QUESTION BANK

(Software Engineering)

Q.1 Define Software Engineering.

Ans. Software Engineering is defined as the application of systematic, disciplined, quantified approach to the development, operations, and maintenance of software.

Q.2 List out the elements in Computer-Based System?

Ans. Elements in Computer-Based System are:

□ Software □ Hardware

□ People □ Database

□ Documentation □ Procedures.

Q.3 What are the factors to be considered in the System Model Construction?

Ans. Factors to be considered in the System Model Construction are:

□ Assumption □ Simplification

□ Limitation □ Constraints

□ Preferences

Q.4 What does a System Engineering Model accomplish?

Ans. 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

□ Represent all Linkages that enable engineer to better understand view.

Q.5 Define Framework.

Ans. Framework is the Code Skeleton that can fleshed out with specific classes or functionality and Designed to address specifies problem at hand.

Q.6 What are the important roles of Conventional Component within the Software Architecture?

Ans. The important roles of Conventional component within the Software Architecture are:

- **Control Component:** That coordinates invocation of all other problem domain.
- **Problem Domain Component:** That implements Complete or Partial function required by customer.
- **Infrastructure Component:** That responsible for functions that support processing required in problem domain.

Q.7 Differentiate Software Engineering methods, tools and procedures.

Ans. **Methods:** Broad array of tasks like project planning cost estimation etc.

Tools: Automated or semi automated support for methods.

Procedures: Holds the methods and tools together. It enables the timely development of computer software.

Q.8 Who is called as the Stakeholder?

Ans. Stakeholder is anyone in the organization who has a direct business interest in the system or product to be built.

Q.9 Write about Real Time Systems.

Ans. It provides specified amount of computation with in fixed time intervals. RTS sense and control external devices, respond to external events and share processing time between tasks.

Q.10 Define Distributed system.

Ans. It consists of a collection of nearly autonomous processors that communicate to achieve a coherent computing system.

Q.11 What are the characteristics of the software?

Ans. 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.

Q.12 What are the various categories of software?

Ans. **T**he various categories of software are:

- System software Application.
- Software Engineering / Scientific.
- Software Embedded software.
- Web Applications.
- Artificial Intelligence software.

Q.13 What are the challenges in software?

Ans. The challenges in software are:

- Copying with legacy systems.
- Heterogeneity challenge.
- Delivery times challenge.

Q.14 Define Software process.

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

Q.15 What are the fundamental activities of a software process?

Ans. The fundamental activities of a software process are:

- Specification
- Design and Implementation
- Validation
- Evolution

Q.16 What are the umbrella activities of a software process?

Ans. The umbrella activities of a software process are:

- Software project tracking and control.
- Risk Management.
- Software Quality Assurance.
- Formal Technical Reviews.
- Software Configuration Management.
- Work product preparation and production.
- Reusability management, Measurement.

Q.17 List the activities during project Initiation.

Ans. Important activities during project initiation phase:

- Management team building.
- Enables the team members to understand one another.
- Minimize the impact of cultural and language barriers.
- Scope and high level work division agreements.
- Management reporting and escalating procedures.
- Involvement of infra structure / support groups.
- Team formation.
- Project kick off meeting is attended by formally all concerned so that everyone has a common understanding of what is expected.

Q.18 What is work breakdown structure?

Ans. Work breakdown structure is the decomposition of the project into smaller and more manageable parts with each part satisfying the following criteria-

- Each WBS unit has a clear outcome.
- The outcome has a direct relationship to achieve the overall project goal.
- Each point has single point of accountability.

Q.19 What are the issues that get discussed during project closure?

Ans. The issues that get discussed during project closure are:

- What were the goals that we set out to achieve?
- How effective were the in process metrics?
- What were the root causes for under-achievement or over achievement?
- Was our estimation effort correct?
- What were the factors in the environment that would like to change?
- What did we gain from the system or environment?
- Was our estimation of the hardware correct?

Q.20 Give any two activities of project initiation.

Ans. Management team building and Team formation.

Q.21. What are the external dependencies in project planning?

Ans. Staffing, Training, Acquisition and Commissioning of new hardware, Availability of modules, Travel.

Q.22. What are internal milestones?

Ans. They are the measurable and quantifiable attributes of progress. They are the intermediate points in the project which ensure that we are in the right track. They are under the control of project manager.

Q.23. What is the role of the project board?

Ans. The overall responsibility for ensuring satisfaction progress on a project is the role of the project board.

Q.24. What is the role of project manager?

