What do you mean by software development methodologies?

Software development methodologies are a set of processes, practices, and guidelines that software development teams follow to build high-quality software. These methodologies provide a structured approach to the development process and help to ensure that software is developed efficiently, on time, within budget, and to meet the client's requirements.

What is the software development life cycle?

The software development life cycle (SDLC) is a process used to design, develop, test, and deploy software. It includes various stages, such as requirements gathering, design, implementation, testing, deployment, and maintenance.

Which are popular SDLC models?

Some of the popular SDLC models are Waterfall model, Iterative model, Spiral model, Agile model, and V-shaped model.

Mention a few issues of the Waterfall model. Some of the issues with the Waterfall model include:

The model is inflexible and doesn't allow for changes once a stage has been completed. Testing is done only after the development stage is complete, which may result in significant rework if issues are found. Communication between different teams is limited. The model doesn't prioritize customer feedback, which may lead to a mismatch between the client's requirements and the final product.

What do you mean by the Iterative model? What are the benefits of the Iterative model over the Waterfall model?

The Iterative model is an SDLC model in which the development process is broken down into smaller iterations. Each iteration includes requirements gathering, design, development, and testing. The product is delivered in increments, with each increment adding new features or functionality.

Benefits of the Iterative model over the Waterfall model include:

It allows for flexibility and changes to be made during the development process. Testing is done throughout the development process, which reduces the risk of major issues at the end. It prioritizes customer feedback, which helps to ensure that the final product meets the client's requirements.

What is the Agile manifesto? The Agile manifesto is a set of guiding values and principles for Agile software development. It emphasizes collaboration, flexibility, and responsiveness to change, with the goal of delivering high-quality software that meets the client's requirements.

Mention a few Agile methodologies. Some Agile methodologies include Scrum, Kanban, Lean, Extreme Programming (XP), and Crystal.

What is Scrum? Scrum is an Agile framework for managing and completing complex projects. It emphasizes collaboration, self-organization, and continuous improvement, and it is widely used in software development.

What do you mean by a Sprint? A Sprint is a timeboxed period during which the Scrum team works to deliver a potentially releasable product increment. Sprints are typically two to four weeks long.

What are the different events in Scrum?

The different events in Scrum are:

Sprint planning Daily Scrum (stand-up) Sprint review Sprint retrospective

Why do companies prefer Agile over iterative or waterfall?

Companies prefer Agile over iterative or waterfall for various reasons, such as:

Agile prioritizes customer feedback and collaboration, which helps to ensure that the final product meets the client's requirements. It allows for changes to be made during the development process, which increases flexibility and reduces the risk of major issues at the end. Agile emphasizes continuous improvement and self-organization, which can lead to better teamwork and higher productivity.

What are the responsibilities of a Scrum Master?

The Scrum Master is responsible for ensuring that the Scrum framework is followed, facilitating communication and collaboration among team members, and removing any impediments that may prevent the team from achieving its goals.

Who manages the dev team in scrum ?

In Scrum, the development team is self-organized and self-managing. The team is responsible for managing the development process and for making decisions on how to achieve the goals of the project.

Why is it important to get feedback from client in early stages

It is important to get feedback from the client in the early stages of the development process because it helps to ensure that the final product meets the client's requirements. Getting feedback early allows the development team to make changes and adjustments to the product before too much time and effort have been invested in it. This can help to reduce the risk of major issues or misunderstandings later in the development process and can lead to a better end product. Additionally, getting feedback from the client early can help to build trust and establish a good working relationship between the client and the development team.

What is virtualization?

Virtualization is the process of creating a virtual version of something, such as an operating system, server, storage device, or network resources, using software. The virtual version behaves like the original physical resource and can be managed and utilized in the same way as the physical resource.

What are the types of virtualization?

The main types of virtualization are:

Server virtualization Network virtualization Storage virtualization Desktop virtualization

What are the differences between NAS and SAN?

