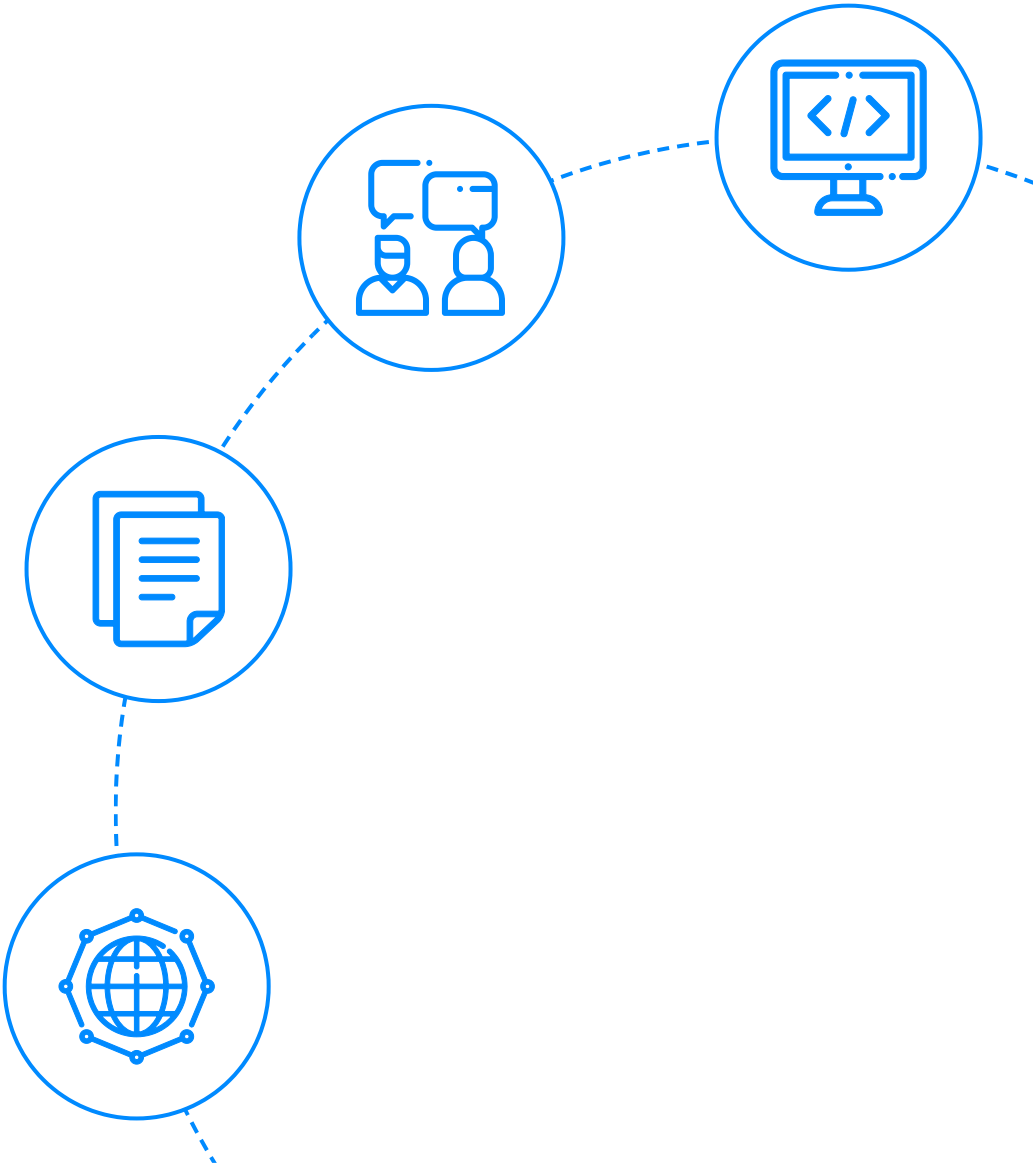
InterviewBit

SDLC Interview Questions



To view the live version of the page, click here.

