## The five-main phase of SDLC:

* Requirements gathering
* Design
* Development
* Testing
* Implementation & Maintenance

**SDLC Models**

* Waterfall Model
* Iterative Model
* Spiral Model
* V-Model
* Big Bang Model

**Waterfall Model**

The outcome of one phase acts as the input for the next phase sequentially.

Some situations where the use of Waterfall model is most appropriate are:

* Requirements are very well documented, clear and fixed.
* Product definition is stable.
* Technology is understood and is not dynamic.
* There are no ambiguous requirements.
* Ample resources with required expertise are available to support the product.
* The project is short.

Some of the major advantages of the Waterfall Model are as follows:

* Simple and easy to understand and use
* Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
* Phases are processed and completed one at a time.
* Works well for smaller projects where requirements are very well understood.
* Clearly defined stages.
* Well understood milestones.
* Easy to arrange tasks.
* Process and results are well documented.

The major disadvantages of the Waterfall Model are as follows −

* No working software is produced until late during the life cycle.
* High amounts of risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.
* Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
* It is difficult to measure progress within stages.
* Cannot accommodate changing requirements.
* Adjusting scope during the life cycle can end a project.
* Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

**Iterative Model**

Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).

This model is most often used in the following scenarios:

* Requirements of the complete system are clearly defined and understood.
* Major requirements must be defined; however, some functionalities or requested enhancements may evolve with time.
* There is a time to the market constraint.
* A new technology is being used and is being learnt by the development team while working on the project.
* Resources with needed skill sets are not available and are planned to be used on contract basis for specific iterations.
* There are some high-risk features and goals which may change in the future.

The advantages of the Iterative and Incremental SDLC Model are as follows:

* Some working functionality can be developed quickly and early in the life cycle.
* Results are obtained early and periodically.
* Parallel development can be planned.
* Progress can be measured.
* Less costly to change the scope/requirements.
* Testing and debugging during smaller iteration is easy.
* Risks are identified and resolved during iteration; and each iteration is an easily managed milestone.
* Easier to manage risk - High risk part is done first.
* With every increment, operational product is delivered.
* Issues, challenges and risks identified from each increment can be utilized/applied to the next increment.
* Risk analysis is better.
* It supports changing requirements.
* Initial Operating time is less.
* Better suited for large and mission-critical projects.
* During the life cycle, software is produced early which facilitates customer evaluation and feedback.

The disadvantages of the Iterative and Incremental SDLC Model are as follows:

* More resources may be required.
* Although cost of change is lesser, but it is not very suitable for changing requirements.
* More management attention is required.
* System architecture or design issues may arise because not all requirements are gathered in the beginning of the entire life cycle.
* Defining increments may require definition of the complete system.
* Not suitable for smaller projects.
* Management complexity is more.
* End of project may not be known which is a risk.
* Highly skilled resources are required for risk analysis.
* Projects progress is highly dependent upon the risk analysis phase.

## Spiral Model

The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

The following pointers explain the typical uses of a Spiral Model:

* When there is a budget constraint and risk evaluation is important.
* For medium to high-risk projects.
* Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
* Customer is not sure of their requirements which is usually the case.
* Requirements are complex and need evaluation to get clarity.
* New product line which should be released in phases to get enough customer feedback.
* Significant changes are expected in the product during the development cycle.

The advantages of the Spiral SDLC Model are as follows:

* Changing requirements can be accommodated.
* Allows extensive use of prototypes.
* Requirements can be captured more accurately.
* Users see the system early.
* Development can be divided into smaller parts and the risky parts can be developed earlier which helps in better risk management.

The disadvantages of the Spiral SDLC Model are as follows:

* Management is more complex.
* End of the project may not be known early.
* Not suitable for small or low risk projects and could be expensive for small projects.
* Process is complex
* Spiral may go on indefinitely.
* Large number of intermediate stages requires excessive documentation.

## Agile Model

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

The advantages of the Agile Model are as follows:

* Is a very realistic approach to software development.
* Promotes teamwork and cross training.
* Functionality can be developed rapidly and demonstrated.
* Resource requirements are minimum.
* Suitable for fixed or changing requirements
* Delivers early partial working solutions.
* Good model for environments that change steadily.
* Minimal rules, documentation easily employed.
* Enables concurrent development and delivery within an overall planned context.
* Little or no planning required.
* Easy to manage.
* Gives flexibility to developers.

The disadvantages of the Agile Model are as follows:

* Not suitable for handling complex dependencies.
* More risk of sustainability, maintainability and extensibility.
* An overall plan, an agile leader and agile PM practice is a must without which it will not work.
* Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
* Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
* There is a very high individual dependency, since there is minimum documentation generated.
* Transfer of technology to new team members may be quite challenging due to lack of documentation.

# **SOAP**

* SOAP is a communication protocol designed to communicate via Internet.
* SOAP can extend HTTP for XML messaging.
* SOAP provides data transport for Web services.
* SOAP can exchange complete documents or call a remote procedure.
* SOAP can be used for broadcasting a message.
* SOAP is platform- and language-independent.
* SOAP is the XML way of defining what information is sent and how.
* SOAP enables client applications to easily connect to remote services and invoke remote methods.

