**Date:** *14-May-2025*

**Name:** *Vivek Samant*

**Empid:** *109085619*

**alias:** *samantvs*

**What is SDLC?**

SDLC stands for Software development life cycle. It is a framework that encompasses the whole process of building a software from the scratch to deploying and maintaining the software and providing constant customer support post the launch. There are various stages to it: Planning, Designing, testing, deploying, maintaining and at each step there are various stakeholders involved that provide valuable inputs. This helps in building great software via collaborative work. There are various models under SDLC such as Waterfall model, Verification and validation model, Iterative model, Agile model, XP model. These find uses in various scenarios. Some of these are rigid and sequential which are suited for predefined software needs and some are very flexible like Agile, XP that are suitable for evolving scenario.

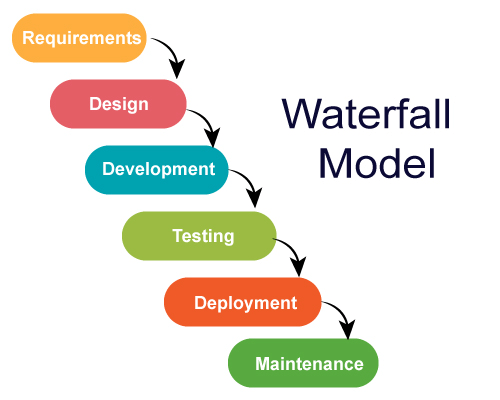
**Why is SDLC required?**

SDLC enables the team to deliver great software product within deadline, under the budget, and with expected requirements and functionality. It enables collaboration between different stakeholders and maximizes customer satisfaction. It brings into the development a paradigm of *planning* that is useful not just in developing, but even post the deploying and maintenance.

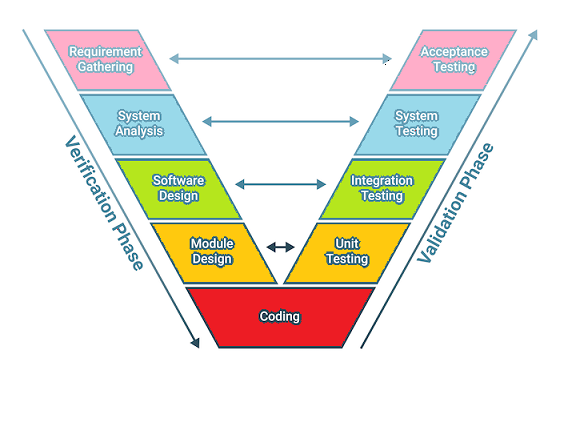
**Different steps of SDLC:**

1. Planning & requirement analysis, defining requirements stage
2. Design (system, high level)
3. Implementation (writing code)
4. Testing
5. Deployment and customer support
6. Maintenance after market (introducing upgrades regularly, extending support to different environment)

**Different models of SDLC:**

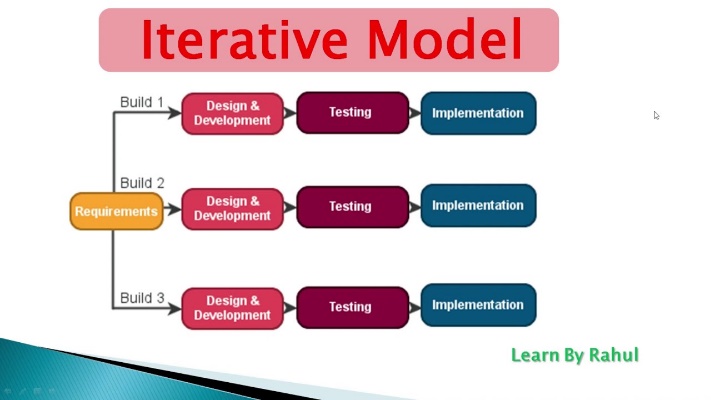
1. **Waterfall model:** This is a rigid model wherein each step of the cycle is defined rigidly. The steps followed are sequential. This is generally followed when there are clear and fixed requirements. The project is short and simple and the environment is stable.
2. **Verification and validation model:**

