**Software Development Life Cycle**

1. Explain SDLC at a high level?

The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software. The following phases of software development-

* Requirement analysis
* Planning
* Software design such as architectural design
* Software development
* Testing
* Deployment

1. What is waterfall and why it is still relevant?

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. The most prevalent development methodology, it is not the greatest way to go about designing complicated applications. This strategy frequently leaves the customer with little state about upgrades and changes once improvement has begun.

1. Explain Agile Model with a use case and the role of SCRUM in that?

Agile SDLC model is a blend of iterative and gradual procedure models with centre around process flexibility and consumer loyalty by quick conveyance of working programming item. Scrum is a subset of Agile. It is a lightweight procedure structure for quick turn of events, and the most broadly utilized one.

1. Who is Scrum Master?

A scrum master is the facilitator for a spry advancement group. Scrum is a philosophy that permits a group to self-sort out and make changes rapidly, as per coordinated standards. The scrum ace deals with the procedure for how data is traded.

1. What is Epic & Story?

An epic is a large body of work that can be broken down into a number of smaller stories, or sometimes called “Issues” in Jira. Epics often encompass multiple teams, on multiple projects, and can even be tracked on multiple boards. Epics are almost always delivered over a set of sprints.

1. What is called Velocity in SCRUM?

Velocity is a measure of the amount of work a Team can tackle during a single Sprint and is the key metric in Scrum. Velocity is calculated at the end of the Sprint by totalling the Points for all fully completed User Stories.

1. Explain the SCRUM ceremonies?

Scrum ceremonies and artifacts are aimed to support the Scrum team in delivering on its objectives. All these events and activities are important to Agile software development teams and should be fulfilled to ensure team performance and effectiveness are at their peak.

1. What is grooming?

Grooming is a telephone term that typically refers to managing high-capacity lines between central offices, carriers, ISPs and very large corporations rather than subscriber lines (local loops).

1. Differentiate between SCRUM & Waterfall?

Scrum- 1. Work is divided in tea

2. It is save the time.

3. its process works well for difficult and complex projects.

4. it has no define stages.

Waterfall – 1. Work is divided into phases.

2. It may take extra time as reviewing.

3. Its process works well for smaller projects.

4. It has clear and defined stages.

1. Explain the responsibilities of Product Owner?

A Scrum Product Owner is responsible for maximizing the value of the product resulting from the work of the Development Team. ... Ordering the items in the Product Backlog to best achieve goals and missions.

**Client-Server**

1. Tell us about the features of client/server.

A **client-server** network is designed for end-users, called clients, to access resources such as files, songs, video collections, or some other service from a central computer called a server. A server’s sole purpose is to do what its name implies – serve its clients! You may have been using this configuration and not even have known it. There are different feathers of Client/server architecture and these are -

All data are centralized on a single server, simplifying security checks and updates data and software. The technology supporting the client/server are more mature than others.

1. What is a Web server in a client-server environment?  
   Web server is software or hardware that uses HTTP (Hypertext Transfer Protocol) and other protocols to respond to client requests made over the World Wide Web (WWW). Web server software controls how a user accesses hosted files. Web server relays in the presentation layer (Tier 1).
2. What is the role of the presentation layer  
   The presentation layer is responsible for the delivery and formatting of information to the application layer for further processing or display. This type of service is needed because different computer architectures use different data representations. The presentation layer handles all issues related to data presentation and transport, including translation, encryption, and compression.
3. They say this architecture is secure, how is it done in your opinion?  
   The client-server architecture is secure as it starts securing from the first tier which is presentation tier. And in the middle of each tier, there is a firewall system which helps to secure the connection and also with the input that comes from the client. The Backups and network security are controlled centrally throughout the presentation and application layer.
4. What is a Database Server in a client-server environment?  
   A database server is a server which uses a database application that provides database services to other computer programs or to computers, as defined by the client-server model. It basically a response to the client request in presentation layer throughout the application layer. It stores all the databases.
5. Explain 2-Tier and 3-Tier architecture?  
   A two-tier architecture is a database architecture where the presentation layer runs on a client and data is stored on a Server. The 3-tier architecture is consisting of the Presentation layer (PC, Tablet, Mobile, etc.), the Application layer (server) and Database Server.
6. What is a File server?  
   It shares files and folders, storage space to hold files and folders, or both, over a network.

**SOA & Microservices**

1. What are the main benefits of SOA?

The main benefits of SOA are below-

* Services are reusable.
* Services are easily maintained.
* Reliability and availability.
* Platform Independence.

