

Software Engineering

DETAILED CONTENTS

1. Introduction to Software Engineering

Concept of systems: Types of systems : open, closed, static and dynamic systems.

Introduction, Programmes v/s Software Products

Emergence of Software Engineering- Early Computer Programming, High-level Language Programming, Control flow based Design, Data Structure Oriented Design, Object Oriented Design

2. Software Life Cycle Models

Requirement of Life Cycle Model, Classic Waterfall Model, Prototyping Model, Evolutionary Model, Spiral Model, introduction to agile methodology.

Comparison of different Life Cycle Models

3. Software Planning

Responsibilities of Software Project Manager

- Metrics for Project Size Estimation- LOC(Lines of Code), Function Point Metric

- Project estimation Techniques- Using COCOMO Model.

4. Requirement Analysis and Specification

Requirement gathering and Analysis, Software Requirement Specifications(SRS), Characteristics of good SRS

5. Software Design and Implementation

Characteristics and features of good Software Design Cohesion and Coupling, Software design Approaches- Function Oriented Design(Data flow diagrams, Data dictionary, Decision Trees and tables), Object Oriented Design, Structured Coding Techniques, Coding Styles, documentation

6. Software Testing

Concept of Testing, Verification v/s Validations, Unit Testing, Black Box Testing, White Box Testing, Integration testing, System testing, Configuration management.

Chapter 1 Introduction to Software Engineering

Short Questions

1. What is Software Engineering?

Ans:- Software Engineering is application of a systematic and disciplined approach to the development, operation and maintenance of software in an efficient and cost effective way.

2. What is the purpose of Software Engineering?

Or

Why is Software Engineering needed?

Ans:- The aim of Software Engineering is to improve the quality of software products. It is also used to increase the productivity in software development.

3. What is Program?

Ans:- Program is a set of instructions that are used to perform a particular task.

4. What is Software?

Ans:- Software is a set of programs in which each program performs a particular task.

5. What is Software Design?

Ans:- Software Design is a part of the software development process that lists specifications used in Software Engineering.

6. What are the levels of Software Design?

Ans:- (i) System design or External design (ii) Internal design or Detail design.

7. Define System Design or External Design.

Ans:- This design involves planning and specifying the external behavior of the software product.

8. Define Internal Design or Detailed Design.

Ans:- This design involves planning and specifying the internal structure of the software product.

9. What is software process?

Ans:- It specifies the abstract set of activities that need to be performed to get the final software product.

10. What is software project?

Ans:- Actual execution of the activities specified in the software process is called software project.

11. What is software product?

Ans:- The finally deliverable developed software is called software product.

12. What is algorithm?

Ans:- Algorithm is a step by step solution of a problem.

13. What is flow chart?

Ans:- A flow chart is a graphical representation of an algorithm.

14. What is Maintainability?

Ans:- Maintainability is the ease with which a product can be maintained in order to correct defects or improve performance.

15. What is High Level Programming?

Ans:- High Level Programming uses English like source code which is very easy to learn and use. Translators are used to convert High Level Language to Machine Language.

Long questions

1. Define Software Process, Software Project and Software Product.

Ans:- Software Process : It specifies the abstract set of activities that need to be performed to get the final software product.

Software Project: Actual execution of the activities specified in the software process is called software project.

Software Product: The finally deliverable developed software is called software product.

2. What is the difference between Program and Software?

Ans:-

Program	Software
1. Program is a set of instructions that are used to perform a particular task.	Software is a set of programs in which each program performs a particular task.
2. In programs, very little documentation is required.	Software products must be well documented.
3. Programs are smaller in size.	Software products are very large in size.
4. Programs are developed by individuals.	Software products are developed by a team.

3. What is Software Design? Explain its levels.

Ans:- Software Design is a part of the software development process that lists specifications used in Software Engineering.

There are two levels of Software Design - (i) System design or External design (ii) Internal design or Detail design.

(i) System design or External design :- It is focus on deciding which modules are needed. It also specifies the interface among the modules. This design involves planning and specifying the external behaviour of the software product.

(ii) Internal design or Detail design :- This design involves planning and specifying the internal structure of the software product. This design includes more detail of processing logic and data structure.

4. Describe Control Flow based design.

Ans:- To help the programmer to design program having good control flow structure, flow-charting technique is used. In flow-charting technique, the algorithm is represented using flowchart. A flowchart is a graphical representation that arranges the sequence of operations to be carried out to perform a particular task. In Control Flow based design, goto statement should be avoided.

5. Describe Data Structure oriented design.

Ans:- There are some techniques that emphasize on designing the data structure and on deriving program structure from it. These techniques are called Data Structure oriented design.

