

# Software Engineering Interview Questions

A list of frequently asked **Software Engineering Interview Questions and Answers** are given below.

## What is Software Engineering?

**Software engineering is defined as the function of the systematic, disciplined, quantified approach to the development, operations, and maintenance of software.**

---

## What are the elements to be considered in the System Model Construction?

**Elements to be considered in the System Model Construction are:**

**Assumption ◦ Simplification ◦ Limitation ◦ Constraints ◦ Preferences**

---

## What does a System Engineering Model accomplish?

**System Engineering Model accomplishes the following:**

**Define Processes that serve needs of view . Represent behavior of process and assumption . Explicitly define Exogenous and Endogenous Input . It represents all Linkages that enable an engineer to understand aspect better.**

---

### **Define Framework.**

**A framework is the Code Skeleton that can be fleshed out with particular classes or functionality and designed to address the specific problem at hand.**

---

### **5) What are the characteristics of the software?**

**Characteristics of the software are:**

**Software is engineered, not manufactured. . Software does not wear out.**

**Most software is custom-built rather than being assembled from components.**

---

### **6) What are the various categories of software?**

**The various categories of software are:**

**System software Application.**

**Software Engineering / Scientific. . Software Embedded software.**

**Web Applications. . Artificial Intelligence software.**

---

### **7) What are the challenges in software?**

**The challenges in the software are:**

**Copying with legacy systems.**

**Heterogeneity challenge.**

**Delivery times challenge.**

---

### **8) Define Software process.**

**A software process is defined as the structured set of activities that are required to develop the software system.**

---

### **9) What are the internal milestones?**

**They are the significant and quantifiable attributes of progress. They are the standard methods in the project which provide that we are on the right track. They are under the authority of the project manager.**

---

### **10) What is the limitation of RAD Model?**

**Limitation of RAD Model are:**

**It requires a sufficient number of Human Resources to create enough number of teams.**

**Developers and Users are not committed, the system fails.**

**It is not Properly Modularized building component may be Problematic.**

**It is not applicable when there is more possibility for Technical Risk.**

---

### **11) What are the disadvantages of classic life cycle model?**

**Disadvantages of the classic life cycle model are: ◦ Real projects rarely follow the sequential flow. Iteration always occurs and creates a problem.**

**Challenging for the customer to state all requirements. ◦ The working version of the program is not available. So the customer must have patience.**

---

### **12) What are the merits of the incremental model?**

**The merits of the incremental model are:**

**The incremental model can be accepted when there is less number of people include in the project.**

**Technical risks can be handle with each increment.**

**For a minimal period, at least the core product can be delivered to the user.**

---

### **13) What is the disadvantage of the spiral model?**

**The disadvantage of the spiral model are:**

**It is based on user communication. If the interface is not proper, then the software product which gets created will not be the up to the mark.**

**It demands a vast risk assessment. If the risk assessment is completed correctly, then only the successful product can be obtained.**

---

### **14) Name the Evolutionary process Models.**

**Evolutionary powers models are:**

**Incremental model ◦ Spiral model ◦ WIN-WIN spiral model  
◦ Concurrent Development**

---

### **15) Define Software Prototyping.**

**Software prototyping is represented as rapid software development for validating the requirements.**

---

### **16) What are the benefits of prototyping?**

**The benefits of prototyping are:**

**Prototype services as a basis for developing system specification.**

**Design quality can be revised. ◦ The system can be managed easily.**

**Development efforts may get decreased. ◦ System usability can be upgraded.**

---

### **17) What are the prototyping methods in software process?**

**The prototyping methods in the software process are:**

**Evolutionary prototyping: In this method of system development, the initial prototype is arranged, and it is then precise through the number of phases to the final stage.**

**Throw-away prototyping: Using this method, a rough practical implementation of the system is produced. The requirement issues can be identified from this implementation. It is then rejected. System is then developed using some various engineering paradigm.**