NAS (Network Attached Storage) and SAN (Storage Area Network) are two different storage technologies used in enterprise environments. The main differences between NAS and SAN are:

NAS is a file-level storage technology that uses a standard Ethernet network to share files across multiple devices. SAN is a block-level storage technology that uses a dedicated network to provide fast and direct access to storage devices. NAS is easy to set up and manage, and is often used for small-scale storage needs. SAN is more complex and expensive to set up and manage, but provides high performance and scalability for large-scale storage needs. NAS is suitable for file-sharing and collaboration, while SAN is more suited for applications that require high-speed data access, such as databases and virtual machines.

What are the differences between type I and type II virtualization?

Type I and Type II virtualization are two different approaches to virtualization. The main differences are:

Type I virtualization runs directly on the host machine's hardware, without the need for a separate operating system. This is also known as bare-metal virtualization. Type II virtualization runs on top of a host operating system, like any other software application.

Type I virtualization provides better performance and security, as it has direct access to the host hardware. Type II virtualization is easier to set up and manage, as it uses a familiar host operating system.

Examples of Type I virtualization include VMware ESXi, Microsoft Hyper-V, and Citrix XenServer. Examples of Type II virtualization include Oracle VirtualBox and VMware Workstation.

In which scenario would you use Type I and Type II?

Type I virtualization is typically used in enterprise environments where performance and security are critical, such as hosting mission-critical applications or virtualizing multiple servers on a single physical host. Type II virtualization is typically used for desktop virtualization, software development, or testing.

Define the following terms:

IaaS (Infrastructure as a Service) - a cloud computing service model that provides virtualized computing resources over the internet, including servers, storage, and networking.

PaaS (Platform as a Service) - a cloud computing service model that provides a platform for developing, testing, and deploying software applications, without the need for managing the underlying infrastructure.

*SaaS (Software as a Service) - a cloud computing service model that provides access to software applications over the internet, without the need for installation or maintenance on the user's device.

FaaS (Function as a Service) - a cloud computing service model that provides serverless computing, where the cloud provider runs and manages individual functions or pieces of code on behalf of the user.\

DaaS (Desktop as a Service) - a cloud computing service model that provides access to a virtual desktop environment over the internet, without the need for a physical desktop computer.

What are the characteristics of cloud?

The main characteristics of cloud computing are:

On-demand self-service Broad network access Resource pooling Rapid elasticity Measured service

What are the differences between private and public cloud?

Private cloud refers to cloud infrastructure that is dedicated to a single organization or enterprise, and is operated within the organization's own data center or on-premises. Public cloud, on the other hand, is a cloud infrastructure that is open to the public and hosted by a third-party service provider. Public cloud infrastructure is shared by multiple organizations, and customers pay for the resources they use.

In which scenario would you prefer using hybrid cloud?

Hybrid cloud is a combination of both private and public cloud infrastructures. Organizations use hybrid cloud in situations where they need to have control over some of their data and applications, but also want the scalability and flexibility of the public cloud for other applications. Hybrid cloud is often used by organizations that have compliance or regulatory requirements that necessitate keeping some data on-premises, while also wanting to take advantage of the benefits of the public cloud.

List popular cloud providers.

Some popular cloud providers include Amazon Web Services (AWS), Microsoft Azure, Google Cloud, IBM Cloud, Oracle Cloud, and Alibaba Cloud.

What do you mean by AWS?

Amazon Web Services (AWS) is a comprehensive cloud computing platform provided by Amazon. It offers a wide range of cloud services, including computing, storage, database, analytics, and machine learning, among others.

How would you create a virtual machine in AWS?

To create a virtual machine (EC2 instance) in AWS, you can follow these general steps:

Log in to the AWS Management Console Navigate to the EC2 dashboard Click on the "Launch Instance" button Choose the appropriate Amazon Machine Image (AMI) for your application Select the instance type and configure the instance settings Set up security group rules to control access to the instance Launch the instance and connect to it List the steps to upload a React application in the cloud.

