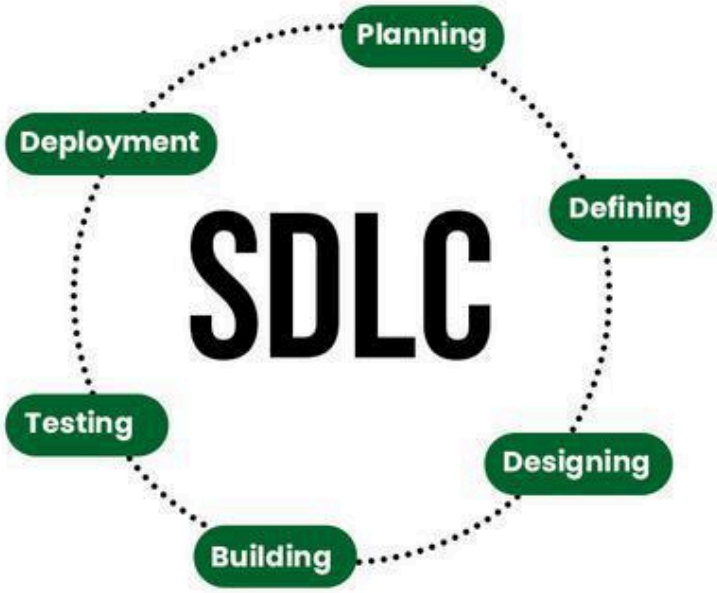
Software development life cycle (SDLC) is a structured process that is used to design, develop, and test good-quality software. SDLC, or software development life cycle, is a methodology that defines the entire procedure of software development step-by-step. The **goal of the SDLC life cycle model** is to deliver high-quality, maintainable software that meets the user's requirements. SDLC in software engineering models outlines the plan for each stage so that each stage of the software development model can perform its task efficiently to deliver the software at a low cost within a given time frame that meets users requirements. In this article we will see Software Development Life Cycle (SDLC) in detail.

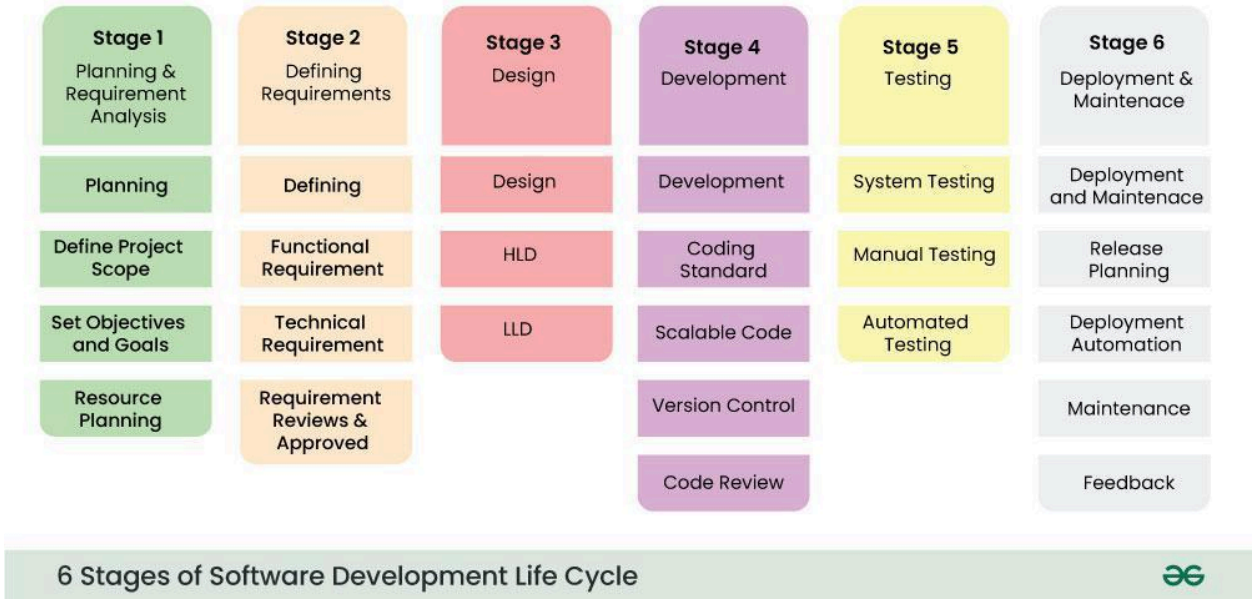
What is the Software Development Life Cycle (SDLC)?

SDLC is a process followed for software building within a software organization. SDLC consists of a precise plan that describes how to develop, maintain, replace, and enhance specific software. The life cycle defines a method for improving the quality of software and the all-around development process.