Contents

SDLC Interview Questions for Freshers

SDLC Interview Questions for Experienced

SDLC Interview Questions for Experienced

(.....Continued)

19. What is Level-0 DFD?
20. What is Capability Maturity Model?
21. What are Capability Maturity Model(CMM) levels?
22. Briefly explain Scrum methodology in the Agile model.
23. What do you know about Scrum impediments?
24. What do a Software Project Manager's responsibilities entail?
25. What is Software Configuration Management, and how does it work?
26. List Top SDLC tools.
27. What are the different environments related to development while following SDLC?
28. What is the main aim of prototyping?
29. Differentiate between quality assurance and quality control?
30. Briefly describe the RAD model.

Let's get Started

Introduction

The Software Development Life Cycle, or SDLC, is a method for producing high-quality, low-cost software in the least amount of time. **SDLC** is a well-structured flow of stages that enables a company to swiftly develop high-quality software that has been thoroughly tested and is ready for production. To assure high-quality goods or services development in hardware or software organisations, this Software Development Life Cycle will ensure delivery of efficient and high-quality software or hardware products to the end customer or clients.

In this blog, you will come across some of the important questions and answers that would help you crack SDLC interviews.

These questions are divided into two categories, one part is for freshers and the other for experienced professionals. So, let's get started.

SDLC Interview Questions for Freshers

1. What is SDLC?

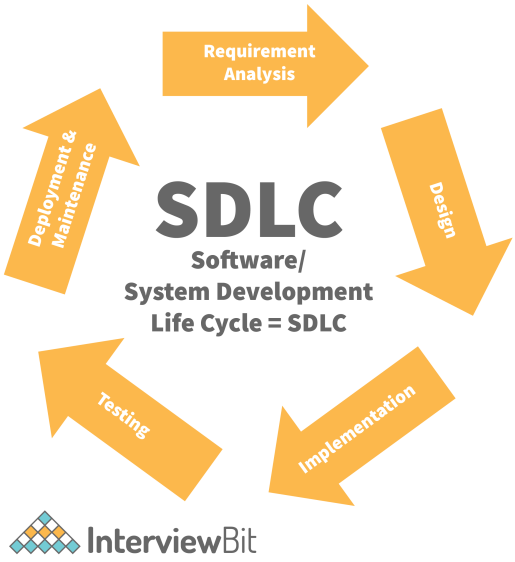
The **Software Development Life Cycle** (SDLC) is a well-defined procedure for producing high-quality, low-cost software in the shortest amount of time possible. The SDLC's purpose is to create exceptional software that exceeds all customer expectations and demands. The SDLC develops and describes a detailed plan that includes stages, or phases, each with its own process and deliverables. It describes the entire development process, including all tasks involved in planning, developing, testing, and distributing a software product.

2. What is the importance of the SDLC process?

Following are some of the points that briefly describes the importance of the SDLC process:

- It explains why project planning, scheduling, and budgeting are necessary.
- It allows for some control over the development process in order to guarantee that the final product meets the original criteria.
- It provides a structure for a standard set of projects and deliverables.
- It helps in the growth process by increasing the visibility of project planning to all involved parties.
- It assures that the design and testing processes that lead to a solution's release are good and well-managed.
- It is the most effective method for project management and tracking.
- It aids in the acceleration of development and the improvement of client relationships.
- It aids in the reduction of project risk and overhead in the project administration plan.