To upload a React application to the cloud, you can follow these general steps:

Build your React application using a tool such as webpack or create-react-app Choose a cloud provider and create an instance (such as an EC2 instance on AWS) to host your application Install the necessary dependencies and web server (such as nginx or Apache) on the instance Copy the built React application files to the instance Configure the web server to serve the React application files Start the web server and access the application from a web browser List the steps to upload an Express application in the cloud.

To upload an Express application to the cloud, you can follow these general steps:

Choose a cloud provider and create an instance (such as an EC2 instance on AWS) to host your application Install the necessary dependencies and web server (such as nginx or Apache) on the instance Copy the Express application files to the instance Install Node.js and run the application using the appropriate command Configure the web server to forward requests to the Express application Start the web server and access the application from a web browser

Why are companies moving data to the cloud?

Companies are moving data to the cloud because it provides several benefits, including:

Scalability: the ability to quickly and easily scale up or down as needed

Flexibility: the ability to access data and applications from anywhere

Cost savings: the ability to reduce the cost of maintaining physical infrastructure and personnel

Security: the ability to leverage the security expertise and resources of cloud providers

Reliability: the ability to improve uptime and reduce downtime through redundancy and failover

What are the benefits of using cloud ?

Benefits of using cloud:

Scalability: Cloud services allow for easy and quick scaling of resources up or down as needed. Flexibility: Cloud services offer flexibility in terms of the amount and type of resources used, and can be easily adapted to meet changing business needs. Cost-effectiveness: Cloud services can be more cost-effective than traditional on-premises infrastructure, as they eliminate the need for expensive hardware and software investments. Availability: Cloud services offer high availability and can ensure that applications and data are always accessible. Security: Cloud services typically offer robust security measures and can provide better data protection than many on-premises solutions.

What is the issue you face because of scaling ?

Scaling issues:

Load balancing: Distributing workloads across multiple servers can be challenging, especially when dealing with large amounts of traffic. Data consistency: Ensuring that data is consistent across multiple servers can be difficult and requires careful management. Cost: Scaling up infrastructure can be expensive, especially if demand is unpredictable and fluctuates frequently. Vertical vs horizontal scaling:

What are the differences between vertical and horizontal scaling ? which one would you prefer and why ?

Vertical scaling: Involves increasing the resources of a single server, such as adding more CPU or RAM to increase capacity. Vertical scaling is often limited by the hardware capacity of a server, and can be more expensive in the long run.

Horizontal scaling: Involves adding more servers to a system, rather than increasing the resources of a single server. This allows for more resources to be added in a cost-effective way, and can also provide increased availability and fault tolerance.

How to implement horizontal scaling in real world ?

Implementation of horizontal scaling:

Load balancing: Distribute the workload across multiple servers to ensure that no single server is overloaded.

Auto-scaling: Automatically provision additional servers as demand increases, and de-provision them as demand decreases.

Data partitioning: Split data into multiple partitions to enable storage across multiple servers, improving scalability and performance.

The roles In Scrum Include: Product Owner: responsible for defining and prioritizing the product backlog, ensuring that the team is working on the most important features and value. Scrum Master: responsible for facilitating the Scrum process, ensuring that the team is following the Scrum framework and removing any obstacles that may be preventing the team from delivering value. Development Team: responsible for designing, building, and testing the product. A sprint is a timeboxed iteration during which the team works to deliver a set of features. A story is a description of a feature from the user's perspective, usually expressed in the form of a user story. An epic is a large user story that can be broken down into smaller stories. Tasks are the specific work items that the team must complete in order to deliver a story or feature.