Stages of the Software Development Life Cycle

SDLC specifies the task(s) to be performed at various stages by a software engineer or developer. It ensures that the end product is able to meet the customer's expectations and fits within the overall budget. Hence, it's vital for a software developer to have prior knowledge of this software development process. SDLC is a collection of these six stages, and the stages of SDLC are as follows:



6 Stages of Software Development Life Cycle

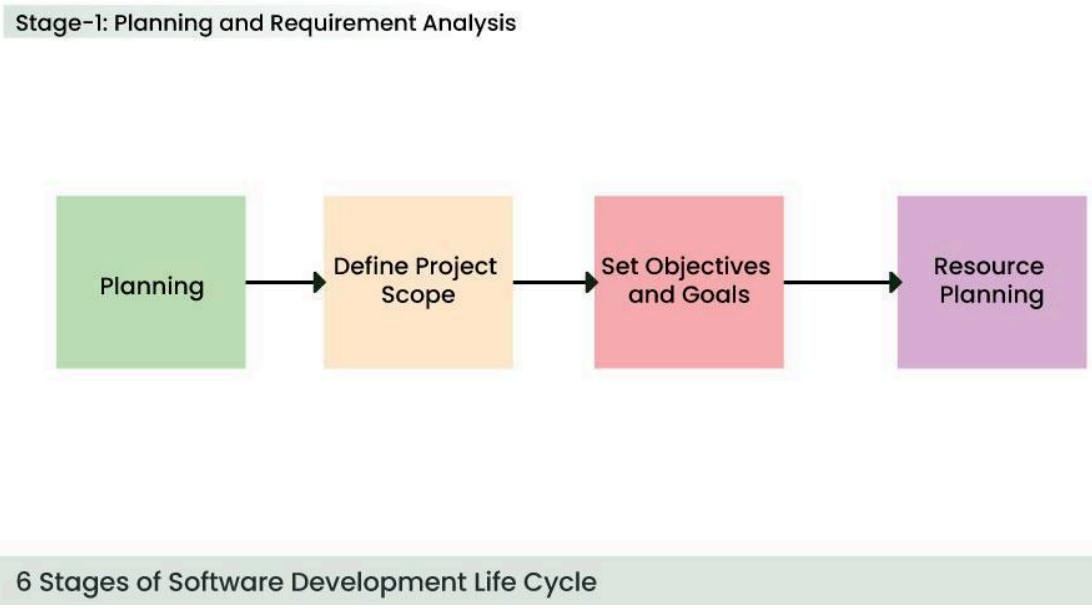
Software Development Life Cycle Model SDLC Stages

The **SDLC Model involves six phases or stages** while developing any software.

Stage-1: Planning and Requirement Analysis

Planning is a crucial step in everything, just as in software development. In this same stage, requirement analysis is also performed by the developers of the organization. This is attained from customer inputs, and sales department/market surveys.

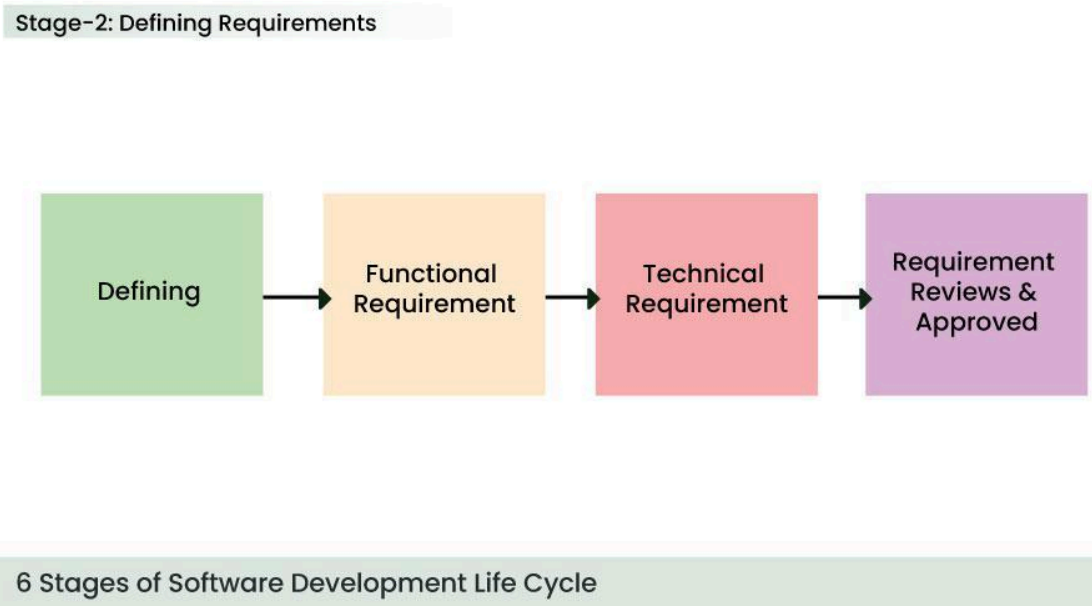
The information from this analysis forms the building blocks of a basic project. The quality of the project is a result of planning. Thus, in this stage, the basic project is designed with all the available information.