6. Describe Object Oriented Design.

Ans:- Object Oriented Design not only includes the best features of Structured programming but also some new and powerful features such as Encapsulation, Abstraction, Inheritance and Polymorphism etc. Its salient features are :-

i) Simplicity

- ii) Reusability
- iii) More robust
- iv) Easy maintenance
- v) Lower development cost
- vi) Lower development time

7. What is software maintenance? Explain the types of maintenance in Software Engineering.

Ans:- Software maintenance is the modification of a software product to correct faults, to improve performance or other attributes.

Software Maintenance falls into the following categories:

- 1. Adaptive** – In this type of maintenance, some modification is being done in the system to keep it compatible with the changing environments.
- 2. Perfective** – Checks for fine tuning of all elements of the system, functionalities and abilities to improve system performance.
- 3. Corrective** – Detecting bugs & errors in the existing solution and fixing them to make the system works efficiently.
- 4. Preventive** – Preventive software maintenance help in preventing the system from any upcoming vulnerabilities.

Very Long Questions

1. What is software design? Explain the levels and types of software design.

Ans:- Software Design is a part of the software development process that lists specifications used in Software Engineering.

There are two levels of Software Design - (i) System design or External design (ii) Internal design or Detail design.

- (i) System design or External design :- It is focus on deciding which modules are needed. It also specifies the interface among the modules. This design involves planning and specifying the external behaviour of the software product.
- (ii) Internal design or Detail design :- This design involves planning and specifying the internal structure of the software product. This design includes more detail of processing logic and data structure.

There are three types of software design which are explained as below:-

- (a) Control Flow based design :- To help the programmer to design program having good control flow structure, flow-charting technique is used. In flow-charting technique, the algorithm is represented using flowchart. A flowchart is a graphical representation that arranges the sequence of operations to be carried out to perform a particular task. In Control Flow based design, goto statement should be avoided.
- (b) Data Structure oriented design :- There are some techniques that emphasize on designing the data structure and on deriving program structure from it. These techniques are called Data Structure oriented design.
- (c) Object Oriented Design :- Object Oriented Design not only includes the best features of Structured programming but also some new and powerful features such as Encapsulation, Abstraction, Inheritance and Polymorphism etc.

Chapter 2

Software Life Cycle Models

Short Questions

1. What is SDLC (Software Development Life Cycle) or Software Life Cycle?

Ans:- A software life cycle is often called a Software Development Life Cycle (SDLC). It is the period conceived and ends when the product is no longer available for use.

2. What is Waterfall Model?

Ans:- This is the basic or fundamental model. It is a sequential or linear software development model.

3. What is SRS?

Ans:- The exact requirements of the customer are analyzed and a proper documentation mentioning the requirements is prepared. This document is called SRS (Software Requirements Specification).

4. What is Acceptance Testing?

Ans:- Acceptance Testing is system testing performed by the customer himself.

5. Write the types of maintenance.

Ans:- Corrective Maintenance, Preventive Maintenance, Perfective Maintenance, Adaptive Maintenance.

6. What is the purpose of coding phase?

Ans:- The purpose of coding phase is to translate the software design into source code.

7. What is financial feasibility?

Ans:- It is tested that if the project can be completed under the available resources or not or if the finance is available to fulfill the requirements.

8. What is technical feasibility?

Ans:- In this feasibility, it is tested that if the required technology for the project is available in the market or not.

9. What is system design or external design?

Ans:- This design involves planning and specifying the external behavior of the software product.

10. What is detailed design or internal design?

Ans:- This design involves planning and specifying the internal structure of the software product.

11. What is Unit Testing?

Ans:- The purpose of Unit Testing is to test each module separately from other module.

12. What is Integration Testing?

Ans:- In Integration Testing, modules are integrated one by one in step and after each addition of one module, the resulting system is tested again and again.

13. What is System Testing?

Ans:- The purpose of System Testing is to ensure that the developed software meets the requirements specified in the SRS document.

14. What is software maintenance?

Or

Why software maintenance is required?

Ans:- Software maintenance is the modification of a software product to correct faults, to improve performance or other attributes.

15. Explain the requirement analysis and specification.

Ans:- During this phase, the exact requirements of the customer are analyzed and a proper documentation mentioning the requirements is prepared. This document is called SRS (Software Requirements Specification).

16. What is Prototype model?

Ans:- The main purpose of prototype model is to illustrate to the customer about the format of input data, format of output data, format of reports etc. A prototype is like a toy of the actual system.

17. What is Evolutionary Model?

Ans:- In this model, system is broken down into several modules or small units that can be implemented incrementally one after another and delivered successively.

18. Describe Spiral Model.

Ans:- In the Spiral Model, whole process of the software development is divided into four phases. These four phases of development are called quadrants of the spiral.

19. What is Testability?

Ans:- Testability refers to the ability of a software to test it properly for finding bugs.

