

Question: Explain the design principle of software Engineering.

Answer: - A particular area provided by design principle for the judgments of particular aspects of design. We have three types of principles which are explained below:

1. **Division of problems** - The base of these principles is to divide a big problem in to little parts. Every little part developed by different programs individually. Every little part can be individually altered.

- This helps the system to become more sufficient.
- This principle reduce the size of the problem and make simple and easy to service or maintenance.
- Leads to hierarchy in the design.

For the solution of big problem it is necessary to became proper coordination between these small pieces of problems.

2. **Abstraction** - To get the information in concerned to software parts from the outside is called the abstraction.

3. **Top down and bottom up design planning** - According to this principle a big problem divided in two little parts which is called modules and solved these modules one by one individually so that no one module can effected each other. We have two types of approaches. The top down approach goes from high level to the lower level. On the other side the bottom up approach goes the opposite that mean it goes lower level to top level.

- **Top down design planning** - When planning of system starts from that target which system wants to get then that approach is called top down design planning. When we see the desired task is not easy for achieving then this task divided in parts and these parts is called sub task. These sub tasks have some quality which is:

- Size of problem will be small
- Reduce the level of difficulty
- Easy to achieve

If a task is difficult then we may divide it low difficulty and easily gettable subtasks. Thus the process of division of various tasks in to sub tasks is to make simple and easy which can be used or solved easily. Many types of module based on this approach but this approach is useful only those that case where the target is mentioned clearly.

Benefits

- This method of program development is same to human approach of problem solving .In this approach first we decide the goal and after that we takes the steps to achieve the targets.
- The programmer became aware about the goal at every level.

- It reduces the level of confusion.
- It gives a proper process with the help of this problem solved easily and quickly.
- **Bottom up design planning** - To get the big goal for the system, this approach is used. It started from the lower level and at the end it reached the top level. In this approach individual modules are combined with each other so that a big module can be built which is the target of this system. A good idea is must require for the success of this approach. Until we have not good idea about the operation need at the higher level then we cannot decide what operation support at this time.

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 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.

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 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.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, 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 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.