Stage-1 : Planning and Requirement Analysis

Stage-2: Defining Requirements

In this stage, all the requirements for the target software are specified. These requirements get approval from customers, market analysts, and stakeholders.
This is fulfilled by utilizing SRS (Software Requirement Specification). This is a sort of document that specifies all those things that need to be defined and created during the entire project cycle.

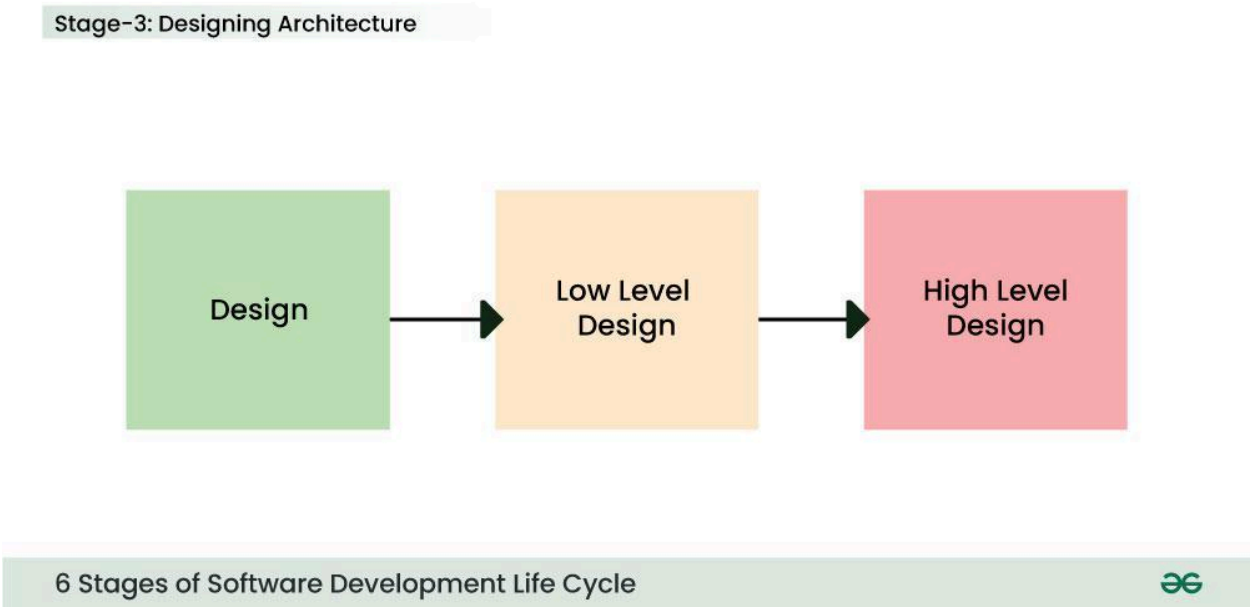


Stage-2 : Defining Requirements

Stage-3: Designing Architecture

SRS is a reference for software designers to come up with the best architecture for the software. Hence, with the requirements defined in SRS, multiple designs for the product architecture are present in the Design Document Specification (DDS).

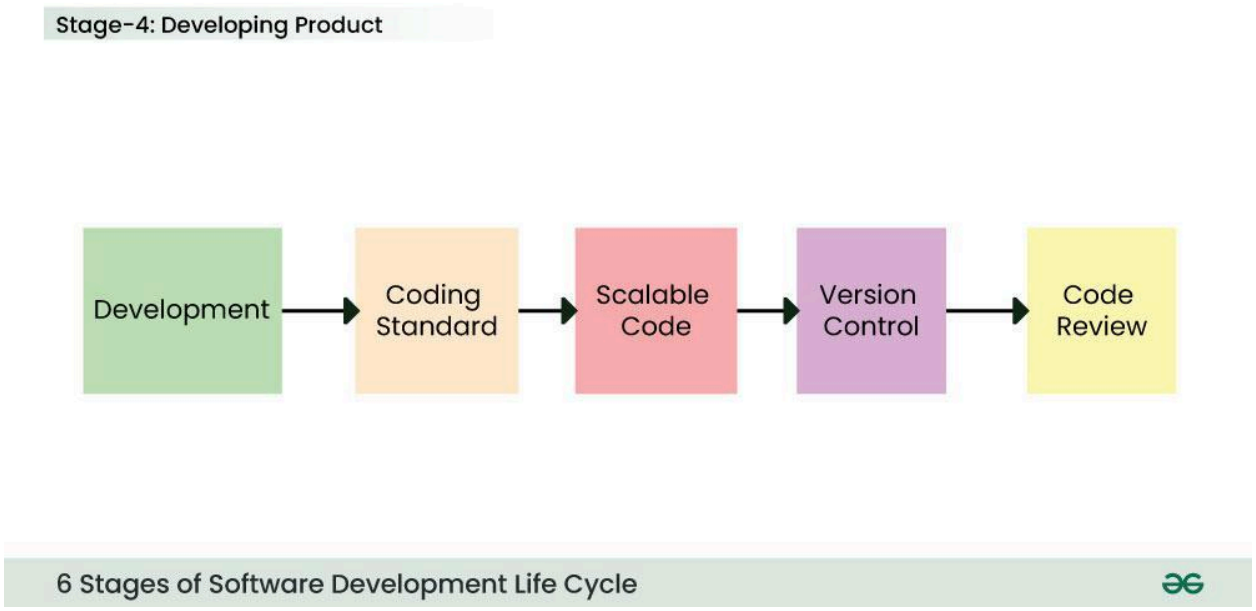
This DDS is assessed by market analysts and stakeholders. After evaluating all the possible factors, the most practical and logical design is chosen for development.