Long Questions

1. Explain the feasibility study.

Ans:- Feasibility Study :- During this phase, we generally study two types of feasibility :

- (i) Financial Feasibility:- It is tested that if the project can be completed under the available resources or not or if the finance is available to fulfil the requirements.
- (ii) Technical Feasibility:- In this feasibility, it is tested that if the required technology for the project is available in the market or not.

2. Explain the requirement analysis and specification.

Ans:- During this phase, the exact requirements of the customer are analyzed and a proper documentation mentioning the requirements is prepared. This document is called SRS (Software Requirements Specification). SRS generally includes the following types of requirements:

- (i) Functional Requirements
- (ii) Non-functional requirements

3. Explain the software design with its levels.

Ans:- Software Design is a part of the software development process that lists specifications used in Software Engineering.

There are two levels of Software Design - (i) System design or External design (ii) Internal design or Detail design.

- (i) System design or External design :- It is focus on deciding which modules are needed. It also specifies the interface among the modules. This design involves planning and specifying the external behaviour of the software product.

- (ii) Internal design or Detail design :- This design involves planning and specifying the internal structure of the software product. This design includes more detail of processing logic and data structure.

4. Explain Waterfall model.

Ans:- This is the basic or fundamental model. It is a sequential or linear software development model. The user requirements must be complete for this project model.

This model divides the software development life cycle into following phases:-

- (a) Feasibility Study
- (b) Requirements Analysis and Specification
- (c) Design
- (d) Coding and Unit Testing
- (e) Integration and System Testing
- (f) Maintenance

5. Explain Prototype Model.

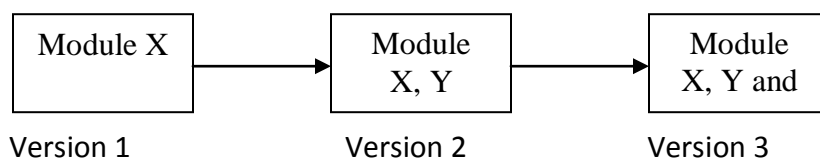
Ans:- In the prototyping model, a working prototype of the system is built first. The main purpose of this prototype is to illustrate to the customer about the format of input data, format of output data, format of reports etc. A prototype is like a toy of the actual system. Using prototype, the opinion of the user can be easily and more clearly obtained.

The basic phases of the development in this model are same as in the waterfall model with an additional phase of preparing a prototype.

6. Explain Evolutionary Model.

Ans:- This model is a version based development model. In this model, after each cycle of development a new version of software product is developed. In this model, system is broken down into several modules or small units that can be implemented incrementally one after another and delivered successively.

In this model, a core module of the system is developed initially and then it is refined by adding new functionalities.



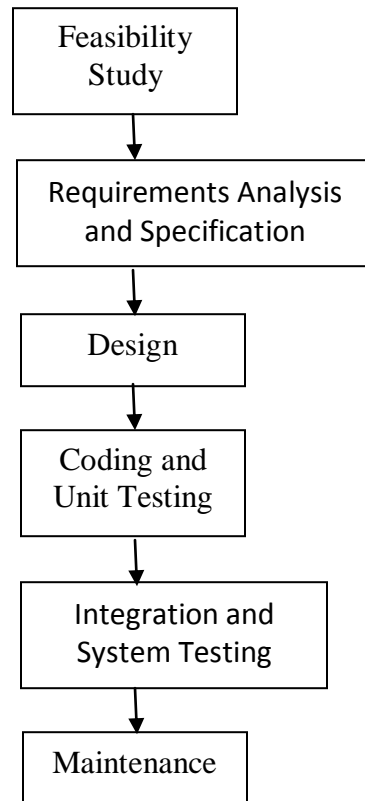
Very Long Questions

1. Explain Classical Waterfall Model.

Ans:- This is the basic or fundamental model. It is a sequential or linear software development model. The user requirements must be complete for this project model.

This model divides the software development life cycle into following phases:-

- (a) Feasibility Study
- (b) Requirements Analysis and Specification
- (c) Design
- (d) Coding and Unit Testing
- (e) Integration and System Testing
- (f) Maintenance



- (a) Feasibility Study :- During this phase, we generally study two types of feasibility :
- (i) Financial Feasibility:- It is tested that if the project can be completed under the available resources or not or if the finance is available to fulfil the requirements.
 - (ii) Technical Feasibility:- In this feasibility, it is tested that if the required technology for the project is available in the market or not.
- (b) Requirement Analysis and Specification:- During this phase, the exact requirements of the customer are analysed and a proper documentation mentioning the requirements is prepared. This document is called SRS (Software Requirements Specification). SRS generally includes the following types of requirements:
- (i) Functional Requirements
 - (ii) Non-functional requirements
- (c) Design:- Software Design is a part of the software development process that lists specifications used in Software Engineering.
There are two levels of Software Design - (i) System design or External design (ii) Internal design or Detail design.
- (i) System design or External design :- It is focus on deciding which modules are needed. It also specifies the interface among the modules. This design involves planning and specifying the external behaviour of the software product.
 - (ii) Internal design or Detail design :- This design involves planning and specifying the internal structure of the software product. This design includes more detail of processing logic and data structure.
- (d) Coding and Unit Testing:- The purpose of coding phase is to translate the software design into source code. The coding phase is also sometimes called implementation

phase. The purpose of Unit Testing is to test each module separately from other module.