1. How can you achieve loose coupling in SOA?

S**ervice loose coupling** is a design principle in SOA. Anything that connects has coupling.  **Loose Coupling** means one class is dependent on interface rather than class. To achieve it, there should be less dependencies. One strategy for achieving loose coupling is to use the service interface (the WSDL for a SOAP Web Service) to limit this dependency, hiding the service implementation from the consumer.

1. Are web services and SOA the same?

Web services are a popular form of SOA implementation. Service Oriented Architecture, as the name says is an architectural concept which focuses on having different services communicating with each other to carry out a bigger job. A web service is a basic building block in a SOA.

1. What is a reusable service?

The principle of Service Reusability emphasizes the positioning of services as enterprise resources with agnostic functional contexts. In other words, Reusable services are applied within the service-orientation design paradigm, to create services that can be reused across a business.

1. What are the disadvantages of SOA?

* SOA is costly in terms of human resource, development, and technology.
* In SOA, all inputs are validated before it is sent to the service. If multiple services are being used, then it will overload system with extra computation.
* It needs a high-speed server with a lot of data bandwidth to run a web service.

1. What is ESB and where does it fit in?

An Enterprise service Bus (ESB for short) refers to software architecture that allows for the integration of enterprise applications and services, such as middleware infrastructure platforms*.* ESB is an open standards-based distributed messaging middleware that provides secure interoperability between enterprise applications via XML, Web services interfaces and standardized rules-based routing of documents.

1. In SOA do we need to build a system from scratch?

According to my understanding, No. if there is a need to integrate or make an existing system as a business service, then just need to create loosely coupled wrappers which will wrap custom systems and expose the systems functionality in a generic fashion to the external world. Each service in an SOA embodies the code and data integrations required to execute a complete, discrete business function (e.g., checking a customer’s credit, calculating a monthly loan payment, or processing a mortgage application). The service interfaces provide loose coupling, meaning they can be called with little or no knowledge of how the integration is implemented underneath. The services are exposed using standard network protocols—such as SOAP (simple object access protocol)/HTTP or JSON/HTTP—to send requests to read or change data.

1. What is the most important skill needed to adopt SOA? technical or cultural?

Cultural is most important skill. SOA does require people to think of business and technology differently. While implementing SOA, experts will first think in terms of business functions, or services instead of thinking of technology first.

1. List down the advantages of Microservices Architecture.

* In microservices architecture, large software projects are broken down into smaller, more independent modules.
* Code for different services can be written in different languages.
* Easy integration and automatic deployment; using open-source continuous  tools such as Jenkins.

1. What are the best practices to design Microservices?

The best practices to design Microservices are Use Distributed Configuration, continuous delivery, monitoring everything, using API gateways to aggregate Data into Specific clients.

1. How does Microservice Architecture work?

The Microservices work by splitting its business components in small services that can be deployed and operated independently from each other.

1. What are the pros and cons of Microservice Architecture?

It has Greater agility, faster time to market, better scalability, faster development cycles, platform and language agnostic services.

With pros, there are some cons too. Harder to test monitor because of the complexity of the architecture. Harder to maintain the network. Security issues (harder to maintain transaction safety, distributed communication goes wrong more likely, etc.).

1. What is the difference between SOA and Microservices Architecture?

The main difference between SOA and Microservices Architecture is that SOA designed to share resources across services while   
Designed to host services which can function independently.

1. What are the challenges you face while working Microservice Architectures?

The challenges are big while working in Microservice Architectures. The first one is testing is much more complex due to different services. The second big challenge is fault tolerance. Individuals services do not bring down the overall system. With the complexity, microservices need to be able to withstand internal and external failures. Another challenge is monitoring the services. The traditional forms of monitoring and diagnostics will not align well with microservices architecture.

1. What are the characteristics of Microservices?

Microservices are small, each running in their own process, using lightweight communication mechanisms and built around business capabilities. Microservices use external user interface and there is a possible communication between units.

**HTTP Fundamentals**

1. What are the Basic Features of HTTP?

It is the protocol that allows web servers and browsers to exchange data over the web. It is a request-response protocol. It uses the reliable TCP connections by default on TCP port 80. It is stateless means each request is considered as the new request.

1. What are request methods in HTTP?

GET, POST, PUT, DELETE, HEAD, OPTIONS, CONNECT.