Stage 3: Design

Stage-4: Developing Product

At this stage, the fundamental development of the product starts. For this, developers use a specific programming code as per the design in the DDS. Hence, it is important for the coders to follow the protocols set by the association. Conventional programming tools like compilers, interpreters, debuggers, etc. are also put into use at this stage. Some popular languages like C/C++, Python, Java, etc. are put into use as per the software regulations.



Stage 4: Development

Stage-5: Product Testing and Integration

After the development of the product, testing of the software is necessary to ensure its smooth execution. Although, minimal testing is conducted at every stage of SDLC. Therefore, at this stage, all the probable flaws are tracked, fixed, and retested. This ensures that the product confronts the quality requirements of SRS.

Documentation, Training, and Support: Software documentation is an essential part of the software development life cycle. A well-written document acts as a tool and means to information repository necessary to know about software processes, functions, and maintenance. Documentation also provides information about how to use the product. Training in an attempt to improve the current or future employee performance by increasing an employee's ability to work through learning, usually by changing his attitude and developing his skills and understanding.

Stage-5: Product Testing and Integration

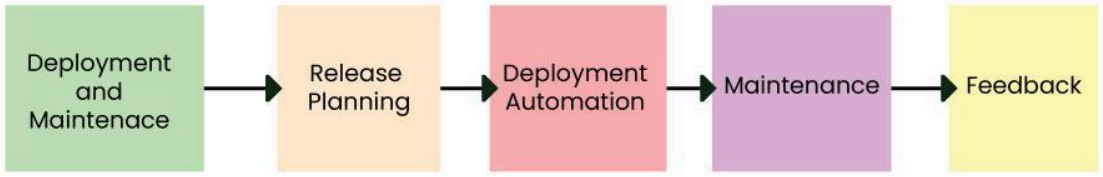
6 Stages of Software Development Life Cycle

Stage 5: Testing

Stage-6: Deployment and Maintenance of Products

After detailed testing, the conclusive product is released in phases as per the organization's strategy. Then it is tested in a real industrial environment. It is important to ensure its smooth performance. If it performs well, the organization sends out the product as a whole. After retrieving beneficial feedback, the company releases it as it is or with auxiliary improvements to make it further helpful for the customers. However, this alone is not enough. Therefore, along with the deployment, the product's supervision.

Stage 6: Deployment and Maintenance of Products



6 Stages of Software Development Life Cycle

Stage 6: Deployment and Maintenance

Automation Testing – Software Testing

Automated Testing means using special software for tasks that people usually do when checking and testing a software product. Nowadays, many software projects use automation testing from start to end, especially in agile and DevOps methods. This means the engineering team runs tests automatically with the help of software tools . It will help to keep the testing team to make the process faster. Continuous delivery (CD) and quickly sends the new code to users.

Automated testing is important for this because it converts the manual steps into automation. Continuous integration (CI) checks the new code changes to prevent issues. CD gets after CI done everything well. Automated testing, CI & CD will together prove that the new code is error-free and ready for deployment quickly for the project purpose.

What is Automation Testing?

On the other side, it's a technique where the Tester writes scripts independently and uses suitable Software or Automation Tools to test the software. It is an Automation Process of a Manual Process. It allows for executing repetitive tasks without the intervention of a Manual Tester.

- It is used to automate the testing tasks that are difficult to perform manually.
- Automation tests can be run at any time of the day as they use scripted sequences to examine the software.
- Automation tests can also enter test data compare the expected result with the actual result and generate detailed test reports.
- The goal of automation tests is to reduce the number of test cases to be executed manually but not to eliminate manual testing.
- It is possible to record the test suit and replay it when required.

Automated Testing uses specialized software to replace manual testing tasks, speeding up the process and integrating seamlessly with CI/CD pipelines. It allows for continuous code verification and quicker deployment.