(e) Integration and System Testing:- In Integration Testing, modules are integrated one by one in step and after each addition of one module, the resulting system is tested again and again. The purpose of System Testing is to ensure that the developed software meets the requirements specified in the SRS document.

(f) Maintenance:- Maintenance of a software product requires more effort than the effort necessary to develop the software product itself.

Advantages :- (i) It is simple to use.

(ii) It is simple to understand.

(iii) Required amount of resources are minimal.

(iv) It works well for smaller projects.

Disadvantages :- (i) The requirements must be complete for this project model at initial stage.

(ii) Small changes may cause a lot of problems.

2. Explain Prototype Model.

Ans:- In the prototyping model, a working prototype of the system is built first. The main purpose of this prototype is to illustrate to the customer about the format of input data, format of output data, format of reports etc. A prototype is like a toy of the actual system. Using prototype, the opinion of the user can be easily and more clearly obtained.

The basic phases of the development in this model are same as in the waterfall model with an additional phase of preparing a prototype.

Advantages :- (i) Users are actively involved in the development.

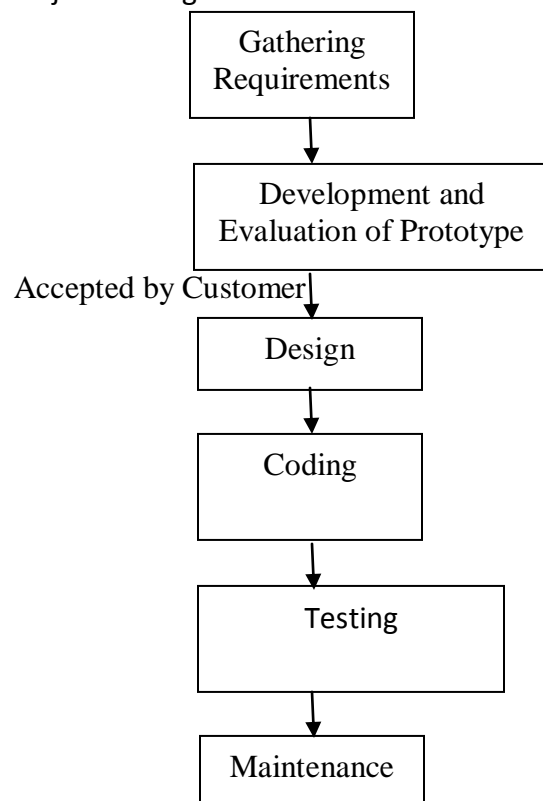
(ii) Increased development speed.

(iii) Reduced development cost.

(iv) Easy to understand for users.

Disadvantages :- (i) Not suitable for large applications.

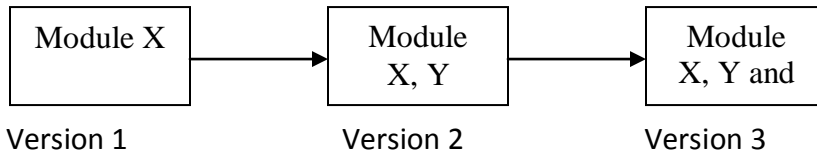
(ii) Project Management difficulties.



3. Explain Evolutionary Model.

Ans:- This model is a version based development model. In this model, after each cycle of development a new version of software product is developed. In this model, system is broken down into several modules or small units that can be implemented incrementally one after another and delivered successively.

In this model, a core module of the system is developed initially and then it is refined by adding new functionalities.



Version 1

Version 2

Version 3

Advantages :- (i) It is easy to test each version.

(ii) The exact requirements of the user can be easily included in it.

Disadvantages :- (i) It is used for large projects only.

4. Explain Spiral Model.

Ans:- This is the modern model. In the Spiral Model, whole process of the software development is divided into four phases. After completion of a set of four phases, it is evaluated by the customer. This process continues till the final software product. These four phases of development are called quadrants of the spiral.

Advantages :- (i) Risk Analysis and Management.

(ii) Better productivity.

(iii) Errors are removed in early phase of software development.