3. Explain the phases in a typical SDLC process briefly.



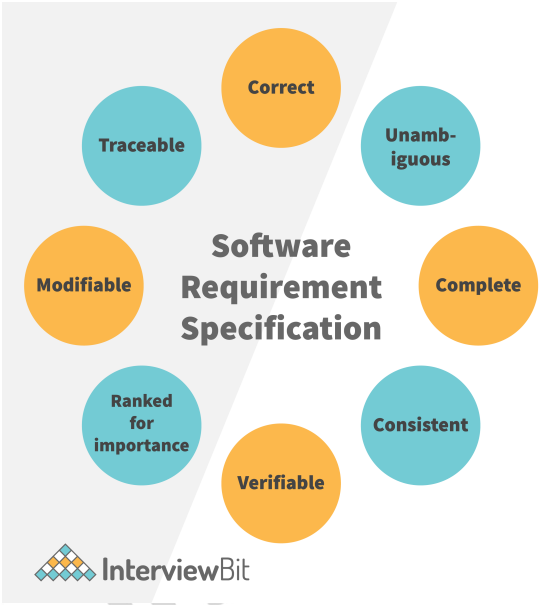
- **Planning:** The first stage of the SDLC is all about determining, what do clients want. Project planning is an important component of the software delivery lifecycle because it is here that the team estimates the cost and outlines the new program's needs.
- **Gathering Requirements:** Defining requirements is part of the planning process to figure out what the application is supposed to perform and what it needs. The development team examines the requirements while keeping the software's design and code in mind.
- **Design:** The following phase entails distilling all of the software project's requirements, analysis, and design information. This phase is the culmination of the previous two, such as customer feedback and requirement collecting. It is a simulation of how a software application will work. Some particulars of this phase are architecture, platform, security and user interface.
- **Development:** This is where the code is really written. Writing code is the first step in putting a design into action. Developers must adhere to the coding requirements set forth by their bosses. Many other jobs are included in the coding process. Many developers need to brush up on their abilities or collaborate with others. It's vital to find and resolve problems and flaws. If any changes or upgrades are needed, the developers can show the work to the business analysts.
- **Testing:** Before making an application available to users, it's vital to test it. The testing team examines the system's overall functionality. This phase aids with reducing the number of faults and issues seen by consumers. As a result, there is a higher level of user satisfaction and a higher rate of utilisation.
- **Deployment:** Once the product has been thoroughly tested and is ready for deployment, it is made available to customers. The deployment's complexity is determined by the project's size. Many businesses prefer to have the deployment step automated.
- **Maintenance:** The developed product is looked after throughout this period. The programme is updated on a regular basis to keep up with the changing user end environment or technology. Users find flaws that were not discovered during testing. These issues must be addressed, which may result in new development cycles.

4. Explain the types of SDLC models.

Following are the predominant models that come under SDLC:

- **Waterfall model:** The waterfall model is a prominent software engineering and product development approach that takes a linear, sequential approach to the software development life cycle (SDLC). The waterfall approach emphasises a logical step-by-step process. It was the first model in the software business to be extensively adopted. It is divided into phases, with one phase's output becoming the input for the next. **Learn More**.
- **Agile model:** Agile approaches divide jobs into smaller iterations or sections and avoid long-term planning entirely. The scope and requirements of the project are defined at the start of the development phase. The number of iterations, duration, and scope of each iteration are all clearly determined ahead of time. In the Agile process model, each iteration is a small-time "frame" that lasts anywhere from one to four weeks. **Learn More**.
- **Iterative model:** One of the most straightforward software development life cycle models is the iterative approach. There are several situations when the initial or basic software requirements are well-defined, but the project's complete scope or set of features is unclear. It primarily focuses on preliminary growth and design, then gradually develops momentum as more complex and needs are met until the final software is completely constructed. **Learn More**.
- **Spiral model:** The spiral model is a risk management strategy that combines the iterative development process model with parts of the Waterfall approach. The spiral approach is preferred by software engineers for large, expensive, and complex projects. **Learn More**.
- **V-model model:** The V-model is an SDLC paradigm in which processes are executed in a V-shape in a sequential manner. The Verification and Validation model is another name for it. The waterfall model is extended by the V-Model. Every phase of the development cycle has a testing phase that is directly linked to it. **Learn More**.

5. What is SRS?

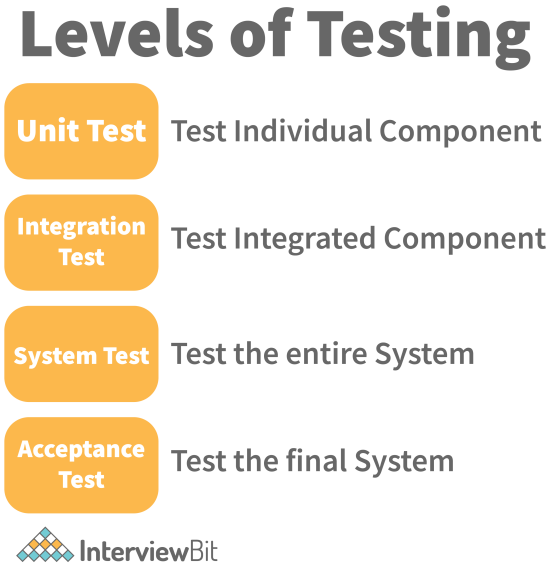


A **Software Requirements Specification** (SRS) is a document that explains what the software will accomplish and how it will work. It is a formal report that serves as a representation of software and allows customers to assess whether it (SRS) meets their needs. It also outlines the functionality that the product must have in order to meet the needs of all stakeholders. This report is created after all requirements have been solicited and analysed, and it serves as a foundation for software engineering tasks.