What Kinds of Software Tests Should Be Automated First?

1. End-to-End tests

End-to-end testing is a type of software testing used to test whether the flow of software from the initial stage to the final stage is behaving as expected. The purpose of end-to-end testing is to identify system dependencies and to make sure that the data integrity is maintained between various system components and systems. End-to-end testing: End-to-end testing, also known as end-to-end functional testing, is a type of testing that validates the flow of a system from start to finish.

2. Unit tests

Unit testing is automated and is run each time the code is changed to ensure that new code does not break existing functionality. Unit tests are designed to validate the smallest possible unit of code, such as a function or a method, and test it in isolation from the rest of the system.

3. Integration tests

Integration testing is the process of testing the interface between two software units or modules. It focuses on determining the correctness of the interface. The purpose of integration testing is to expose faults in the interaction between integrated units. Once all the modules have been unit-tested, integration testing is performed.

4. Performance tests

Performance Testing is a type of software testing that ensures software applications perform properly under their expected workload. It is a testing technique carried out to determine system performance in terms of sensitivity, reactivity, and stability under a particular workload.

Parameters	Manual Testing	Automated Testing
Reliability	Manual testing is not accurate at all times due to human error, thus it is less reliable.	Since it is performed by third-party tools and/or scripts, therefore it is more reliable.
Investment	Heavy investment in human resources.	Investment in tools rather than human resources.

Time efficiency	Manual testing is time-consuming due to human intervention where test cases are generated manually.	Automation testing is time-saving as due to the use of the tools the execution is faster in comparison to manual testing.
Programming knowledge	There is no need to have programming knowledge to write the test cases.	It is important to have programming knowledge to write test cases.
Regression testing	There is a possibility that the test cases executed the first time will not be able to catch the regression bugs due to the frequently changing requirements.	When there are changes in the code, regression testing is done to catch the bugs due to changes in the code.

Integration Testing – Software Engineering

Last Updated : 26 Sep, 2024

Integration testing is the process of testing the interface between two software units or modules. It focuses on determining the correctness of the interface. The purpose of integration testing is to expose faults in the interaction between integrated units. Once all the modules have been unit-tested, integration testing is performed.

What is Integration Testing?

Integration testing is a software testing technique that focuses on verifying the interactions and data exchange between different components or modules of a software application. The goal of integration testing is to identify any problems or bugs that arise when different components are combined and interact with each other. Integration testing is typically performed after unit testing and before system testing. It helps to identify and resolve integration issues early in the development cycle, reducing the risk of more severe and costly problems later on.

Unit Testing – Software Testing

Unit Testing is a software testing technique in which individual units or components of a software application are tested in isolation. These units are the smallest pieces of code, typically functions or methods, ensuring they perform as expected.

Unit testing helps in identifying bugs early in the development cycle, enhancing code quality, and reducing the cost of fixing issues later. It is an essential part of **Test-Driven Development (TDD)**, promoting reliable code.

What is unit testing?

Unit testing is the process of testing the smallest parts of your code, like individual functions or methods, to make sure they work correctly. It's a key part of software development that improves code quality by testing each unit in isolation.

You write **unit tests** for these code units and run them automatically every time you make changes. If a test fails, it helps you quickly find and fix the issue. Unit testing promotes modular code, ensures better test coverage, and saves time by allowing developers to focus more on coding than manual testing.

What is a unit test?

A **unit test** is a small piece of code that checks if a specific function or method in is an application works correctly. It will work as the function inputs and verifying the outputs. These tests check that the code work as expected based on the logic the developer intended.

https://www.geeksforgeeks.org/top-8-software-development-models-used-in-industry/

1. Waterfall Model

Waterfall model is a famous and good version of SDLC(System Development Life Cycle) for software engineering. The waterfall model is a linear and sequential model, which means that a development phase cannot begin until the previous phase is completed. We cannot overlap phases in waterfall model.

We can imagine waterfall in the following way

> "Once the water starts flowing over the edge of the cliff, it starts falling down the mountain and the water cannot go back up."

Similarly waterfall model also works, once one phase of development is completed then we move to the next phase but cannot go back to the previous phase. In the waterfall model, the output of one phase serves as the input for the other phase.

Disadvantages of Waterfall Model

- In this model, complete and accurate requirements are expected at the beginning of the development process.
- Working software is not available for very long during the development life cycle.
- We cannot go back to the previous phase due to which it is very difficult to change the requirements.
- Risk is not assessed in this, hence there is high risk and uncertainty in this model.

2. V-Model

V-Model is an SDLC model, it is also called Verification and Validation Model. V-Model is widely used in the software development process, and it is considered a disciplined model. In V-Model, the execution of each process is sequential, that is, the new phase starts only after the previous phase ends.