Agile project management is a flexible and iterative approach to project management that emphasizes collaboration, continuous improvement, and quick response to change. Traditional project management (Waterfall) is a linear and sequential approach to project management, with distinct phases and a focus on detailed planning upfront. Some differences between the two include: Agile involves more collaboration and communication between team members and stakeholders, whereas Waterfall relies heavily on documentation and formalized communication channels.

Agile projects are more flexible and adaptable to change, whereas Waterfall projects are more rigid and require extensive planning and requirements gathering upfront. Agile emphasizes delivering value to the customer through iterative development, whereas Waterfall aims to deliver a complete and final product at the end of the project.

Selenium is an open-source testing framework for web applications. It allows testers to write automated tests that simulate user interactions with a web application, such as clicking links, filling out forms, and verifying page content. Selenium can be used to test web pages in various browsers and operating systems.

Docker is a platform for building, packaging, and deploying containerized applications. It allows applications to be packaged and run consistently across different environments, making it easier to deploy and scale applications. Some advantages of Docker include improved portability, faster deployment, and better resource utilization.

Docker Compose is a tool for defining and running multi-container Docker applications. It allows developers to define a set of services and dependencies in a single file, and then run and manage those services using a single command. Some advantages of Docker Compose include simplified configuration and deployment, improved scalability, and better resource utilization.

10. What is kubernetes? What do you mean by container orchestration?

Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. It provides a unified API for managing containers, allowing you to deploy and manage applications across multiple hosts and clusters.

Container orchestration refers to the automated management of containers, including deployment, scaling, and networking. It allows you to manage large numbers of containers in a consistent and efficient manner, reducing manual overhead and improving application reliability.

Explain purpose of GIT. Explain commands to clone, pull and push code.

Git is a distributed version control system that helps developers to track and manage changes made to their code, collaborate with other developers, and maintain multiple versions of the same codebase. The purpose of Git is to provide a reliable and efficient way of versioning code that makes it easy to keep track of changes,

fix errors, and manage development workflows. Here are the commands to clone, pull, and push code using Git

Clone: To copy a repository from a remote server to your local machine, use the git clone* command followed by the URL of the repository.

Pull: To update your local repository with the latest changes from the remote repository, use the *git pull* command. For example, if you are working on a team project and other team members have pushed changes to the remote repository.

Push: To upload your local changes to the remote repository, use the 'git push' command followed by the name of the remote repository and the branch you want to push to.

How we can use GIT to create & merge branch?

GIT provides the ability to create and merge branches. A branch is a separate line of development that can be created to isolate changes from the main codebase. Once the changes on the branch have been tested and reviewed, they can be merged back into the main codebase.

To create a new branch in GIT, you can use the 'git branch command. This will create a new branch with the specified name. To switch to the new branch, you can use the *git checkout <branch-name$ command.

To merge changes from one branch to another, you can use the git merge command. This command will merge the specified branch into the currently checked out branch. GIT will automatically attempt to merge any changes that are not in conflict, but will prompt you to resolve any conflicts that it encounters. Once all conflicts are resolved, the merge can be completed.

To create a new branch in GIT, you can use the command "git branch ". To switch to the new branch, use the command "git checkout ". To merge the changes from one branch into another, use the command "git merge ".

What is micro-service? How is its advantage over monolithic application?

A microservice is a type of software architecture that structures an application as a collection of small, independent services. Each service is responsible for a specific task or feature, and communicates with other services over well-defined APIs. The advantages of using a microservice architecture include: Improved scalability and flexibility Easier maintenance and updates Better fault isolation and resilience Ability to use different technologies for different services

What is UML? Explain use case diagram, class diagram and sequence diagram.

UML (Unified Modeling Language) is a standardized visual language for software design. It provides a way to describe the structure and behavior of software systems using diagrams and other notations. Use case diagrams are used to represent the interactions between a system and its users or external systems. Class diagrams are used to represent the structure of the system's classes and their relationships. Sequence diagrams are used to represent the interactions between different parts of the system over time, showing the order of messages and events.