1. What are the differences between GET and POST methods?

GET is used to request data from a specified resource. GET is one of the most common HTTP methods Whereas POST is used to send data to a server to create/update a resource.

1. What is status code in HTTP?

HTTP response status codes indicate whether a specific HTTP request has been successfully completed. Responses are grouped into five classes: Informational responses (100–199), Successful responses (200–299), Redirects (300–399), Client errors (400–499), and Server errors (500–599).

1. What are the header fields in HTTP?

HTTP header fields provide required information about the request or response, or about the object sent in the message body. There are four types of HTTP message headers: General header: These header fields have general applicability for both request and response messages.

Client Request-header: These header fields have applicability only for request messages.

Server Response-header: These header fields have applicability only for response messages.

Entity-header: These header fields define meta-information about the entity-body or, if nobody is present, about the resource identified by the request.

1. What is URI?

URI stands for Uniform Resource Identifier, and it's the official name for those things you see all the time on the Web that begin 'http:' or 'mailto:'

1. What are Idempotent methods and why do we call them?

An idempotent HTTP method is an HTTP method that can be called many times without different outcomes. It would not matter if the method is called only once, or ten times over. The result should be the same. Of the request methods defined by this specification, PUT, DELETE, and safe request methods are idempotent.

1. Explain HTTP Request & Response Messages.

HTTP requests are messages sent by the client to initiate an action on the server. The response contains status information about the request and may also contain the requested content. After receiving and interpreting a request message, a server responds with an HTTP response message.

1. What is Session State in HTTP?

Session state is a method keep track of the user session during a series of HTTP requests. Session state allows a developer to store data about a user as he/she navigates through web pages in web application.

1. What is HTTPS?

HTTPS stands for Hypertext Transfer Protocol Secure. It is the protocol where encrypted HTTP data is transferred over a secure connection.

**Introduction to API**

1. Explain REST and RESTFUL?

REST stands for representational state transfer. That means that the state itself is not transferred but a mere representation of it is. RES is basically an architectural style of development which follows some rules- It should be stateless, It should access all the resources from the server using only URI, It does not have a session, and it uses one and only one protocol - HTTP.

1. Mention what are the HTTP methods supported by REST?

Get, post, put and delete and head.

1. Explain the architectural style for creating web API?

The Web API architecture style is a hybrid style derived from REST, RPC styles (Remote Procedure Call) and SOAP style.

1. Explain the RESTFul Web Service?

Restful Web Service is a lightweight, maintainable, and scalable service that is built on the REST architecture. Just like SOAP (Simple Object Access Protocol), which is used to develop web services by the XML method, RESTful web services use web protocol i.e. HTTP protocol method. RESTful web service implementation defines the method of accessing various resources that are required by the client and he has sent the request to the server through the web browser.

1. Explain what is a “Resource” in REST?

A resource is an object with a type, associated data, relationships to other resources, and a set of methods that operate on it. It is similar to an object instance in an object-oriented programming language, with the important difference that only a few standard methods are defined for the resource (corresponding to the standard HTTP GET, POST, PUT and DELETE methods).

1. Which protocol is used by RESTful web services?

HTTP

1. What is messaging in RESTful web services?

RESTful Web Services make use of HTTP protocols as a medium of communication between client and server. A client sends a message in the form of an HTTP Request and the server responds in the form of an HTTP Response. This technique is termed as Messaging. These messages contain message data and metadata i.e. information about the message itself.

1. State the core components of an HTTP Request?

Verb: Includes methods like GET, PUT, POST, etc., Uniform Resource Identifier for identifying the resources available on the server. HTTP Version for specifying the HTTP version. HTTP Request header contains metadata for the HTTP Request message as key-value pairs. For example, client (or browser) type, format supported by client, the format of the message body. HTTP Request body that contains the representation of the resources or Message content in use.

1. State the core components of an HTTP response?

Request Code: This contains various codes which determine the status of the server response. HTTP Version for specifying the HTTP version. HTTP Response header for containing the information about the data. HTTP Response body that contains the representation of the resources in use.

1. What do you understand about payload in RESTFul web service?

Payload refers to JSON-formatted text that is either posted (via an HTTP POST) to a web service when a user creates a resource or returned from a web service (via an HTTP GET) when a user requests a resource (or resources).

1. Explain the caching mechanism?

When a caching mechanism is in place, it helps improve delivery speed by storing a copy of the asset you requested and later accessing the cached copy instead of the original.