6. What is a Feasibility Study?

As the name implies, a feasibility study is an analysis or measurement of a software product in terms of how advantageous product development will be for the business in terms of practicality. It determines whether the project is legally, technically, and commercially feasible.

7. What is the testing phase in the SDLC model?



One of the most important procedures in the Software Development Lifecycle is testing (SDLC). It enables businesses to do a thorough evaluation of software and verify that their product meets the expectations of their customers. The testing procedures' main purpose is to report, monitor, troubleshoot, and retest software components until they meet the quality requirements established in the initial SRS. During the testing phase, there are numerous forms of testing, including quality assurance testing (QA), system integration testing (SIT), and user acceptability testing (UAT).

8. Which SDLC model is best and Why?

According to the annual State of Agile report, **Agile** is the best SDLC methodology and also one of the most widely utilised SDLC in the IT industry. Unlike other predictive approaches, the adaptive agile methodology does not necessitate comprehensive preparation. If a change is required, it can be made during the sprint. It's ideal for projects that require a lot of customer involvement and projects that have a constantly changing environment.

9. What are the advantages of the SDLC process?

At the end of each stage, a formal review is established to provide for maximum management oversight. SDLC aids in the creation of extensive system documentation. This guarantees that system needs can be linked to specified business goals. It generates a large number of intermediate products that may be evaluated to see if they fit the user's requirements and adhere to industry standards. These can be improved further if necessary, ensuring that the company receives exactly what it requires.

10. What are the disadvantages of the SDLC process?

Before you begin, make sure you understand all of the details of the project. During the development phase, there was a lot of paperwork. It is difficult to alter or change due to a lack of flexibility. If the planning isn't done properly, the project will take longer and cost more. When there are a lot of flaws in the code, fixing them can take a long time and cause deadlines to be missed.

11. Explain HLD.

It refers to the general design of the system. It describes the application's overall description and architecture. It includes a system architecture description, database design, a brief overview of systems, services, platforms, and module relationships. From the primary module to all submodules, it creates the overall architecture of the system. Architects will provide the High-Level Design in order to begin the development process. This is quite helpful for developers in comprehending the system's flow.

12. Explain LLD.

LLD (Low-LevelLevel Design) is a term that refers to the process of detailing. It provides a full description of each module, including actual logic for each system component and a thorough examination of each module's specifications. Every program undergoes logic design, which is subsequently recorded as program specifications. A unit test plan is prepared for each software. The micro-level or intricate design is another name for it. After the High-Level Design, the Low-Level Design is created.

SDLC Interview Questions for Experienced

13. What is the use of JAD session?

JAD is a strategy for defining business system requirements that are commonly utilised in the early phases of a systems development project. JAD's goal is to bring MIS and end-users together in a structured workshop setting in order to extract outcome system needs. It allows clients and developers to swiftly agree on a project's fundamental scope, objectives, and specifications.

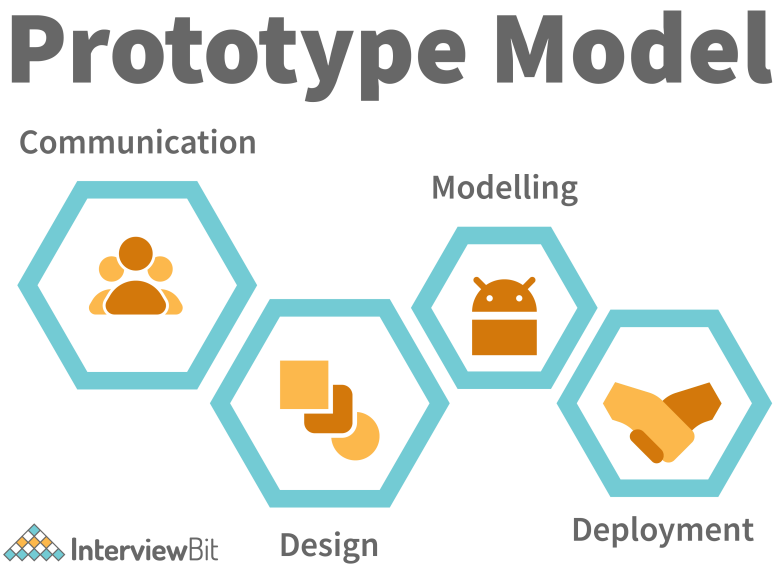
14. What is the Software release process?