- It is based on the association of testing phase with each development phase that is in V-Model with each development phase, its testing phase is also associated in a V-shape in other words both software development and testing activities take place at the same time.
- So in this model, Verification Phase will be on one side, Validation Phase will be on the other side that is both the activities run simultaneously and both of them are connected to each other in V-Shape through Coding Phase, hence it is called V-Model.
- **V-Design:** In V-Design the left side represents the development activity, the right side represents the testing activity.

3. Incremental Model

In Incremental Model, the software development process is divided into several increments and the same phases are followed in each increment. In simple language, under this model a complex project is developed in many modules or builds.

- For example, we collect the customer's requirements, now instead of making the entire software at once, we first take some requirements and based on them create a module or function of the software and deliver it to the customer. Then we take some more requirements and based on them add another module to that software.
- Similarly, modules are added to the software in each increment until the complete system is created. However, the requirements for making a complex project in multiple iterations/parts should be clear.

- If we understand the entire principle of Incremental methodology, then it starts by developing an initial implementation, then user feedback is taken on it, and it is developed through several versions until an accepted system is developed. Important functionalities of the software are developed in the initial iterations.

4. RAD Model

RAD model stands for rapid application development model. The methodology of RAD model is similar to that of incremental or waterfall model. It is used for small projects.

- If the project is large then it is divided into many small projects and these small projects are planned one by one and completed. In this way, by completing small projects, the large project gets ready quickly.
- In RAD model, the project is completed within the given time and all the requirements are collected before starting the project. It is very fast and there are very less errors in it.

5. Iterative Model

In Iterative model we start developing the software with some requirements and when it is developed, it is reviewed. If there are requirements for changes in it, then we develop a new version of the software based on those requirements. This process repeats itself many times until we get our final product.

- So, in Iterative model a software is developed by following several iterations. Iteration means that we are repeating the development process again and again. For example, we develop the first version of the software following the SDLC process with some software requirements.
- After the first version is developed, if there is a need to change the software , then a new version is developed with the second iteration. Now again we will see if the new version is enough, if not then we will make changes in it with the third iteration. The iteration will be repeated until the complete software is ready.
- The basic concept of Iterative model is that the software should be developed through repeated cycles or what we also call iteration and only a small part of it should be developed at a time. This model was developed to overcome the drawbacks of the classical waterfall model.

6. Spiral Model

Spiral model is a software development process model. This model has characteristics of both iterative and waterfall models. This model is used in projects which are large and complex. This model was named spiral because if we look at its figure, it looks like a spiral, in which a long curved line starts from the center point and makes many loops around it. The number of loops in the spiral is not decided in advance but it depends on the size of the project and the changing requirements of the user. We also call each loop of the spiral a phase of the software development process.

A software project goes through these loops again and again in iterations. After each iteration a more and more complete version of the software is developed. The most special thing about this model is that risks are identified in each phase and they are resolved through prototyping. This feature is also called Risk Handling.

7. Prototype model

Prototype model is an activity in which prototypes of software applications are created. First a prototype is created and then the final product is manufactured based on that prototype.

- The prototype model was developed to overcome the shortcomings of the waterfall model.
- This model is created when we do not know the requirements well.
- The specialty of this model is that this model can be used with other models as well as alone.

8. Agile Model

Agile model is a combination of iterative and incremental models, that is, it is made up of iterative and incremental models.

- In Agile model, focus is given to process adaptability and customer satisfaction.
- In earlier times, iterative waterfall model was used to create software. But in today's time developers have to face many problems. The biggest problem is that in the middle of software development, the customer asks to make changes in the software. It takes a lot of time and money to make these changes.
- In the agile model, the software product is divided into small incremental parts. In this, the smallest part is developed first and then the larger one.
- And each incremental part is developed over iteration.
- Each iteration is kept small so that it can be easily managed. And it can be completed in two-three weeks. Only one iteration is planned, developed and deployed at a time.

Agile has the following models

1. Scrum
2. Crystal methods
3. DSDM
4. Feature driven development (FDD
5. Lean software development
6. Extreme programming (xp)

https://www.geeksforgeeks.org/physical-components-of-computer-network/

1. NIC(Network Interface Card)

NIC or Network Interface Card is a network adapter used to connect the computer to the network. It is installed in the computer to establish a LAN. It has a unique ID that is written on the chip, and it has a connector to connect the cable to it. The cable acts as an interface between the computer and the router or modem. NIC card is a layer 2 device, which means it works on the network model's physical and data link layers.

Types of NIC

- **Wired NIC:** Cables and Connectors use Wired NIC to transfer data.
- **Wireless NIC:** These connect to a wireless network such as Wifi, Bluetooth, etc.