---

### **18) What are the advantages of evolutionary prototyping?**

**The advantages of evolutionary prototyping are:**

**Fast delivery of the working system.**

**User is contained while developing the system. ◦ The more useful system can be delivered.**

**Specification, design and implementation work in equivalent manner.**

---

### **19) What are the various Rapid prototyping techniques?**

**The various rapid prototyping techniques are:**

**Dynamic high-level language development.**

**Database programming. ◦ Component and application assembly.**

---

### **20) What are the uses of User-Interface Prototyping?**

**This prototyping is used to pre-specify the looks and effectively feel of customer interface.**

---

### **21) What is the principle of the prototype model?**

**A prototype is built to quickly determine to the user what the product would look like. The only minimal functionality of the actual product is supported during the prototyping phase.**

---

### **Define System Context Diagram (SCD)?**

**System Context Diagram (SCD):**

- Establish data boundary between System being implemented and Environment in which system operates.**
- Describes all external producers, external consumers, and entities that communicate through the customer interface.**

---

### **Define Quality Function Deployment (QFD)?**

**Quality Function Deployment (QFD) is a method that translates the needs of the user into a technical requirement. It concentrates on maximizing user satisfaction from the software engineering process.**

---

### **What is Requirement Engineering?**

**Requirement engineering is the process of establishing services which the user required from the system and constraint under which it operates and is developed.**

---

### **What is ERD?**

**Entity Relationship Diagram is the graphical description of the object relationship pair. It is primarily used in the database application.**

---

### **What is DFD?**

**Data Flow Diagram depicts the data flow and the transforms which are applied to the data as it moves from input to output.**

---

### **27) What is a state transition diagram?**

**State transition diagram is a collection of states and events. The events cause the operation to change its state. It also describes what actions are to be taken on the occurrence of particular events.**

### **What is Software Quality Assurance?**

---

**Software Quality Assurance is a set of auditing and documenting functions that assess the effectiveness and completeness of quality control activities.**

---

**What is the use of CMM?**

**Software Quality means Conformance to state functional explicitly and performance requirements, explicitly documented development standards, inherent characteristics expected for professionally developed software.**

---

**30) What is coupling?**

**Coupling is the significant measure of the degree to which classes are linked to one another. Coupling should be kept as low as possible.**

---

**31) What is cohesion?**

**Cohesion is the indication of the relative functional strength of a module. It is a natural extension of Information Hiding and Performs a single task, requiring little integration with other components.**

---

**32) Define Refactoring.**

**Refactoring means changing a software system in a way that does not alter the external behavior of code.**



---

### 33) What is Software Architecture?

**Software Architecture means the overall structure of the software and how that software provides conceptual integrity for the system.**

---

### 34) Define Stamp coupling.

**When a portion of the data structure is passed via the module interface, then it is called as stamp coupling.**

---

### 35) Define common coupling.

**When several modules reference a global data area, then the coupling is called common coupling.**

---

### 36) Define temporal cohesion.

**When a module contains tasks that are related by the fact that all must be executed within the same period, then it is termed as temporal cohesion.**

---

### 37) Define metrics.

**Metrics are defined as the degree to which a system component or process possesses a given attribute.**

---

### 38) What is COCOMO model?

**Constructive Cost Model is a cost model, which gives the estimate of several staff-months it will take to develop the software product.**

---

### **39) What is the purpose of the timeline chart?**

**The objective of the timeline chart is to emphasize the scope of the individual task. Hence set of functions are given as input to the timeline chart.**

---

### **40) Define Smoke Testing?**

**Smoke testing is Integration Testing and frequently used when software products are being developed.**

---

### **41) What are the benefits of Smoke Testing?**

**Benefits of doing Smoke Testing are:**

**Integration Risk is minimized.**

**Quality of end-product is improved.**

**Error diagnosis and Correction are simplified. • Progress is easy to assess.**

---

### **42) What is Equivalence Partition?**

**Equivalence Partitions Derives an input domain of a program into classes of data from which test cases are derived. It is a Set of Objects have linked by relationships as Symmetric, Transitive, and Reflexive an equivalence class is present.**

---

### **43) What are the steps followed in testing?**

**The steps followed in testing are:**

**Unit testing:** The individual elements are tested in this type of testing.

**Module testing:** Related group of independent items is tested. **Sub-system testing:** This is a type of integration testing. Different modules are integrated into a sub-system, and the entire subsystem is tested.