1. List the main differences between SOAP and REST?

The main difference b/w SOAP and REST is that SOAP has high security, less flexibility, more complexity and low performance. However, in REST, it supports better performance, more flexibility, more performance but less security.

1. Enlist advantages and disadvantages of ‘Statelessness’.

Advantage of stateless applications is that the maintaining parties don't have the responsibility of managing the resident memory. Stateless applications can be less costly than a similar stateful application. They can also be less complex, as there’s no requirement to take in data and keep it on hand, to record it for use later. Every session is brand new and the same programming occurs.

There are some disadvantages too and these are- they will not keep information about a user session. This issue is evident in the design of cookies to handle user session information on the internet.

**Object Oriented Programming Fundamentals**

1. What is the main difference between a class and an object?

An object is a member or an "instance" of a class. A class defines object properties including a valid range of values, and a default value. A class also describes object behaviour.

1. What is Encapsulation? Explain with a used case.

In object-oriented computer programming languages, the encapsulation refers to the bundling of data, along with the methods that operate on that data, into a single unit. For example, in Student.java-

package com. javatpoint;

public class Student {

private String name;

public String getName (){

return name;

}

public void setName (String name){

this.name=name

}

}

And in Test.java-

package com.javatpoint;

class Test{

public static void main(String[] args){

//creating instance of the encapsulated class

Student s=new Student();

//setting value in the name member

s.setName("Dil");

//getting value of the name member

System.out.println(s.getName());

}

}

1. What is Polymorphism? Explain with a used case

Polymorphism is the ability of an object to take on many forms. The most common use of polymorphism in OOP occurs when a parent class reference is used to refer to a child class object.

For example-

public interface Vegetarian{}

public class Animal{}

public class Rabbit extends Animal implements Vegetarian{}.

Now, the Rabbit class is polymorphic since this has multiple inheritance.

1. Explain Overriding & Overloading and its advantages.

Overridingmeans having two methods with the same method name and parameters (i.e., method signature). One of the methods is in the parent class and the other is in the child class. Overriding allows a child class to provide a specific implementation of a method that is already provided its parent class. Whereas, Overloading occurs when two or more methods in one class have the same method name but different parameters.

1. What is Inheritance and different types of inheritance? Explain with a used case.

**Inheritance** is a mechanism in which one class acquires the property of another class. For example, a child inherits the traits of his/her parents. There are three types of inheritance- **Single inheritance** where one class extends another class (one class only). **Multiple Inheritance where** one class extending more than one class. Java does not support multiple inheritance. At last, Multilevel Inheritance where one class can inherit from a derived class. Hence, the derived class becomes the base class for the new class.

1. What is an abstract class?

An abstract class is a template definition of methods and [variables](https://whatis.techtarget.com/definition/variable) of a [class](https://whatis.techtarget.com/definition/class) (category of [objects](https://searchapparchitecture.techtarget.com/definition/object)) that contains one or more [abstracted](https://whatis.techtarget.com/definition/abstraction) methods. **An abstract class can declare both abstract and concrete methods** Abstract classes are used in all object-oriented programming ([OOP](https://searchapparchitecture.techtarget.com/definition/object-oriented-programming-OOP)) languages.

1. What is an interface and how multiple inheritance is achieved with this?

Interface is a type, just as a class is a type. Like a class, an interface defines methods. Unlike a class, an interface never implements methods; instead, classes that implement the interface implement the methods defined by the interface. A class can implement multiple interfaces. In Multiple inheritance, one class can have more than one superclass and inherit features from all its parent classes. Multiple inheritance by interface occurs if a class implements multiple interfaces or also if an interface itself extends multiple interfaces.

1. What are the access modifiers?

Access modifiers are keywords in object-oriented languages like java that set the accessibility of classes, methods, and other members. Access modifiers are a specific part of programming language syntax used to facilitate the encapsulation of components. For example, in Java, protector, public, private and package are the modifier.

1. What are the various types of constructors?

There are different types of constructor and these are -

* Default constructed
* Parameterized constructor
* Copy constructor
* Conversion constructor
* Explicit constructor

1. What is ‘this’ pointer?

‘This’ pointer is a pointer accessible only within the non static member functions of a**class, struct,** or**union** type. It points to the object for which the member function is called. Static member functions don't have this pointer.

1. What is static and dynamic Binding?

Static Binding- The binding which can be resolved at compile time by compiler is known as static or early binding. Binding of all the static, private and final methods is done at compile-time

Dynamic Binding -In Dynamic binding compiler doesn’t decide the method to be called. Overriding is a perfect example of dynamic binding. In overriding both parent and child classes have same method.