Disadvantages :- (i) Success is dependent on risk analysis.

(ii) This is very costly model.

Chapter 3 Software Planning

Short Questions

1. What is the objective of project planning?

Ans:- The objective of software project planning is to provide a framework that enables the project manager to make reasonable estimate of resources, cost and schedule.

2. Describe any two responsibilities of a software project manager.

Ans:- (i) Understanding the client's requirements
(ii) Estimation of effort, cost and project duration

3. What is project size estimation?

Ans:- Project size estimation is an activity in Software Engineering that is used to estimate the size of a software application.

4. What is empirical estimation technique?

Ans:- In this technique, the project parameters are estimated using the past working experience.

5. What is heuristic estimation technique?

Ans:- In this technique, the project parameters are estimated using some mathematical expressions. Example :- COCOMO model.

6. What is analytical estimation technique?

Ans:- In this technique, the project parameters are estimated using some scientific assumptions. Example :- Halstead's Software Science.

7. What is LOC (Lines of Code)?

Ans:- This metric measures the size of a project by counting the number of instructions in the source code of program.

8. What is FP (Function Point) metric?

Ans:- This metric describes that the size of software product is directly dependent on the number and types of different functions it performs.

9. What is COCOMO Model?

Ans:- It is a cost estimation technique and uses bottom-up approach.

10. Write the three stages of COCOMO Model.

Ans:- There are three stages of COCOMO Model:-

- (i) Basic COCOMO
- (ii) Intermediate COCOMO
- (iii) Complete COCOMO

Long Questions

1. Describe different activities involved in project planning.

Ans:- Planning is one of the most important management activities. Various activities that are done during project planning are as follows:-

- (i) Risks Identification and Abatement
- (ii) Quality Assurance
- (iii) Configuration Management Plan
- (iv) Estimation of effort, cost and project duration

- (v) Project Scheduling
- (vi) Project Complexity
- (vii) Selecting Software Development process model
- (viii) Recruitment and Training

2. Describe the responsibilities of software project manager.

Ans:-

- (i) Risk Analysis
- (ii) Understanding the client's requirements
- (iii) Estimation of effort, cost and project duration
- (iv) Making sure that the clients requirement are met
- (v) Making sure that the project is completed on time
- (vi) Making sure that the project is completed within budget
- (vii) Making sure that quality standards are met.
- (viii) Keeping track of software project progress.

3. Explain Project Estimation Techniques.

Ans:- There are following three techniques which are used for estimating project parameter.

Empirical Estimation:- In this technique, the project parameters are estimated using the past working experience.

Heuristic Technique:- In this technique, the project parameters are estimated using some mathematical expressions. Example :- COCOMO model.

Analytical Estimation Technique:- In this technique, the project parameters are estimated using some scientific assumptions. Example :- Halstead's Software Science.

Very Long Questions

1. Explain LOC (Lines of Code) in brief.

Ans:- LOC (Lines of Code) is a project size estimation activity. It is the simplest among all the metrics. This metric measures the size of a project by counting the number of instructions in the source code of program. While counting the instructions, Comments and Header lines are ignored.

Advantages :- (i) LOC is very simple to use.

- (ii) Lines of code can be easily counted without having the knowledge of software source code.

Disadvantages :- (i) It is difficult to estimate LOC at beginning of project.

- (ii) LOC measures are programming language dependent.
- (iii) LOC depends on the coding style of different programmers.
- (iv) LOC can't guess the quality of the software.
- (v) LOC can't guess the functionality of the software.

2. Explain FP(Function Point) Metric in brief.

Ans:- Function-oriented metrics are dependent on the functionality delivered by the application. This approach describes that the size of software product is directly dependent on the number and types of different functions it performs.

Following five different characteristics are used to compute the Function Points (FP) of the software product:

- (i) Number of user inputs
- (ii) Number of outputs
- (iii) Number of files
- (iv) Number of user inquiries
- (v) Number of External Interfaces

Advantages :- (i) The size of software product is directly dependent on the number and types of different functions it performs.

(ii) This metric is language-independent.

Disadvantages :- (i) This metric does not take into account the complexity of a software.

(ii) This metric is dependent on the engineers working on the project.

3. Explain COCOMO Model in brief.

Ans:- COCOMO (Constructive Cost Estimation Model) uses the Heuristic technique. It was proposed by Boehm in 1981. It is a cost estimation technique and uses bottom-up approach. There are three stages of COCOMO Model:-

- (iv) Basic COCOMO
- (v) Intermediate COCOMO
- (vi) Complete COCOMO

According to Boehm, software development projects can be classified into three categories based on their complexity – organic, semidetached and embedded.

Organic :- These types of projects are small in size. They are easy to use and understand. The size of development team is small.