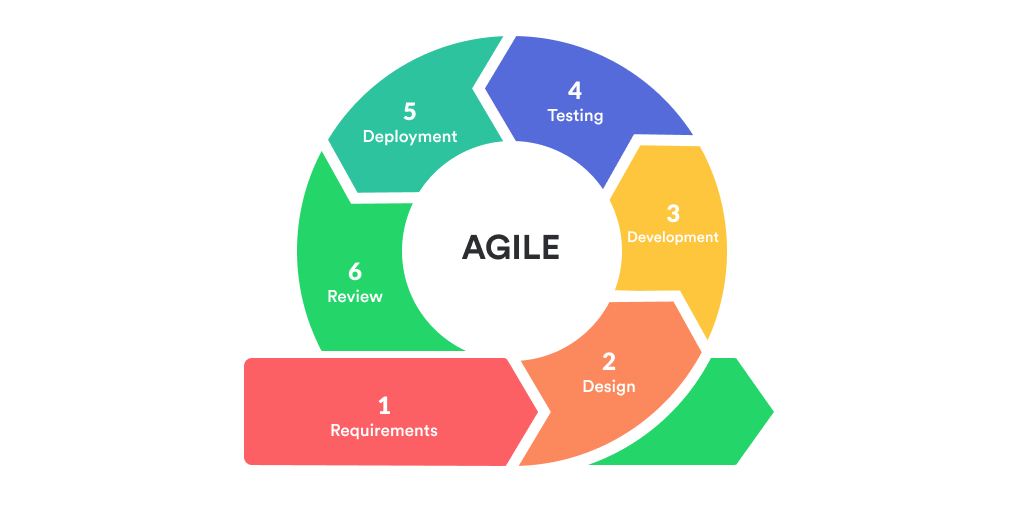
This model follows a paradigm wherein each development stage is followed by verification and validation simultaneously. Each development phase has a corresponding testing phase, creating a V-shaped structure that emphasizes parallel verification and validation activities. Verification ensures the software is built correctly according to the specifications whereas validation ensures the software meets the actual user needs and requirements.

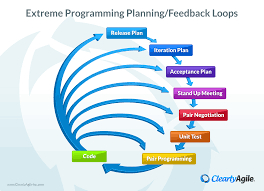


1. **Iterative model:**

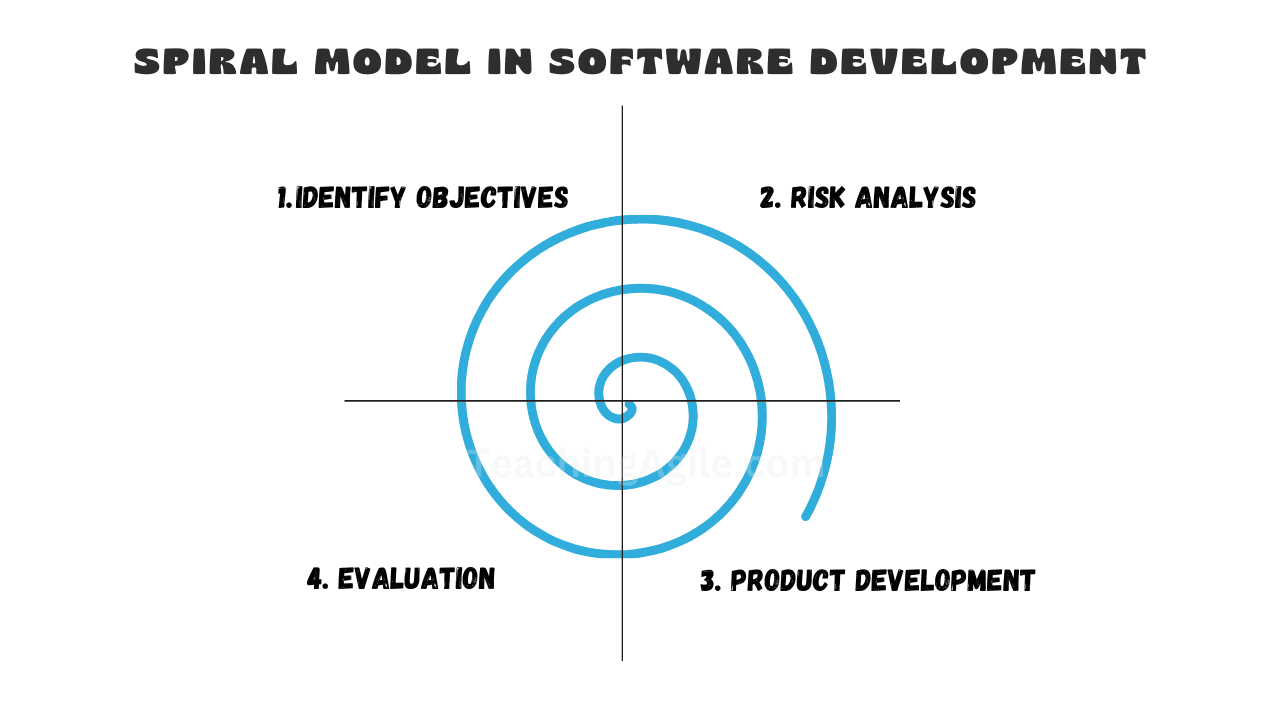
This follows a paradigm such that cycle for *each component* mimics a full development cycle (full SDLC cycle i.e.), from planning and design to testing and deployment. This works great in sense that there is early user feedback, and easier bug detection and flexible. However, it tends to be costlier due to so many iterations, requires too much inputs from multiple stakeholders and can lead to project delays due to so many repeated cycles.