1. How many instances can be created for an abstract class and why?

Zero instance because it is not possible to create any object from abstract class

1. Which OOPS concept is used as a reuse mechanism and explain with a use case

Inheritance is the OOPS concept can be used as reuse mechanism.

Package com.Java.inheritrance;

class SuperClassA {

public void off(){

System.out.println(“SuperClassA”)

class SubClassB extends SuperClass{

public void onn(){

System.out.println(“SubClassB”)

}

}

public class Test {

public static void main(String args[]){

SubClassB a = new SubClassB();

a.off();

a.onn();

}

}

**Unit Testing & JUnit**

1. What is unit testing?

UNIT TESTING is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

1. What is the difference between manual testing and automated testing?

Automation Testing uses automation tools to execute test cases whereas in manual testing, test cases are executed by a human tester and software.

1. Is it necessary to write the test case for every logic? If yes, why

The answer is yes, you should unit test everything you can. Doing so creates a legacy for later so changes down the road can be done with peace of mind. It ensures that your code works as expected.

1. What are the features of JUnit?

* JUnit is an open source framework, which is used for writing and running tests.
* Provides annotations to identify test methods.
* Provides assertions for testing expected results.
* Provides test runners for running tests.
* JUnit tests allow you to write codes faster, which increases quality.

1. What are the important JUnit annotations? And its usage in coding

The important JUNIT annotations are following

* @BeforeClass – Run once before any of the test methods in the class, public static void
* @AfterClass – Run once after all the tests in the class have been run, public static void
* @Before – Run before @Test, public void
* @After – Run after @Test, public void
* @Test – This is the test method to run, public void

1. What does Assert class?

Assert class is useful in determining Pass or Fail status of a test case, The assert methods are provided by the class org.junit.Assert which extends java.lang.Object class.

1. What is Code Coverage?

Code coverage is a term used in software testing to describe how much program source code is covered by a testing plan. Developers look at the number of program subroutines and lines of code that are covered by a set of testing resources and techniques.

1. What are the best practices to perform Unit Testing?

* Unit Tests Should Be Maintainable and Readable.
* Unit Tests Should Verify a Single-Use Case.
* Unit Tests Should Be Isolated.
* Unit Tests Should Be Automated

1. What is Mocking?

Mocking is primarily used in unit testing. An object under test may have dependencies on other (complex) objects. To isolate the behaviour of the object you want to replace the other objects by mocks that simulate the behaviour of the real objects. This is useful if the real objects are impractical to incorporate into the unit test.

**GIT**

1. What is GIT and its significance in SDLC.

Git is a version control system for tracking changes in files and coordinating work on those files among multiple people. It is primarily used for source code management in software development. It is a distributed revision control system and is very useful to support software development workflows.

1. What is the difference between GIT and SVN?

The difference between Git and SVN version control systems is that Git is a distributed version control system, whereas SVN is a centralized version control system. Git uses multiple repositories including a centralized repository and server, as well as some local repositories

1. What are the advantages of using GIT?

* Most of the Git operations are fast, mainly because they are performed on local repository copy.
* Adding and moving files.
* It is possible to check the status of changes.
* It is possible to ignore certain files in the local repository directories using a file named. gitignore.

1. What is “Staging Area” or “Index” in GIT?

 ”Index” and “stage” is "cache", as in git diff --cached. It's all the same: it's the proto-commit that will turn into a commit when run git commit and supply a commit message.

1. What is GIT stash?

Use git stash when you want to record the current state of the working directory and the index but want to go back to a clean working directory. The command saves local modifications away and reverts the working directory to match the HEAD commit.

1. What is the function of git clone?

git clone is primarily used to point to an existing repository and make a clone or **copy** of that repository at in a new directory, at another location.

1. How can you create a repository in Git?

* Create a directory to contain the project.
* Go into the new directory.
* Type git init.
* Write some code.
* Type git add to add the files
* Type git commit.

1. What is the difference between ‘git remote’ and ‘git clone’?

Git clone is used to copy or clone a different repository whereas git remote is used to refer to a remote repository or your central repository.

1. What is the function of ‘git diff ’ in git?

**Diff command is used in git to track the difference between the changes made on a file.**

1. Explain what the commit message is?

A git commit is a change (or “patch”) to code. A commit message is attached to that change — not the code itself. Accordingly, when you write a commit message you are writing it as if it's about to be applied, rather than about what you just did.

