INTERVIEW QUESTIONS

on

SOFTWARE ENGINEERING

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
- Implementation. T

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 very 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? What is his role?

A. A software project manager is a person who undertakes the responsibility of carrying out the software project.

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 contraints.

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 discreet 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),
- Data Dictionary
- Use Case Diagram
- State Diagram
- UML Diagram
- HIPO (Hierarchical Input Process Output) diagrams
- ER (Entity Relationship) Diagrams
- Decision tables etc.

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 structured design?

A. Structured design is a conceptualization of problem into several well-organized 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. In software development process what is the meaning of debugging?

A. Debugging is the process that results in the removal of error. It is very important part of the successful testing.

Q.Differentiate validation and verification?

A.

Criteria	Verification	Validation
Definition	The process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase.	The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements.
Objective	To ensure that the product is being built according to the requirements and design specifications. In other words, to ensure that work products meet their specified requirements.	To ensure that the product actually meets the user's needs and that the specifications were correct in the first place. In other words, to demonstrate that the product fulfills its intended use when placed in its intended environment.
Question	Are we building the product *right*?	Are we building the *right* product?
Evaluation Items	Plans, Requirement Specs, Design Specs, Code, Test Cases	The actual product/software.

Activities	• Reviews • Walkthroughs • Inspections	• Testing

It is entirely possible that a product passes when verified but fails when validated. This can happen when, say, a product is built as per the specifications but the specifications themselves fail to address the user's needs.

- Trust but Verify.
- Verify but also Validate.

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, design and structure essential	No	Yes
Knowledge of Software Implementation essential	No	Yes
Who conducts this test on software	Software Testing Employee	Software Developer
baseline reference for tester	Requirements 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**

  implementing changes in existing or new requirements of user

- **Preventive**
  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.

Q. Which process model removes defects before software get into trouble?

A. Clean room software engineering method removes defects before software gets into trouble.

Q. How you can make sure that your written code which can handle various kinds of error situation?

A. I can write tests that define the expected error situations.

Q. Explain the differences between a Thread and a Process?

A. A process is instance of the computer program.In a single program it is possible to have one or more threads.

Q. Tell me the difference between an EXE and a DLL?

A. An exe is an executable program while a DLL is a file that can be loaded and executed by programs dynamically. It is an external code repository for programs. As both are different programs, reuse the same DLL instead of having that code in their file. It also reduces required storage space.

Q. What is strong-typing and weak-typing? Which is preferred? Why?

A. Strong typing checks the types of variables at compile time. On the other hand, weak typing checks the types of the system at run-time. Among them, Strong typing is always preferred because it minimizes the bugs.

Q. Describe the difference between Interface-oriented, Object-oriented and Aspect-oriented programming.

A.

- Interface programming is contract based.
- Object-oriented is a way to write granular objects which have a single purpose.

- Aspect Oriented Programming is to segregate the code in such a manner that various objects carry the main tasks, and the subsidiary tasks are carried by independent objects.

Q. Give me differences between object-oriented and component-based design?
A. Object-oriented design can easily be encapsulated to some degree in component-based design.

Q. What is essential for testing the quality of the code?
A. According to me, the unit testing framework is essential for testing the quality of the code.

Q. Do you think that maintenance of software is expensive?
A. According to me, maintenances of software will never be expensive if we are using proper development process.

Q. What is Agile?
A. The word 'agile' means −

- Able to move your body quickly and easily.
- Able to think quickly and clearly.

In business, 'agile' is used for describing ways of planning and doing work wherein it is understood that making changes as needed is an important part of the job. Business'agililty' means that a company is always in a position to take account of the market changes.

In software development, the term 'agile' is adapted to mean 'the ability to respond to changes − changes from Requirements, Technology and People.'

Q. What are the principles of Agile Software Development?
A.
1. Highest priority is to satisfy the customer through early and continuous delivery of business valuable software
2. Welcome changing requirements, even late in development
3. Deliver working software frequently
4. Business people and developers must work together with transparency on daily basis throughout the project
5. Build projects around motivated individuals
6. The best form of communication is to do face-to-face conversation
7. Working software is the primary measure of progress
8. Able to maintain a constant pace
9. Continuous attention to technical excellence
10. Simplicity – the art of maximizing the amount of work not done – is essential
11. Self-organizing teams
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

Q. How is agile methodology different than traditional waterfall process?

A. In agile methodology, features of the software are delivered frequently, so that the testing activity is done simultaneously with the development activity. Testing time is shortened as only small features are need be tasted at once.