The Software Development Life Cycle (SDLC) release phase is historically connected with production, deployment, and post-production operations, which generally include software maintenance and support. So, release management is the process of managing, planning, scheduling, and controlling a full software development at every stage and environment, including testing and releasing software releases.

15. What is FRS document?

This document captures the user's voice from the outside, or the end user's perspective. A Business System Analyst creates it (BSA). This paper demonstrates how a system will react when a user interacts with it in order to meet the BRD and SRD standards. The key area of interest for software experts is the Functional Requirement Specification (FRS). An FRS is useful for software testers to learn the situations in which the product is intended to be tested, just as it is for developers to understand what product they are planning to produce. An FRS's ultimate purpose is to meet all of the requirements outlined in the SRS and BRS regulations.

16. Explain prototyping in SDLC process.



Building software application prototypes that display the capabilities of the product under development, but may not have the exact logic of the original software, is referred to as software prototyping. It's built, tested, and revised until it's deemed suitable as a prototype. It also serves as a foundation for the final system or programme. It's best used in situations where the project's requirements aren't fully understood. Software prototyping is gaining popularity as a software development strategy because it allows for an early understanding of customer requirements.

17. What are different types of prototype model?

There are four types of Prototyping models:

- Rapid Throwaway prototypes.
- Evolutionary prototype.
- Incremental prototype.
- Extreme prototype.

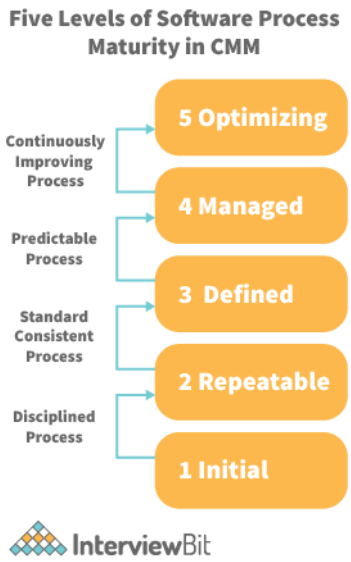
18. How can DDLC and SDLC work together?

The **DDLC** (Documentation Development Life Cycle) is a software documentation development life cycle used by technical documenters to prepare software documentation. The life cycle is followed in tandem with the SDLC, as testers and developers work on the programme at the same time. Because the documentation requires input and feedback from the various phases of the SDLC, the DDLC has stages that are comparable to the SDLC.

19. What is Level-0 DFD?

Context Diagram is another name for DFD Level 0. It's a high-level overview of the entire system or process that's being studied or modelled. It's meant to be a quick peek into the system, displaying it as a single high-level process with its connections to external entities. Stakeholders, business analysts, data analysts, and developers should all be able to understand it readily.

20. What is Capability Maturity Model?



The Capability Maturity Model (CMM) is a cross-discipline and technical paradigm for facilitating and refining software development processes and system improvement. This methodology is at the heart of most management systems that aim to improve the quality of all product and service development and delivery.

21. What are Capability Maturity Model(CMM) levels?

Following are the five Capability Maturity Model Levels:

- **Initial:** The first step is to create an unstable process environment. The software development process is considered haphazard and even chaotic at times. There are few methods that have been specified, and success is based on individual effort and heroism.
- **Repeatable:** Work is planned and monitored, making it repeatable. To track cost, schedule, and functionality, basic project management techniques are implemented.
- **Defined:** This level encompasses written and defined standards that evolve over time and support consistent performance. The work is well defined at this point.
- **Managed:** Extensive data on the software development process and product quality are gathered. Both the software development process and the end products are quantified and managed.
- **Optimized:** Work is based on continuous improvement (optimization). The focus on continuously improving process performance is a significant feature of this level.

22. Briefly explain Scrum methodology in the Agile model.

Scrum is an agile development approach based on iterative and incremental procedures that are used in the creation of software. It's an agile structure that's adaptable, rapid, flexible, and excellent at delivering value to customers throughout the project's development. Companies of all sizes employ the Agile Scrum technique because of its ability to provide high-end cooperation and efficiency for project-based work. Scrum is a sort of agile approach that breaks projects down into manageable parts known as "sprints." The Agile Scrum methodology is ideal for companies who need to complete projects fast.