2. HUB

A hub is a multi-port repeater. A hub connects multiple wires coming from different branches, for example, the connector in star topology which connects different stations. Hubs cannot filter data, so data packets are sent to all connected devices. In other words, the collision domain of all hosts connected through hub remains one. Hub does not have any routing table to store the data of ports and map destination addresses., the routing table is used to send/broadcast information across all the ports.

Types of HUB

- **Active HUB:** Active HUB regenerates and amplifies the electric signal before sending them to all connected device. This hub is suitable to transmit data for long distance connections over the network.
- **Passive HUB:** As the name suggests it does not amplify or regenerate electric signal, it is the simplest types of Hub among all and it is not suitable for long-distnace connections.
- **Switching HUB:** This is also known as intelligent HUB, they provide some additional functionality over active and passive hubs. They analyze data packets and make decisions based on MAC address and they are operated on DLL(Data Link Layer).

3. Router

A Router is a device like a switch that routes data packets based on their IP addresses. The router is mainly a Network Layer device. Routers normally connect LANs and WANs and have a dynamically updating routing table based on which they make decisions on routing the data packets. The router divides the broadcast domains of hosts connected through it.

4. Modem

A Modem is a short form of Modulator/Demodulator. The Modem is a hardware component/device that can connect computers and other devices such as routers and switches to the internet. Modems convert or modulate the analog signals coming from telephone wire into a digital form that is in the form of 0s and 1s.

5. Switch

A Switch is a multiport bridge with a buffer and a design that can boost its efficiency(a large number of ports implies less traffic) and performance. A switch is a data link layer device. The switch can perform error checking before forwarding data, which makes it very efficient as it does not forward packets that have errors and forward good packets selectively to the correct port only.

7. Media

It is also known as Link which is going to carry data from one side to another side. This link can be Wired Medium (Guided Medium) and Wireless Medium (Unguided Medium). It is of two types:

7.1 Wired Medium

- **Ethernet:** Ethernet is the most widely used LAN technology, which is defined under IEEE standards 802.3. There are two types of Ethernet:
- **Fibre Optic Cable:** In fibre optic cable data is transferred in the form of light waves.
- **Coaxial Cable:** Coaxial Cable is mainly used for audio and video communications.
- **USB Cable:** USB Stands for Universal Serial Bus it is mainly used to connect PCs and smartphones.

8. Repeater

Repeater is an important component of computer networks as it is used to regenerate and amplify signal in the computer networks. Repeaters are used to improve the quality of the networks and they are operated on the Physical Layer of the OSI Model.

9. Server

A server is a computer program that provides various functionality to another computer program. The server plays a vital role in facilitating communication, data storage, etc. Servers have more data storage as compared to normal computers. They are designed for the specific purpose of handling multiple requests from clients.

Layer 1 – Physical Layer

The lowest layer of the OSI reference model is the **Physical Layer**. It is responsible for the actual physical connection between the devices. The physical layer contains information in the form of **bits.** Physical Layer is responsible for transmitting individual bits from one node to the next. When receiving data, this layer will get the signal received and convert it into 0s and 1s and send them to the Data Link layer, which will put the frame back together. Common physical layer devices are Hub, Repeater, Modem, and Cables.

Layer 2 – Data Link Layer (DLL)

The data link layer is responsible for the node-to-node delivery of the message. The main function of this layer is to make sure data transfer is error-free from one node to another, over the physical layer. When a packet arrives in a network, it is the responsibility of the DLL to transmit it to the Host using its MAC address. Packet in the Data Link layer is referred to as **Frame.** Switches and Bridges are common Data Link Layer devices.

The Data Link Layer is divided into two sublayers:

- Logical Link Control (LLC)
- Media Access Control (MAC)

The packet received from the Network layer is further divided into frames depending on the frame size of the **NIC(Network Interface Card)**. DLL also encapsulates Sender and Receiver's MAC address in the header.

The Receiver's MAC address is obtained by placing an ARP(Address Resolution Protocol) request onto the wire asking "Who has that IP address?" and the destination host will reply with its MAC address.

Layer 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 i.e. selection of the shortest path to transmit the packet, from the number of routes available. The sender and receiver's IP address are placed in the header by the network layer. Segment in the Network layer is referred to as **Packet.** Network layer is implemented by networking devices such as routers and switches.

Layer 4 – Transport Layer

The transport layer provides services to the application layer and takes services from the network layer. The data in the transport layer is referred to as **Segments**. It is responsible for the end-to-end delivery of the complete message. The transport layer also provides the

acknowledgment of the successful data transmission and re-transmits the data if an error is found. Protocols used in Transport Layer are TCP, UDP NetBIOS, PPTP.

At the sender's side, the transport layer receives the formatted data from the upper layers, performs **Segmentation**, and also implements **Flow and error control** to ensure proper data transmission. It also adds Source and Destination port number in its header and forwards the segmented data to the Network Layer.

- Generally, this destination port number is configured, either by default or manually. For example, when a web application requests a web server, it typically uses port number 80, because this is the default port assigned to web applications. Many applications have default ports assigned.