**System testing:** The entire system is tested in this system.

**Acceptance testing:** This type of testing contains testing of the system with user data if the system behaves as per client need, then it is accepted.

---

#### **44) Distinguish between Alpha and Beta testing.**

**Alpha and Beta testings** are the two types of acceptance testing.

**Alpha test:** The alpha testing is attesting in which the customer tests the version of complete software under the supervision of the developer. This testing is implement at the developer's site.

**Beta test:** The beta testing is a testing in which the customer tests the version of the software without the developer being present. This testing is performed at the customer's site.

---

#### **45) What are the types of Static Testing tools?**

**There are the three types of static testing tools.**

**Code-based testing tools:** These tools take source code as input and generate test cases.

**Specialized testing tools:** Using this language, the detailed test specification can be written for each test case.

**Requirement-based testing tools:** These tools help in designing as per user requirements.

---

#### **46) Define maintenance.**

**Maintenance** is described as the process in which changes are implemented by either modifying the existing system's architecture or by adding new components to the

---

#### **47) What are the types of software maintenance?**

**Types of software maintenance are:**

**Corrective Maintenance:** It means the maintenance for correcting the software faults.

**Adaptive maintenance:** It means maintenance for adapting the change in environment.

**Perfective maintenance:** It means modifying or enhancing the system to meet the new requirements.

**Preventive maintenance:** It means changes made to improve future maintainability.

---

#### **What is CASE Tools?**

**CASE Tools** stands for Computer-Aided Software Engineering. It is system software that provides automated support for software process activities. It contains program

**used to support software process operations such as Requirement Analysis, System Modeling. Debugging and Testing.**

---

### **What is Risk management?**

**Risk management is the phase of anticipating hurdles in carrying out the original plan and providing alternate methods so that the impact on the anticipated initially outcome is minimal.**

**Q.What is computer software?**

**A. Computer software is a complete package, which includes software program, its documentation and user guide on how to use the software. Q.Can you differentiate computer software and computer program?**

**A. A computer program is piece of programming code which performs a well defined task where as software includes programming code, its documentation and user guide.**

**Q.What is software engineering?**

**A. Software engineering is an engineering branch associated with software system development.**

**Q.When you know programming, what is the need to learn software engineering concepts?**

**A. A person who knows how to build a wall may not be good at building an entire house. Likewise, a person who can write programs may not have knowledge of other concepts of Software Engineering. The software engineering concepts guide programmers on how to assess requirements of end user, design the algorithms before actual coding starts, create programs by coding, testing the code and its documentation.**

**Q.What is software process or Software Development Life Cycle (SDLC)?**

**A. Software Development Life Cycle, or software process is the systematic development of software by following every stage in the development process namely, Requirement Gathering, System Analysis, Design, Coding, Testing, Maintenance and Documentation in that order.**

**Q. What are SDLC models available?**

**A. There are several SDLC models available such as Waterfall Model, Iterative Model, Spiral model, V-model and Big-bang Model etc. Q. What are various phases of SDLC?**

**A. The generic phases of SDLC are: Requirement Gathering, System Analysis and Design, Coding, Testing and implementation. The phases depend upon the model we choose to develop software. Q. Which SDLC model is the best?**

**A. SDLC Models are adopted as per requirements of development process. It may vary software-to-software to ensuring which model is suitable.**

**We can select the best SDLC model if following answers are satisfied -**

**Is SDLC suitable for selected technology to implement the software ?**

**Is SDLC appropriate for client's requirements and priorities ?**

**Is SDLC model suitable for size and complexity of the software ?**

**Is the SDLC model suitable for type of projects and engineering we do ?**

**Is the SDLC appropriate for the geographically co-located or dispersed developers ?**

**Q. What is software project management?**

**A. Software project management is process of managing all activities like time, cost and quality management involved in software development. Q. Who is software project manager?**

**A. A software project manager is a person who undertakes the responsibility of carrying out the software project. Q. What does software project manager do?**

**A. Software project manager is engaged with software management activities. He is responsible for project planning, monitoring the progress, communication among stakeholders, managing risks and resources, smooth execution of development and delivering the project within time, cost and quality constraints.**