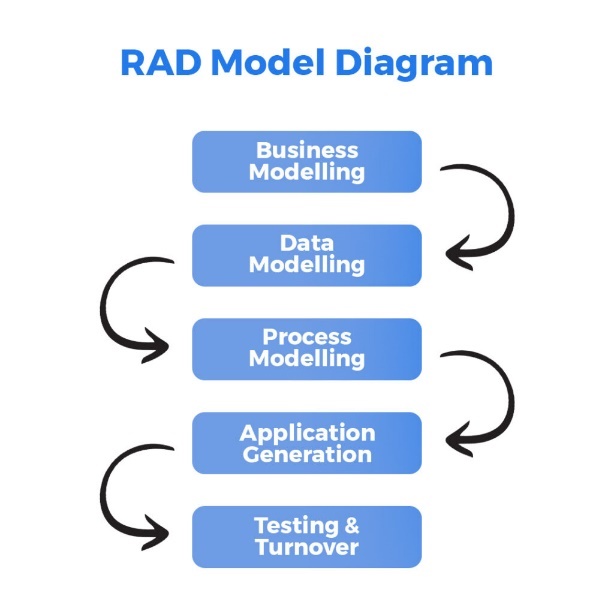
1. **AGILE methodology:**  It is a development approach that follows a paradigm of working on one module at a time. The product is divided into many incremental modules and it is supposed to start with smallest module at the beginning. It is also iterative and teams have sprint with well defined goals and go over and above the bar to achieve that aspect within defined timeframe.
2.  **Extreme programming (XP) model:** This is based on agile principles but extends them to extreme levels. It is test driven development and has continuous integration. The drawback being it requires very skilled developers and learning curve is steep for this.



1. **Spiral model:** It combines iterative development with waterfall. It is good for large, complex projects. It has generally 4 phases: Planning, risk analysis, engineering, evaluation.



1. **RAD (Rapid Application development) model:** It’s main focus is on rapid prototyping. It has very minimal planning, quick development cycles and heavy user involvement. It is best for project with immediate deadlines or very short timelines.



1. **Prototyping model:** It creates early working version (known as prototype). This is a preliminary version upon which further improvements are made basis user feedback.



**Different network types:**

Based on area coverage:

Personal area network (PAN), Local area network(LAN), Metropolitan area network (MAN), Wide area network(WAN),

Based on connection type:

Wired: Ethernet, fiber optic, Coaxial cable

Wireless: Wi-Fi, cellular, satellite

Based on network architecture:

Peer-to-Peer(P2P) – no servers, direct communication

Client-server – controlled access to client

Based on network relationships:

Intranet, Extranet, Internet(Global public)

Based on network access:

Private, public, hybrid

Based on network topology:

Bus, star, ring, tree

**Types of servers:**

Web servers:

Used to access websites and web apps. It is done by establishing connection via HTTP/ HTTPS.

Database servers:

It stores and manages data and provides a way to access that data. For e.g.: via sql

Mail servers:

Used to send, store, receive email.

DNS servers:

Very useful and form basis of accessing websites via string. It translates domain names of websites to IP addresses so a connection can be established with remote server via that IP and port so user can access website.

Proxy servers:

It acts as a proxy between client and other remote server so client’s credentials are not exposed while establishing connection.

**Task 8: What is TCP and UDP? What is the difference?**

TCP and UDP both are protocols used on transport layer of networking to establish a connection between a client and a remote server.