23. What do you know about Scrum impediments?

Obstacles or challenges that the scrum team faces slow down their work speed are referred to as **impediments**. An obstacle is anything that tries to prevent the scrum team from getting work "Done." Impediments can take many different forms. Some of the roadblocks include resource shortages or sick team members, technical, operational, and organisational issues, a lack of management support systems, and business issues.

24. What do a Software Project Manager's responsibilities entail?

The Software Project Manager is in charge of seeing the project through to completion. The Software Project Manager is responsible for ensuring that the entire team follows a methodical and well-defined approach to software development. They also handle project planning, tracking project status, resource management, and risk management.

25. What is Software Configuration Management, and how does it work?

The process of tracking and regulating changes that occur during the software development lifecycle is known as software configuration management. Any modification made during the development of software must be tracked using a well-defined and controlled process. Any modifications performed during software development are regulated through a well-defined process, thanks to configuration management. Revision control and the establishment of baselines are two SCM procedures.

26. List Top SDLC tools.

- **Jira:** This software is intended to make workflow management easier for a wide range of groups. Jira was created with the intention of being a simple system for recording tasks and errors. However, it has since matured into a robust workflow management solution.
- **Git** is a distributed version management system that is open-source. Developers aiming to examine changes and contributions to the overall code might considerably benefit from a version control system or VCS. This software customisation management tool is an important part of the SDLC.
- **Confluence:** During this stage, Confluence is a wonderful tool for developing product research docs and sharing design assets.
- **Asana:** From daily activities to larger projects, Asana assists teams in orchestrating their work. Teams are more confident, move faster, and accomplish more with less when they use Asana, regardless of where they are based.

27. What are the different environments related to development while following SDLC?

Following environments is used while following SDLC:

- **Dev:** A development environment is a workspace where developers may make changes without damaging anything in a live environment. The development environment is frequently referred to as a workspace for developers.
- **SIT/QA:** System Integration Testing/ Quality Analysis: In a QA environment, you test your update operation against data, hardware, and software that closely resembles the production environment, and you allow intended users to test the outcome.
- **UAT:** User Acceptance Testing: User acceptance testing (UAT) environments, also known as staging environments, let the application's primary users try out new features before they're deployed into production.
- **PROD:** The "Production" environment, sometimes known as "Live" is where real customers/users interact with the software product.

28. What is the main aim of prototyping?

It mainly offers a mini-model of the proposed system.

29. Differentiate between quality assurance and quality control?

- Quality Assurance ensures that the software delivered has the fewest possible defects. Quality Control is the process of ensuring that a product's quality is maintained over time.
- Quality Assurance is handled by the project's testing team, whereas Quality Control is handled by a dedicated support team that is accountable for the product's quality even if it is in the maintenance phase of software engineering.

30. Briefly describe the RAD model.

The Rapid Application Development (RAD) paradigm is a software development method that relies on prototyping rather than detailed design. It should be utilised when a system that can be modularized in two to three months is required. It should be employed if there is a large number of designers available for modelling and the budget allows for their costs as well as the costs of automated code generation technologies.

Additional Interview Preparation Resources

- Software Engineer / Developer Salary in India
- Software Developer Vs Software Engineer
- Software Engineer MCQs
- Agile Interview Questions

Links to More Interview Questions

C Interview Questions	Php Interview Questions	C Sharp Interview Questions
Web Api Interview Questions	Hibernate Interview Questions	Node Js Interview Questions
Cpp Interview Questions	Oops Interview Questions	Devops Interview Questions
Machine Learning Interview Questions	Docker Interview Questions	Mysql Interview Questions
Css Interview Questions	Laravel Interview Questions	Asp Net Interview Questions
Django Interview Questions	Dot Net Interview Questions	Kubernetes Interview Questions
Operating System Interview Questions	React Native Interview Questions	Aws Interview Questions
Git Interview Questions	Java 8 Interview Questions	Mongodb Interview Questions
Dbms Interview Questions	Spring Boot Interview Questions	Power Bi Interview Questions
Pl Sql Interview Questions	Tableau Interview Questions	Linux Interview Questions
Ansible Interview Questions	Java Interview Questions	Jenkins Interview Questions