.

1. **Tree Topology**



Hierarchical structure like a tree

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

OSI (Open System Interconnection ) Model -> A framework that standardizes network communication functions into seven distinct layers.

Layers:

1. **Application Layer**:- This layer provides network services directly to end-users or applications. It includes protocols like HTTP, FTP, SMTP, and enables user authentication, data sharing, and application services.
2. **Presentation Layer**: - Handles data formatting, encryption, and compression. This layer ensures data is readable by the receiving system and manages data syntax translation between different systems.
3. **Session Layer**: - Establishes, maintains, and terminates connections between applications. It handles session restoration through checkpointing and recovery if interruptions occur.
4. **Transport Laye**r: - Ensures complete data transfer and manages reliability of communication through flow control, segmentation, and error checking. TCP and UDP operate at this layer.
5. **Network Layer**: -Handles packet routing and logical addressing (IP addresses). This layer determines the best path for data delivery and manages traffic control across networks.
6. **Data Link Layer**: -Provides node-to-node data delivery using physical addresses (MAC). It handles error detection, flow control, and access to physical media.
7. **Physical Layer**: - Transmits raw binary data over physical network hardware. This layer defines electrical, mechanical, and physical aspects of the physical network medium.