TCP/ Transfer control protocol:

It is connection-oriented. It is slower and adds more overhead. It has flow control and recovery from any transmission error/ losses.

E.g.: SSH, HTTP, FTP, IMAP

UDP/ User datagram protocol:

It is a connectionless protocol. It is faster because it adds very low overhead compared to TCP. It has no follow control and no way of recovering any data loss/ error in transmission.

E.g.: DNS, DHCP, NTP, Real time apps (like skype, discord voice)

**Task 9: What do you know about mac address? What is the difference between Mac address and IP address?**

Mac address is a unique address assigned to each network interface card by the mfg. It operates at Layer 2 (data link/ ethernet) layer of the OSI model

Unicast, multicast, broadcast.

It is made up of 48 bits or 6 octets; *First 3 octets*: Organizationally unique identifier. *Last 3 octets*: network interface controller (NIC) specific

IP address is a unique logical address that identifies a device on a network using the internet protocol. It works on network layer of OSI model whereas MAC addresses are meant to be globally unique and permanently assigned by the manufacturers whereas IP in the private network can be same as IP in another private network. It is only while connecting to internet a public IP is used which is used to identify the network.

**Task 10: What is OSI model?**

IT stands for *Open systems interconnection model*. It is a concept that is a standard for network communication. It has 7 distinct layers:

PDU: protocol data unit – smallest unit of data

7 : *Application* –human interface (pdu: data) -- http, ftp --- data

6: *Presentation* *layer:* responsible for translation, encryption. –

for eg: a website has meta charset = utf-8 which means it encodes the the content of the page in the format (pdu: data)

5. *session*: responsible for session establishment and termination (pdu: data)

4. *Transport*: data transmission over network (using tcp or udp) (pdu: segments) - data+ segment header

3. *network*: chooses best routes to destination, routes packets (pdu: packets) - data+ segment header + packet

2. *data* link: deals with errors on physical layer, responsible for flow control (pdu: frames) - data+ segment header + packet+ frame

1. *physical* – transmission on physical media (pdu: bits)

Data encapsulation 🡺 adding headers

Dta decapsulation 🡪 on receiver side

**Task 11: What is an IPV4 and what are different classes of it?**

It stands for internet protocol version 4. It is a networking address protocol uses 32 bit addresses. Traditionally 5 classes:

1. Class A – first bit 0. Range 0.0.0.0 to 172.255.255.255 (0-127) – used for very large networks

2. Class B – first bits 10. Range 128.0.0.0 to 191.255.255.255 (128-191)– used for medium to large networks

3. Class C – first bits 110. Range 192.0.0.0 to 223.255.255.255 (192-223)– used for small networks

4. Class D – first bits 1110. Range 224.0.0.0 to 239.255.255.255 (224-239)– multicast

5. Class E – first bits 1111. Range 240.0.0.0 to 255.255.255.255 (240-255)– experimental reserved

Modern networks do not rely on these and instead use classles inter domain routing (CIDR) which is more flexible and efficient allocation of IP through subnet mask.

**Advantages of using VPN:**

1. Masking location and maintaining online anonymous, encrypting comm
2. Maintain privacy and safety from third party hackers
3. Access localized services (by switching location virtually to destination where one requires services without physically being there- e.g.: Netflix content access from FR while residing in IN)
4. Better performance sometimes (e.g.: moving ahead of ISP bandwith restriction)

**Main types of VPN:**

Remote acess VPN : connects user to private network. Like usually enjoyed by company employee. Eg: cisco,

Site 2 site VPN : connects multiple fixed locations like offices.

It has subtypes:

*Intranet-based*: connect branches of same company

*Extranet-based:* connect company with partner/ suppliers

**Task 19: Different types of network topology:**

1. BUS : all devices connected to a single central cable(backbone).(like connecting bunch of led’s in serial). One device failure results in all. Central cable is limited in length and performance goes down with big traffic.
2. STAR : devices connected to a central hub/ switch(imagine a network swtich). If hub fails, then entire network also fail. Easy to add new devices. Common today.
3. RING : devices connected in a circular chain. Data travel in one direction. Each device act as repeater. Good for performance with heavy traffic but less common today.
4. MESH : every device connected to every other device like a weave or mesh. It is complex to manage. 2 subtypes: partial mesh, full mesh.
5. TREE : hierarchial structure. Combination of bus and star. Used in large networks.
6. POINT2OPINT : simple direct connection b/w 2 devices. It is simple but has very limited use and scalibility
7. Hybrid : simply a combination of any 2 or more typologies