Semidetached :- These types of projects fall between organic and embedded categories. The team is a mixture of experienced and non-experienced persons.

Embedded :- These types of projects require hardware interaction. These projects are very complicated projects.

(i) **Basic COCOMO Model :-** This is the most common approach for estimating effort. In this stage, a single variable called “Project Size” is used. In this model, the following equations are used:-

$$(a) E = a * (KLOC)^b PM$$

where E is the effort, PM means Person Months

KLOC means One Thousand Lines of Code

a and b are constants.

$$(b) T = c * (Effort)^d Months$$

Where T is the development time

c and d are constants.

(ii) **Intermediate COCOMO Model :-** This model uses number of factors for estimation. It finds the initial estimates using the basic COCOMO model. Then Adjustment Factor is calculated using the number of factors. This Adjustment Factor is then multiplied to initial effort calculated to give the final estimated effort.

(iii) **Complete COCOMO Model :-** Most of the large systems are made up of several smaller subsystems. These subsystems may have different characteristics, different complexities and the different reliability requirements. So, this model is beneficial to find estimation for individual subsystems.

Chapter 4 Requirement Analysis and Specification

Short Questions

1. What is requirement gathering?

Ans:- Requirement Gathering is the process of gathering the information about the requirements of the customer which customer wants in his system or project to be developed.

2. What is Software Requirement Analysis or Requirement Engineering?

Ans:- Requirement Engineering is process of determining user expectations for a new product. It is a method to find out the requirement from the user to develop software. It helps to understand, interpret, classify and organize the software requirement.

3. Describe any two techniques to gather requirement from the user.

Ans:- Brainstorming:- The brainstorming involves both idea generation and reduction.

Questionnaires:- Questionnaires are used when we have to gather information from thousands of people.

4. Write the basic components of a SRS document.

Ans:- (i) Functional Requirements

(ii) Performance requirements

(iii) Design Constraints

(iv) External Interface Requirements

5. Describe any two characteristics of a good SRS.

Ans:- Verifiable :- Every requirement specified in SRS must be verifiable.

Consistent :- SRS is said to be consistent if there is no requirement specified in SRS that conflicts with another requirement specified in SRS.

Long Questions

1. Which techniques are used to gather requirement from the user?

Ans:-

(i) Interview :- An interview is generally conducted by an experienced analyst.

(ii) Brainstorming :- The brainstorming involves both idea generation and reduction.

(iii) Questionnaires :- Questionnaires are used when we have to gather information from thousands of people.

(iv) Prototyping :- In the prototyping model, a working prototype of the system is built first. The main purpose of this prototype is to illustrate to the customer about the format of input data, format of output data, format of reports etc. A prototype is like a toy of the actual system. Using prototype, the opinion of the user can be easily and more clearly obtained.

(v) Group Interview :- More than one person are interviewed in Group Interview. These interviews work well when everyone is at same level or has the same role.

2. Explain Software Requirement Specification (SRS).

Ans:- Requirement Analysis is process of determining user expectations for a new product. It is a method to find out the requirement from the user to develop software. It helps to understand, interpret, classify and organize the software requirement.

While during the requirement specification phase, the requirements identified are documented in a document called Software Requirement Specification (SRS).

SRS document acts as a contract between the software engineer and the customer.

Components of SRS :- The basic components of an SRS document are :

- (i) Functional Requirements
- (ii) Performance requirements
- (iii) Design Constraints
- (iv)** External Interface Requirements

Very Long Questions

1. Explain the characteristics of a good SRS.

Ans:- A good SRS should have the following basic properties :-

- (i) Correctness :- It ensures that what is specified in SRS is done correctly.
- (ii) Completeness :- SRS is said to be complete if everything the software is supposed to do is specified in the SRS.
- (iii) Verifiable :- Every requirement specified in SRS must be verifiable.
- (iv) Consistent :- SRS is said to be consistent if there is no requirement specified in SRS that conflicts with another requirement specified in SRS.
- (v) Unambiguous :- SRS is said to be unambiguous if all the requirements specified in it have one and only one meaning.
- (vi) Modifiable :- SRS should be easy to modify.
- (vii) Traceable :- Origin of each of the requirements must be clear.

Chapter 5 Software Design and Implementation

Short Questions

1. Explain the cohesion.

Ans:- Cohesion is the indication of the relationship within module. It is concept of intra-module. Cohesion has many types but usually highly cohesion is good for software.

2. Explain the coupling.

Ans:- Coupling is the indication of the relationships between modules. It is concept of Inter-module. Coupling has many types but usually low coupling is good for software.

3. What is DFD (Data Flow Diagram)?

Ans:- A data flow diagram (DFD) maps out the flow of information for any process or system.

1. What does Structured Programming mean?

Ans:- It means that the code will execute the instruction by instruction one after the other.