While, in the waterfall model, testing activities take place at the end of the entire development process. Testing time, in this case, is as long as the entire product is to be tested in one go. Waterfall methodology is a closed process where all stakeholders are not involved in the development process whereas agile methodology requires an involvement of various stakeholders including customers.

Q. What is Extreme Programming?

A. XP is a lightweight, efficient, low-risk, flexible, predictable, scientific, and fun way to develop a software.

e**X**treme **P**rogramming (XP) was conceived and developed to address the specific needs of software development by small teams in the face of vague and changing requirements.

Extreme Programming is one of the Agile software development methodologies. It provides values and principles to guide the team behavior. The team is expected to self-organize. Extreme Programming provides specific core practices where −

- Each practice is simple and self-complete.
- Combination of practices produces more complex and emergent behavior.

Q. What is Agile Testing?

A. Agile Testing is a practice that a QA follows in a dynamic environment where testing requirements keep changing according to the customer needs. It is done parallel to the development activity where testing team receives frequent small codes from the development team for testing.

Q. Discuss the principles of agile testing.

A. There are some major points regarding agile testing and they are discussed below.

- Satisfaction of customer
- A big free clean mode
- Customers welcome changes
- The business people and the developers work together as a team
- It focuses on essence rather than lengthy documentation
- Face to face conversation is appreciated
- Promotion of sustainable development

Q. What do you mean by the disadvantages of the agile model?

There are certain disadvantages of the agile system which are discussed below.

- It is not easy to predict. If a big project is assigned you will have a problem to estimate the effort you need to put into that.
- Focusing on design and documentation is not always proper.
- The final product will not satisfy the customers if the guidelines given by the client are not properly understood.
- The high-level decision making is in the hands of the higher authorities which makes the fresher to have little knowledge.

Q. What is the right moment to use agile model?

There are certain methodologies and developments which can use agile like lean software development feature drove development, dynamic development and crystal methodologies.

Q. As a tester what should be your approach when requirements change continuously?

A. When requirement keeps changing, continuously agile tester should take following approach

- Write generic test plans and test cases, which focuses on the intent of the requirement rather than its exact details
- To understand the scope of change, work closely with the product owners or business analyst
- Make sure team understand the risks involved in changing requirements especially at the end of the sprint
- Until the feature is stable, and the requirements are finalized, it is best to wait if you are going to automate the feature
- Changes can be kept to a minimum by negotiating or implement the changes in the next sprint

Q. When do we use Agile Scrum Methodology?

A.

i. When the client is not so clear on requirements
ii. When the client expects quick releases
iii. When the client doesn't give all the requirements at a time

Q. What is a Sprint?

A. In Scrum, the project is divided into Sprints. Each Sprint has a specified timeline (2 weeks to 1 month). This timeline will be agreed by a Scrum Team during the Sprint Planning Meeting. Here, User Stories are split into different modules. The end result of every Sprint should be a potentially shippable product.

Q. Define the roles in Scrum ?

A. There are mainly three roles that a Scrum team have:

1. **Project Owner –** who has the responsibility of managing product backlog. Works with end users and customers and provide proper requirement to the team to build the proper product.
2. **Scrum Master –** who works with scrum team to make sure each sprint gets complete on time. Scrum master ensure proper work flow to the team.
3. **Scrum Team –** Each member in the team should be self-organized, dedicated and responsible for high quality of the work.

Q.Explain the difference between Extreme programming and Scrum?

A.

Scrum	Extreme Programing (XP)
– Scrum teams usually have to work in iterations called sprints which usually last up to two weeks to one month long	– XP team works in iteration that last for one or two weeks
– Scrum teams do not allow change into their sprints	– XP teams are more flexible and change their iterations
– In scrum, the product owner prioritizes the product backlog but the team decides the sequence in which they will develop the backlog items	– XP team work in strict priority order, features developed are prioritized by the customer
– Scrum does not prescribe any engineering practices	– XP does prescribe engineering practices