**Q.What is software scope?**

**A. Software scope is a well-defined boundary, which encompasses all the activities that are done to develop and deliver the software product.**

**The software scope clearly defines all functionalities and artifacts to be delivered as a part of the software. The scope identifies what the product will do and what it will not do, what the end product will contain and what it will not contain.**

**Q.What is project estimation?**

**A. It is a process to estimate various aspects of software product in order to calculate the cost of development in terms of efforts, time and resources. This estimation can be derived from past experience, by consulting experts or by using pre-defined formulas.**

**Q.How can we derive the size of software product?**

**A. Size of software product can be calculated using either of two methods**

**-**

**Counting the lines of delivered code**

**Counting delivered function points Q.What are function points?**

**A. Function points are the various features provided by the software product. It is considered as a unit of measurement for software size.**

**Q.What are software project estimation techniques available?**

**A. There are many estimation techniques available.The most widely used are -**

**Decomposition technique (Counting Lines of Code and Function Points)**

**Empirical technique (Putnam and COCOMO). Q.What is baseline?**

**A. Baseline is a measurement that defines completeness of a phase. After all activities associated with a particular phase are accomplished, the phase is complete and acts as a baseline for next phase. Q.What is Software configuration management?**

**A. Software Configuration management is a process of tracking and controlling the changes in software in terms of the requirements, design, functions and development of the product. Q.What is change control?**

**A. Change control is function of configuration management, which ensures that all changes made to software system are consistent and made as per organizational rules and regulations.**

**Q.How can you measure project execution?**

**A. We can measure project execution by means of Activity Monitoring, Status Reports and Milestone Checklists.**

**Q.Mention some project management tools.**

**A. There are various project management tools used as per the requirements of software project and organization policies. They include Gantt Chart, PERT Chart, Resource Histogram, Critical Path Analysis, Status Reports, Milestone Checklists etc. Q.What are software requirements?**

**A. Software requirements are functional description of proposed software system. Requirements are assumed to be the description of target system, its functionalities and features. Requirements convey the expectations of users from the system.**

**Q.What is feasibility study?**

**A. It is a measure to assess how practical and beneficial the software project development will be for an organization. The software analyzer conducts a thorough study to understand economic, technical and operational feasibility of the project.**



**Economic - Resource transportation, cost for training, cost of additional utilities and tools and overall estimation of costs and benefits of the project.**

**Technical - Is it possible to develop this system ? Assessing suitability of machine(s) and operating system(s) on which software will execute, existing developers' knowledge and skills, training, utilities or tools for project.**

**Operational - Can the organization adjust smoothly to the changes done as per the demand of project ? Is the problem worth solving ?**

**Q.How can you gather requirements?**

**A. Requirements can be gathered from users via interviews, surveys, task analysis, brainstorming, domain analysis, prototyping, studying existing usable version of software, and by observation. Q.What is SRS?**

**A. SRS or Software Requirement Specification is a document produced at the time of requirement gathering process. It can be also seen as a process of refining requirements and documenting them.**

**Q.What are functional requirements?**

**A. Functional requirements are functional features and specifications expected by users from the proposed software product. Q.What are non-functional requirements?**

**A. Non-functional requirements are implicit and are related to security, performance, look and feel of user interface, interoperability, cost etc.**

**Q.What is software measure?**

**A. Software Measures can be understood as a process of quantifying and symbolizing various attributes and aspects of software. Q.What is software metric?**

**A. Software Metrics provide measures for various aspects of software process and software product. They are divided into –**

**Requirement metrics : Length requirements, completeness**

**Product metrics :Lines of Code, Object oriented metrics, design and test metrics**

**Process metrics: Evaluate and track budget, schedule, human resource.**

**Q.What is modularization?**

**A. Modularization is a technique to divide a software system into multiple discrete modules, which are expected to carry out task(s) independently. Q.What is concurrency and how it is achieved in software?**

**A. Concurrency is the tendency of events or actions to happen simultaneously. In software, when two or more processes execute simultaneously, they are called concurrent processes.**