At the Receiver's side, Transport Layer reads the port number from its header and forwards the Data which it has received to the respective application. It also performs sequencing and reassembling of the segmented data.

Layer 5 – Session Layer

Session Layer in the OSI Model is responsible for the establishment of connections, management of connections, terminations of sessions between two devices. It also provides authentication and security. Protocols used in the Session Layer are NetBIOS, PPTP.

Layer 6 – Presentation Layer

The presentation layer is also called the **Translation layer**. The data from the application layer is extracted here and manipulated as per the required format to transmit over the network. Protocols used in the Presentation Layer are JPEG, MPEG, GIF, TLS/SSL, etc.

Layer 7 – Application Layer

At the very top of the OSI Reference Model stack of layers, we find the Application layer which is implemented by the network applications. These applications produce the data to be transferred over the network. This layer also serves as a window for the application services to access the network and for displaying the received information to the user. Protocols used in the Application layer are SMTP, FTP, DNS, etc.

Layer	Working	Protocol Data Unit	Protocols

1 – Physical Layer	Establishing Physical Connections between Devices.	Bits	USB, SONET/SDH, etc.
2 – Data Link Layer	Node to Node Delivery of Message.	Frames	Ethernet, PPP, etc.
3 – Network Layer	Transmission of data from one host to another, located in different networks.	Packets	IP, ICMP, IGMP, OSPF, etc.
4 – Transport Layer	Take Service from Network Layer and provide it to the Application Layer.	Segments (for TCP) or Datagrams (for UDP)	TCP, UDP, SCTP, etc.
5 – Session Layer	Establishes Connection, Maintenance, Ensures Authentication and Ensures security.	Data	NetBIOS, RPC, PPTP, etc.
6 – Presentation Layer	Data from the application layer is extracted and manipulated in the required format for transmission.	Data	TLS/SSL, MIME, JPEG, PNG, ASCII, etc.
7 – Application Layer	Helps in identifying the client and synchronizing communication.	Data	FTP, SMTP, DNS, DHCP, etc.

Simplex Mode

In simplex mode, Sender can send the data but the sender unable receive the data. It is a type of on way communication in which communication happens in only one direction. Example of this kind of mode is Keyboard, Traditional Monitors, etc.

Half-Duplex Mode

In half-duplex mode, Sender can send the data and also receive the data one sequentially. It is a bidirectional communication but limited to only one at a time. An example of this is the Walkie-Talkie, in which information is sent one at a time but in bi-directions.

Full Duplex Mode

In Full-duplex mode, Sender can send the data and also can receive the data simultaneously. It is dual way communication that is both way of communication happens at a same time. Example of this kind of transmission is Telephone Network, where communication happens parallel.

https://www.geeksforgeeks.org/binary-tree-data-structure/

https://www.geeksforgeeks.org/binary-search-tree-data-structure/

https://www.geeksforgeeks.org/introduction-to-heap/

DSU stands for **Disjoint Set Union**, also known as **Union-Find**. It is a data structure used to keep track of a set of elements partitioned into disjoint (non-overlapping) subsets. It is particularly useful in problems that involve grouping or connectivity queries, such as determining whether two elements are in the same subset or connecting two subsets.

Key Operations of DSU

1. **Find (Find-Set)**:
    - Determines which subset a particular element belongs to.
    - This can be used to check if two elements are in the same subset.
    - Optimized with **Path Compression**, which flattens the structure of the tree to make future queries faster.
2. **Union (Union-Set)**:
    - Merges two subsets into a single subset.
    - Optimized with **Union by Rank/Size**, which ensures the smaller or shorter tree is always added under the larger or taller tree to keep the structure efficient.

https://www.geeksforgeeks.org/graph-terminology-in-data-structure/

https://www.geeksforgeeks.org/binary-search/

https://www.geeksforgeeks.org/sorting-algorithms/

https://www.geeksforgeeks.org/greedy-algorithms/

https://www.geeksforgeeks.org/job-sequencing-problem/

https://www.geeksforgeeks.org/dynamic-programming/

https://www.geeksforgeeks.org/top-50-graph-coding-problems-for-interviews/

https://www.geeksforgeeks.org/backtracking-algorithms/

https://www.geeksforgeeks.org/convert-infix-expression-to-postfix-expression/

https://www.geeksforgeeks.org/postfix-to-infix/

https://www.geeksforgeeks.org/convert-infix-prefix-notation/

Polish notation (PN) is a mathematical notation where operators appear before their operands
https://www.geeksforgeeks.org/prefix-infix-conversion/
https://www.geeksforgeeks.org/hash-table-data-structure/
https://www.geeksforgeeks.org/what-is-hashing/
https://www.geeksforgeeks.org/different-types-of-queues-and-its-applications/