1. Why is it advisable to create an additional commit rather than amending an existing commit?

Because the amend operation will destroy the state that was previously saved in a commit. If it’s just the commit message being changed then that’s not an issue.

1. What is Rebasing?

In Git, the rebase command integrates changes from one branch into another. It is an alternative to the better known "merge" command.

**Maven Fundamentals**

1. Explain what is Maven? How does it work?

Maven is a powerful project management tool that is based on POM (project object model). It is used for projects build, dependency and documentation.

1. Explain what POM and its significance is?

A Project Object Model or POM is the fundamental unit of work in Maven. It is an XML file that contains information about the project and configuration details used by Maven to build the project.

1. Explain what a Maven artefact is?

A Maven Artefact is a Java class that represents the kind of "name" that gets dereferenced by a repository manager into a repository manager artefact.

1. List out the dependency scope in Maven?

**Maven dependency scope** attribute is used to specify the visibility of a dependency, relative to the different lifecycle phases (build, test, runtime etc). Maven provides six scopes i.e. compile, providing and runtime.

1. List out what are the build phases in Maven?

* validate check if all information necessary for the build is available
* compile: compile the source code
* test-compile: compile the test source code
* test: run unit tests
* package: package compiled source code into the distributable format (jar, war, …)
* integration-test: process and deploy the package if needed to run integration tests
* install install the package to a local repository
* deploy copy the package to the remote repository.

1. Mention the three-build lifecycle of Maven?

There are three built-in build lifecycles: default, clean and site. The default lifecycle handles your project deployment, the clean lifecycle handles project cleaning, while the site lifecycle handles the creation of your project's site documentation.

1. List out what are the aspects does Maven Manages?

* Build.
* Documentation.
* Reporting.
* Dependencies.
* SCMs.
* Releases.
* Distribution.
* Mailing **list**.

1. Explain what a Maven Repository is? What are their types?

A repository is a directory where all the project jars, library jar, plugins or any other project specific artefacts are stored and can be used by Maven easily. There are three types- local, central, remote.

1. Explain how you can exclude dependency?

There we can exclude all transitive dependencies without specifying groupId and artifactId of the dependencies. So need to use astric(\*) character as groupid and artifactid of the dependency.

1. For POM what are the minimum required elements?

The minimum requirement for a POM are the following:

* project root
* modelVersion - should be set to 4.0.0
* groupId - the id of the project's group.
* artifactId - the id of the artifact (project)
* version - the version of the artifact under the specified group

**CI/CD**

1. What are the fundamental differences between DevOps & Agile?

Agile refers to an iterative approach which focuses on collaboration, customer feedback, and small, rapid releases whereas DevOps is considered a practice of bringing development and operations teams together.

Agile helps to manage complex projects whereas DevOps central concept is to manage end-to-end engineering processes.

1. What is the need for DevOps?

DevOps describes a culture and set of processes that bring development and operations teams together to complete software development. It allows organizations to create and improve products at a faster pace than they can with traditional software development approaches.

1. What are the advantages of DevOps?

Faster delivery of features. More stable operating environments. Improved communication and collaboration.

1. What are the success factors for Continuous Integration?

The success Factors for Continuous Integration are the right stuff, communicate results, testing the application, proving requirements, enforcing architecture.

1. What are the differences between continuous integration, continuous delivery, and continuous deployment?

**In Continuous Deployment**, every change that is made is automatically deployed to production.  In **Continuous Integration**, merging all code from all developers to one central branch of the repository many times a day trying to avoid conflicts in the code in the future. Continuous delivery is an extension of continuous integration to make sure that you can release new changes to your customers quickly in a sustainable way.

1. What role does the Quality Assurance (QA) team play in DevOps?

In DevOps, QA quickly feeds fresh development code into the production environment with the collaboration of the DevOps team, who then provide the necessary tools and infrastructure to make it a process smooth, making sure all changes function as expected.

1. What are the best practices for DevOps implementation?

* Version Control For All Production Artefacts
* Continuous Integration and Deployment
* Automated Acceptance Testing
* Proactive Monitoring of the Production Environment.

1. How will you approach when a project needs to implement DevOps?

The DevOps process flow is all about agility and automation. Each phase in the DevOps lifecycle focuses on closing the loop between development and operations and driving production through continuous development, integration, testing, monitoring and feedback, delivery, and deployment.