**Example**

**While you initiate print command and printing starts, you can open a new application.**

**Concurrency, is implemented by splitting the software into multiple independent units of execution namely processes and threads, and executing them in parallel. Q.What is cohesion?**

**A. Cohesion is a measure that defines the degree of intra-dependability among the elements of the module. Q.What is coupling?**

**A. Coupling is a measure that defines the level of inter-dependability among modules of a program.**

**Q.Mentions some software analysis & design tools?**

**A. These can be: DFDs (Data Flow Diagrams), Structured Charts, Structured English, Data Dictionary, HIPO (Hierarchical Input Process Output) diagrams, ER (Entity Relationship) Diagrams and Decision tables. Q.What is level-0 DFD?**

**A. Highest abstraction level DFD is known as Level 0 DFD also called a context level DFD, which depicts the entire information system as one diagram concealing all the underlying details.**

**Q.What is the difference between structured English and Pseudo Code?**

**A. Structured English is native English language used to write the structure of a program module by using programming language keywords, whereas, Pseudo Code is more close to programming language and uses native English language words or sentences to write parts of code. Q.What is data dictionary?**

**A. Data dictionary is referred to as meta-data. Meaning, it is a repository of data about data. Data dictionary is used to organize the names and their references used in system such as objects and files along with their naming conventions.**

**Q.What is structured design?**

**A. Structured design is a conceptualization of problem into several wellorganized elements of solution. It is concern with the solution design and based on 'divide and conquer' strategy.**

**Q.What is the difference between function oriented and object oriented design?**

**A. Function-oriented design is comprised of many smaller sub-systems known as functions. Each function is capable of performing significant task in the system. Object oriented design works around the real world objects (entities), their classes (categories) and methods operating on objects (functions).**

**Q.Briefly define top-down and bottom-up design model.**

**A. Top-down model starts with generalized view of system and decomposes it to more specific ones, whereas bottom-up model starts with most specific and basic components first and keeps composing the components to get higher level of abstraction.**

**Q.What is the basis of Halstead's complexity measure?**

**A. Halstead's complexity measure depends up on the actual implementation of the program and it considers tokens used in the program as basis of measure.**

**Q.Mention the formula to calculate Cyclomatic complexity of a program?**

**A. Cyclomatic complexity uses graph theory's formula:  $V(G) = e - n + 2$**

**Q.What is functional programming?**

**A. Functional programming is style of programming language, which uses the concepts of mathematical function. It provides means of computation as mathematical functions, which produces results irrespective of program state.**

**Q.Differentiate validation and verification?**

**A. Validation checks if the product is made as per user requirements whereas verification checks if proper steps are followed to develop the product.**

**Validation confirms the right product and verification confirms if the product is built in a right way.**

**Q.What is black-box and white-box testing?**

**A. Black-box testing checks if the desired outputs are produced when valid input values are given. It does not verify the actual implementation of the program.**

**White-box testing not only checks for desired and valid output when valid input is provided but also it checks if the code is implemented correctly.**

Criteria	Black Box Testing	White Box Testing
Knowledge of software program, No design and structure essential		Yes
Knowledge of Software No Implementation essential		Yes
Who conducts this test on Software Testing software Employee		Software Developer

**Requirements baseline reference for tester specifications**

**Design and structure details**

**Q. Quality assurance vs. Quality Control?**

**A. Quality Assurance monitors to check if proper process is followed while software developing the software.**

**Quality Control deals with maintaining the quality of software product.**

**Q. What are various types of software maintenance?**

**A. Maintenance types are: corrective, adaptive, perfective and preventive.**

**Corrective**

**Removing errors spotted by users**

**Adaptive tackling the changes in the hardware and software environment where the software works**

**Perfective maintenance implementing changes in existing or new requirements of user**

**Preventive maintenance taking appropriate measures to avoid future problems**

**Q. What is software re-engineering?**

**A. Software re-engineering is process to upgrade the technology on which the software is built without changing the functionality of the software. This is done in order to keep the software tuned with the latest technology.**

**Q. What are CASE tools?**

**A. CASE stands for Computer Aided Software Engineering. CASE tools are set of automated software application programs, which are used to support, accelerate and smoothen the SDLC activities.**