Ans. The project manager is responsible for day to day administration of the project.

Q.25. What is closed system?

Ans. Closed systems are those that do not interact with the environment.

Q.26. What is embedded system?

Ans. A system that is a part of a large system whose primary purpose is non computational.

Q.27. What is a Process Framework?

Ans. Process Framework establishes foundation for a complete software process by identifying a small number of framework activities that are applicable for all software projects regardless of their size and complexity.

Q.28 What are the Generic Framework Activities?

Ans. Generic Framework Activities are:

- Communication.
- Planning.
- Modeling.
- Construction.
- Deployment.

Q.29. Define Stakeholder.

Ans. Stakeholder is anyone who has stake in successful outcome of project such as:

- Business Managers,
- End-users,
- Software Engineer,
- Support People

Q.30. How the Process Model differ from one another?

Ans. Process Model differ from one another due to the following reasons:

- Based on flow of Activities.
- Interdependencies between Activities.
- Manner of Quality Assurance.

- Manner of Project Tracking.
- Tam Organization and Roles.
- Work Products identify a requirement Identifier.

Q.31 Write out the reasons for the Failure of Water Fall Model?

Ans. Reasons for the Failure of Water Fall Model are :

- Real project rarely follow sequential Flow. Iterations are made in indirect manner.
- Difficult for customer to state all requirements explicitly.
- Customer needs more patients as working products reach only at deployment phase.

Q.32 What are the Drawbacks of RAD Model ?

Ans. Drawbacks of RAD Model are :

- Require sufficient number of Human Resources to create enough number of teams.
- Developers and Customers are not committed, system result in failure.
- Not Properly Modularized building component may Problematic.
- Not applicable when there is more possibility for Technical Risk.

Q.33 Define the term Scripts.

Ans. Scripts Specific Process Activities and other detailed work functions that are part of team process.

Q.34. Write the disadvantages of classic life cycle model.

Ans. Disadvantages of classic life cycle model are :

I. Real projects rarely follow sequential flow. Iteration always occurs and creates problem.

II. Difficult for the customer to state all requirements.

III. Working version of the program is not available. So the customer must have patience.

Q.35. What do you mean by task set in spiral Model?

Ans. Each of the regions in the spiral model is populated by a set of work tasks called a task set that are adopted to the characteristics of the project to be undertaken.

Q.36 What is the main objective of Win-Win Spiral Model ?

Ans. The customer and the developer enter into the process of negotiation where the customer may be asked to balance functionality, performance and other product against cost and time to market.

Q.37. Which of the software engineering paradigms would be most effective? Why?

Ans. Incremental / Spiral model will be most effective.

Reasons :

I. It combines linear sequential model with iterative nature of prototyping.

II. Focuses on delivery of product at each increment.

III. Can be planned to manage technical risks.

Q.38. What are the merits of incremental model ?

Ans. The merits of incremental model are :

I. The incremental model can be adopted when there are less number of people involved in the project.

II. Technical risks can be managed with each increment.

III. For a very small time span, at least core product can be delivered to the customer.

Q.39. List the task regions in the Spiral model.

Ans. Task regions in the Spiral model are :

- **Customer Communication :** In this region it is suggested to establish customer communication.
- **Planning :** All planning activities are carried out in order to define resources timeline and other project related activities.
- **Risk Analysis :** The tasks required to calculate technical and management risks.

- **Engineering :** In this the task region, tasks required to build one or more representations of applications are carried out.

- **Construct and Release :** All the necessary tasks required to construct, test, install the applications are conducted.

- **Customer Evaluation :** Customer's feedback is obtained and based on the customer evaluation required tasks are performed and implemented at installation stage.

Q.40. What are the drawbacks of spiral model ?

Ans. The drawbacks of spiral model are :

1. It is based on customer communication. If the communication is not proper then the software product that gets developed will not be the up to the mark.
2. It demands considerable risk assessment. If the risk assessment is done properly then only the successful product can be obtained.

Q.41 Name the Evolutionary process Models.

Ans. Evolutionary powers models are :

I. Incremental model

II. Spiral model

III. WIN-WIN spiral model

IV. Concurrent Development

Q.42 Define Software Prototyping.

Ans. Software prototyping is defined as a rapid software development for validating the requirements.

Q.43 What are the benefits of prototyping ?

Ans. The benefits of prototyping are :

I. Prototype services as a basis for deriving system specification.

II. Design quality can be improved.

III. System can be maintained easily.

IV. Development efforts may get reduced.

V. System usability can be improved.

Q.44. What are the prototyping approaches in software process?