2. What is Data Dictionary?

Ans:- Data dictionaries are simply repositories to store information about all data items.

3. Write any two characteristics of a good software design.

Ans:- Cohesion has many types but usually highly cohesion is good for software.

Coupling has many types but usually low coupling is good for software.

4. Explain the function oriented design.

Ans:- Function Oriented Design is an approach to software design where the design is decomposed into a set of interacting units where each unit has a clearly defined function.

5. Explain the object oriented design.

Ans:- Object Oriented Design (OOD) is a part of the object oriented programming (OOP). It is mainly the process of using an object methodology to design a computing system or application.

6. What is Data Coupling?

Ans:- It is the best coupling. In this type of coupling, data or information is passed from one module to another module.

7. What is Content Coupling?

Ans:- It is the worst coupling. In this type of coupling, One module changes the data of another module.

8. What is Functional Cohesion?

Ans:- It is the best cohesion. In this type of cohesion, two or more elements within a module perform the same function.

9. What is Coincidental Cohesion?

Ans:- It is the worst cohesion. unplanned or random cohesion is called coincidental cohesion.

10. What is Decision Table?

Ans:- Decision table is a brief visual representation for specifying which actions to perform depending on given conditions.

11. What is the need of documentation in Software Engineering?

Ans:- The documentation helps keep track of all aspects of an application and it improves the quality of a software product.

Long Questions

1. Explain Structured Programming Techniques.

Ans:- Structured Programming Approach can be defined as a programming approach in which the program is made as a single structure. It means that the code will execute the instruction by instruction one after the other. It doesn't support the possibility of jumping from one instruction to some other with the help of any statement like GOTO, etc. Therefore, the instructions in this approach will be executed in a serial and structured manner.

The structured program consists of well structured and separated modules, but the entry and exit in a Structured program is a single-time event. It means that the program uses single-entry and single-exit elements.

2. Explain coding styles in Software Engineering.

Ans:- (i) Indentation is useful in making the code easily readable.

(ii) All the codes should be properly commented.

(iii) All curly braces should start from a new line.

(iv) Goto statements should be avoided.

(v) All the variables, functions, and methods should be assigned names that make the code more understandable to the reader.

(vi) Short and clear names should be assigned in place of long names.

3. Differentiate between coupling and cohesion.

Ans:-

Cohesion	Coupling
Cohesion is the concept of intra module.	Coupling is the concept of inter module.
Cohesion represents the relationship within module.	Coupling represents the relationship between modules.
Increasing in cohesion is good for software.	Increasing in coupling is avoided for software.
Cohesion represents the functional strength of modules.	Coupling represents the independence among modules.
In cohesion, module focuses on the single thing.	In coupling, modules are connected to the other modules.

Very Long Questions

1. Explain the types of coupling.

Or

State the classification of coupling.

Ans:- (i) Data :- It is the best coupling. In this type of coupling, data or information is passed from one module to another module.

- (ii) Stamp :- In this type of coupling, same data structure is shared between two modules.
- (iii) Control :- In this type of coupling, flags are passed from one module to another module.
- (iv) Common :- In this type of coupling, a global data is shared between two modules.
- (v) Content :- It is the worst coupling. In this type of coupling, One module changes the data of another module.

2. Explain the types of cohesion.

Or

State the classification of cohesion.

Ans:- (i) Functional :- It is the best cohesion. In this type of cohesion, two or more elements within a module perform the same function.

- (ii) Sequential :- In this type of cohesion, output of one element is input of another element.
- (iii) Communicational :- In this type of cohesion, two or more elements share the same data.
- (iv) Procedural :- In this type of cohesion, two or more tasks are performed in a sequence (step by step).
- (v) Temporal :- In this type of cohesion, many tasks are performed within a time moment.
- (vi) Logical :- In this type of cohesion, many elements belong to a particular logical class or function.
- (vii) Coincidental :- It is the worst cohesion. unplanned or random cohesion is called coincidental cohesion.

3. What are the characteristics of a good software design?

Ans:- A good design should have the following basic properties :-

- (i) Correctness :- a design should meet the user requirements.
- (ii) Completeness :- a design should be complete in all respects according to user requirements.
- (iii) Efficiency :- a design should be efficient.
- (iv) Consistent :- a design is said to be consistent if there is no module that conflicts with another module.
- (v) Maintainability :- a design should be easy to maintain according to the changing requirements.
- (vi) Strongly Cohesive :- Cohesion is the indication of the relationship within module. It is concept of intra-module. Cohesion has many types but usually highly cohesion is good for software.
- (vii) Weakly Coupled :- Coupling is the indication of the relationships between modules. It is concept of Inter-module. Coupling has many types but usually low coupling is good for software.