# **REST FULL API**

* it should be stateless
* It should access all the resources from the server using only URI
* It does not have inbuilt encryption
* It does not have session
* It uses one and only one protocol - HTTP
* For performing CRUD operations, it should use HTTP verbs such as get, post, put and delete
* It should return the result only in the form of JSON or XML, atom, OData etc. (lightweight data )

REST based services follow some of the above principles and not all

RESTFUL services means it follows all the above principles.

Some other notes on GET requests:

* GET requests can be cached
* GET requests remain in the browser history
* GET requests can be bookmarked
* GET requests should never be used when dealing with sensitive data
* GET requests have length restrictions
* GET requests are only used to request data (not modify)

Some other notes on POST requests:

* POST requests are never cached
* POST requests do not remain in the browser history
* POST requests cannot be bookmarked
* POST requests have no restrictions on data length

**PHP**

What is the actually used PHP version?

* Version 7.1 or 7.2 is the recommended version of PHP.

What is the difference between "echo" and "print" in PHP?

* Echo can output one or more string but print can only output one string and always returns 1.
* Echo is faster than print because it does not return any value.

What is the main difference between PHP 4 and PHP 5?

* PHP 5 presents many additional OOP (Object Oriented Programming) features.

Is multiple inheritance supported in PHP?

* PHP supports only single inheritance; it means that a class can be extended from only one single class using the keyword 'extended'.

What is needed to be able to use image function?

* GD library is needed to execute image functions.

What are the ways to define a constant in PHP?

* PHP constants are name or identifier that can't be changed during execution of the script. PHP constants are defined in two ways:
  + Using define() function
  + Using const() function

What does isset() function?

* The isset() function checks if the variable is defined and not null.

How many types of array are there in PHP?

* There are three types of array in PHP:

1. Indexed array: an array with a numeric key.
2. Associative array: an array where each key has its specific value.
3. Multidimensional array: an array containing one or more arrays within itself.

What is the difference between indexed and associative array?

* The indexed array holds elements in an indexed form which is represented by number starting from 0 and incremented by 1. For example:

$season=array("summer","winter","spring","autumn");

* The associative array holds elements with name. For example:

$salary=array("Sonoo"=>"350000","John"=>"450000","Kartik"=>"200000");

What are the methods to submit form in PHP?

* There are two methods GET and POST.

How can you submit a form without a submit button?

* You can use JavaScript submit() function to submit the form without explicitly clicking any submit button.

What are the ways to include file in PHP?

* PHP allows you to include file so that page content can be reused again. There are two ways to add the file in PHP.
  + include
  + require

Differentiate between require and include?

* Require and include both are used to include a file, but if data is not found include sends warning whereas require sends Fatal error.

How can you retrieve a cookie value?

* echo $\_COOKIE ["user"];

What is a session?

* PHP Engine creates a logical object to preserve data across subsequent HTTP requests, which is known as session.
* Sessions generally store temporary data to allow multiple PHP pages to offer a complete functional transaction for the same user.
* Simply, it maintains data of an user (browser).

What is the method to register a variable into a session?

<?php

Session\_register($ur\_session\_var);

?>

What is $\_SESSION in PHP?

* A session creates a file in a temporary directory on the server where registered session variables and their session id are stored. This data will be available to all pages on the site amid that visit.
* The area of the temporary record is controlled by a setting in the php.ini document called session.save\_path.
* At the point when a session is begun following things happen:
* PHP first makes two duplicates of one of a kind session id for that particular session of the client which is an arbitrary string of 32 hexadecimal numbers, for example, 3c7foj34c3jjhkyepop2fc937e3443.
* One copy of unique session id automatically sent to the user?s computer for the sake of synchronization in future ahead, and one copy is being maintained at server side till the session is running.
* Whenever you want to access the page of website or web app, then session id of the current user will be associated with the HTTP header, and that will be compared by the session id which is being maintained at the server. After completing the comparison process, you can easily access the page of the website or web app
* A session ends when the user closes the browser, or after leaving the site, the server will terminate the session after a predetermined period, commonly 30 minutes duration.

What is PHP session\_start() and session\_destroy() function?

* PHP session\_start() function is used to start the session. It starts new or resumes the current session. It returns the current session if the session is created already. If the session is not available, it creates and returns new sessions.

What is the difference between session and cookie?

* The main difference between session and cookie is that cookies are stored on user's computer in the text file format while sessions are stored on the server side.
* Cookies can't hold multiple variables, on the other hand, Session can hold multiple variables.
* You can manually set an expiry for a cookie, while session only remains active as long as browser is open.

| **GET** | **POST** |
| --- | --- |
| GET method is used for requesting data from a specified resource. | POST is used for sending the data to the server as a package in a separate communication with the processing script. |
| Data is sent in the form of URL parameters which are strings of name-value pairs separated by ampersands(&) | Data sent through the POST method will not be seen in the URL |
| GET method cannot be used for sending binary data like images or word documents | The POST method can be used to send ASCII as well as binary data like images and word documents |
| This method must not be used if you have any sensitive information like a password to be sent to the server. | Sensitive information can be sent using this method. |
| It can be used for submitting the form where the user can bookmark the result. | Submissions by form with POST cannot be bookmarked. |
| You can use this method only for data that is not secure. | Data sent through this method is secure. |
| GET method is not safer since parameters may be stored in web server logs or browser history. | POST method is safer than GET because the parameters are not stored in web server logs or browser history. |