**Task 18: Use of a router and difference b/w router and gateway:**

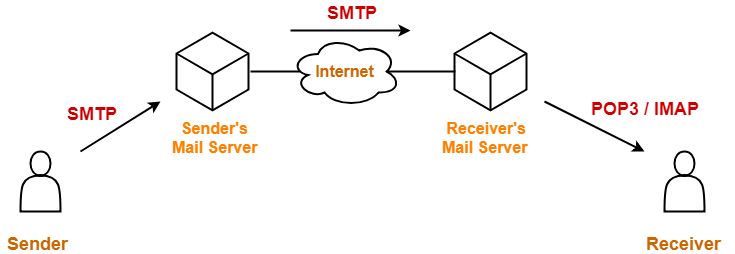
Router : networking device that acts as a traffic controller and directs data b/w devices and the internet ensuring information reaches the correct destination. It establishes this via forwarding data packets b/w computer networks and it also assigns IP via DHCP protocol. It also build routing tables.

It establishes a LAN and making it possible for hosts connected to it to communicate between them as well as internet if router has internet connectivity.

Difference: Gateway act as entrance/ exit point b/w 2 different networks. it translate data between different types of networks/ protocols and work across multiple layers. Routers focus on routing data packets while gateways serve as protocol converters and entry/ exit points between dissimilar networks.

Ex of gateway: payment, mail.

**Task 19: Smtp: for sending mail only**



**Task 20 : Difference between OSI and TCP/IP**

TCP:

Stands for transmission control protocol/ internet protocol. Developed by US defense department.

It has 4 layers: Application, transport, internet, network access

It is practical and widely used in *real networks*

More integrated, less modular than osi.

e.g; ftp, http

OSI:

Stands for open systems interconnection developed by ISO (international standards organization)

It has 7 layers: application, presentation, session, transport, network, data link, physical

It is more of a theoretical model best suited for detailed understanding and design of network architecture.

Each layer is independent, making it easier to replace protocols at one layer.

No specific example protocols

**Task 21 : http and https (oral) – It was accomplished successfully.**

**Task 22 : Low level design (LLD) and high level design(HLD)**

HLD:

It focuses on overall system architecture and its major components. It deals with system blocks, db architecture, tech stack, interfaces, how different modules might interact.

It addresses things like scalability, security and data flow without deep diving on implementation details. It is basically ‘What to build?’

LLD:

It is about detailed implementation. It includes specific class diagrams, DS, algos, method signs, interface definitions etc. It is basically ‘How to build?’ and is the next step after HLD is finalized.

**Task 23: SRS (Software Requirements Specification):**

It is a document describing complete requirements for a software system. This includes things such as functional requirement, non-functional requirement, system constraints, UI requirements, scope, use cases.



**SDLC MCQ**

1.

A feasibility study using the SDLC model is conducted to

determine whether or not the project is technically possible

determine whether the proposal is financially viable

Both a and b

None of the above

2.

A well-documented life cycle model aids in the detection of what during the development phase?

Inconsistencies

Redundancies

Omission

All of the above

3.

How many lines of code does the Build & Fix Model suit for programming exercises?

100-200

300-400

600-700

Above 800+

4.

In which life cycle does regression testing play a significant role?

Waterfall model

V model

Iterative model

All of the above

5.

What determines if the project should go forward?

feasibility assessment

opportunity identification

system evaluation

program specification

6.

What is the most significant disadvantage of employing the RAD Model?

Developers/designers that are highly specialized and skilled are required.

Component reusability is improved.

Encourages client/customer input.

Increases component reusability.

7.

Which of the following developmental models is incremental?

Prototyping, V model, Agile

Prototyping, RAD, Agile, RUP

Prototyping, V model, RAD, Agile, RUP

All of the above

8.

Which of the following is an Agile development characteristic?

Shared code ownership

Test-Driven Development

Implement the simplest solution to meet today's problem

Continual feedback from customer

All of the above

9.

Which of the following steps in the SDLC framework are valid?

Requirement Gathering

Software Design

System Analysis

All of the above

10.

Who is in charge of system development, staffing, budgeting, and reporting, as well as ensuring that deadlines are met?

Project managers

Network engineers

Graphic designers

Systems analysts