4. Explain the function oriented design.

Ans:- Function Oriented Design is an approach to software design where the design is decomposed into a set of interacting units where each unit has a clearly defined function.

This is also called Top-Down Structure.

Function Oriented Design Strategies are as follows:-

Data Flow Diagram (DFD):

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

Data Dictionaries:

Data dictionaries are simply repositories to store information about all data items defined in DFDs. Data dictionaries include Name of the item, Aliases (Other names for items), Description / purpose, Related data items, Range of values, Data structure definition.

Decision Tables and Decision Trees:

Decision table is a brief visual representation for specifying which actions to perform depending on given conditions. The information represented in decision tables can also be represented as decision trees.

5. Explain the object oriented design.

Ans:- Object Oriented Design (OOD) is a part of the object oriented programming (OOP). It is mainly the process of using an object methodology to design a computing system or application. This technique enables the implementation of software based on the concepts of objects.

The other characteristics of Object Oriented Design are as follows:

- (i) Objects are abstractions of the real-world entities.
- (ii) The objects are independent and in an encapsulated state and representation information.
- (iii) System functionality is expressed in terms of object services.
- (iv) Communication between objects is through message passing.
- (v) The objects may be distributed and may execute sequentially or in parallel.

Chapter 6 Software Testing

Short Questions

1. What is Software Testing?

Ans:- Testing is process of executing a program with intent of finding error.

2. What is Alpha Testing?

Ans:- Alpha testing refers to the system testing carried out by the testing team within developing organization.

3. What is Beta Testing

Ans:- Beta testing is done by the end user at the user side.

4. What is Verification?

Ans:- Verification is the process that makes it sure that the software product is developed in the right way.

5. What is Validation?

Ans:- It is a process of finding out if the product being built is right.

6. What is the Black Box Testing?

Ans:- Black Box Testing is used to test the software without knowing the internal structure of code of the software. It is also known as functional testing.

7. What is the White Box Testing?

Ans:- White Box Testing is a software testing method that examines the program structure and derives test data from the program logic.

8. What is Unit Testing?

Ans:- Unit Testing is process of testing a module. Unit Testing comprises the set of tests performed by an individual programmer.

9. What is Integration Testing?

Ans:- It is a phase in software testing in which individual modules are combined and tested as a group.

10. What is System Testing?

Ans:- System testing is testing of behavior of a complete and fully integrated software product based on the Software Requirement Specification (SRS) document.

11. What is User Acceptance Testing?

Ans:- Acceptance Testing is system testing performed by the customer himself.

12. What are test cases?

Ans:- A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly.

Long Questions

1. Explain the verification.

Ans:- The standard definition of verification is “Are we building the product right?”. Verification is the process that makes it sure that the software product is developed in the right way. As the product development goes through different stages, an analysis is done to ensure that all required specifications are met during the verification.

Inspection:- An inspection team formally reviews the documents and software product.

Walk through:- During the walk through meeting, the presenter introduces the material to all the participants.

2. Explain the validation.

Ans:- It is a process of finding out if the product being built is right. A software product should satisfy all the functional requirements. Validation process starts after the Verification process ends. Unit testing, Integration testing, functional testing and Acceptance testing come under Validation process.

3. Explain the Black Box Testing.

Ans:- Black Box Testing is used to test the software without knowing the internal structure of code of the software. It is also known as functional testing.

Black Box Testing Techniques :- The following are main techniques to black box testing :-

- (i) Equivalence class partitioning
- (ii) Boundary value analysis
- (iii) Cause effect graphs
- (iv) Comparison testing

4. Explain the White Box Testing.

Ans:- White Box Testing is a software testing method that examines the program structure and derives test data from the program logic. White box testing requires the intimate knowledge of program internals.

White Box Testing Techniques :- The following are main techniques to white box testing :-

- (i) Basis path testing
- (ii) Structured testing
- (iii) Statement coverage testing
- (iv) Branch coverage testing
- (v) Condition coverage

5. Explain Software Configuration Management (SCM).

Ans:- Software configuration management (SCM) is the discipline for systematically controlling the changes that take place during development. Software configuration management is a process independent of the development process largely because most development models cannot accommodate change at any time during development. SCM has the following three main components:

- (i) Software configuration identification
- (ii) Change control
- (iii) Status accounting and auditing

Very Long Questions

1. Explain the levels of software testing.

Ans:- Software testing has the following levels:-

Unit Testing :- Unit Testing is process of testing a module. Unit Testing comprises the set of tests performed by an individual programmer.

Integration Testing :- It is a phase in software testing in which individual modules are combined and tested as a group.

System Testing :- System testing is testing of behavior of a complete and fully integrated software product based on the Software Requirement Specification (SRS) document.

Acceptance Testing :- Acceptance Testing is system testing performed by the customer himself.