Ans. The prototyping approaches in software process are :

I. **Evolutionary prototyping :** In this approach of system development, the initial prototype is prepared and it is then refined through number of stages to final stage.

II. **Throw-away prototyping :** Using this approach a rough practical implementation of the system is produced. The requirement problems can be identified from this implementation. It is then discarded. System is then developed using some different engineering paradigm.

Q.45. What are the advantages of evolutionary prototyping ?

Ans. The advantages of evolutionary prototyping are :

I. Fast delivery of the working system.

II. User is involved while developing the system.

III. More useful system can be delivered.

IV. Specification, design and implementation work in co-ordinated manner.

Q.46. What are the various Rapid prototyping techniques ?

Ans. The various rapid prototyping techniques are :

I. Dynamic high level language development.

II. Database programming.

III. Component and application assembly.

Q.47. What is the use of User Interface prototyping ?

Ans. This prototyping is used to pre-specify the look and feel of user interface in an effective way.

Q.48. Give the phases of product development life cycle.

Ans. The phase of product development life cycle are :

- **Idea generation :** Ideas come from various sources like customers, suppliers, employees, market place demands.
- **Prototype development phase :** This entails buildings simplistic model of final product.
- **Beta phase :** This iron out the kinks in the product and add necessary supporting infrastructure to roll out the product.
- **Production phase :** In this phase product is ready for prime time.
- **Maintenance and obsolescence phase :** In this critical bugs are fixed after which the product goes into obsolescence.

Q.49. Explain water fall model in detail.

Ans. The project is divided into sequence of well defined phases. One phase is completed before next starts. There is a feedback loop between adjacent phases. What the actual phase are depends on the project.

Advantages :

- Simplicity
- Lining up resources with appropriate skills is easy

Disadvantages :

- Highly impractical for most projects
- Phases are tightly coupled.

Q.50 Explain RAD model in detail

Ans. The customer and developer agree on breaking the product into small units. Development is carried out using modeling tools and CASE tools. Customer is kept in touch so the changes are reflected time. Quality assurance is imposed.

Advantages :

- Responsiveness to change

- Ability to capture user requirements effectively.
- Application turn around time is shorter.

Disadvantages :

- Need for modeling tools which adds expense.
- Places restriction on type and structure.

Q.51. What is the principle of prototype model ?

Ans. A prototype is built to quickly demonstrate to the customer what the product would look like. Only minimal functionality of the actual product is provided during the prototyping phase.

Q.52. What is the advantage of Spiral model ?

Ans. The main advantages of spiral model is, it is realistic and typifies most software development products/projects. It combines the best features of most of the earlier models. It strikes a good balance mechanism for early problem identification and correction while not missing out proactive problem prevention.

Q.53 What is lifecycle model ?

Ans. Here different terms have specialization and responsibility in different life cycle phase.

Q.54. Why Formal Methods are not widely used?

Ans. Formal Methods are not widely used due to the following reasons :

- It is Quite Time Consuming and Expensive.
- Extensive expertise is needed for developers to apply formal methods.
- Difficult to use as their technically sophisticated maintenance may become risk.

Q.55. What are the Objectives of Requirement Analysis ?

Ans. Objectives of Requirement Analysis are :

- It describes what customer requires.
- It establishes a basis for creation of software design.

- It defines a set of requirements that can be validated once the software design is built.

Q.56. Define System Context Diagram (SCD)?

Ans. System Context Diagram (SCD):

- Establish information boundary between System being implemented and Environment in which system operates.
- Defines all external producers, external consumers and entities that communicate through user interface.

Q.57. Define System Flow Diagram (SFD)?

Ans. System Flow Diagram (SFD):

- Indicates Information flow across System Context Diagram region.
- Used to guide system engineer in developing system.

Q.58. What are the Requirements Engineering Process Functions?

Ans. Requirements Engineering Process Functions are:

- Inception
- Elicitation
- Elaboration
- Negotiation
- Specification
- Validation
- Management

Q.59. What are the Difficulties in Elicitations?

Ans. Difficulties in Elicitation are:

- Problem of Scope
- Problem of Understanding
- Problem of Volatility

Q.60. Define Quality Function Deployment (QFD)?

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

Q.61. Write a short note on structure charts.

Ans. These are used in architectural design to document hierarchical structure, parameters and interconnections in a system. No Decision box. The chart can be augmented with module by module specification of $\frac{I}{P}\, and\, \frac{o}{p}$ attributes.

Q.62. What are the contents of HIPO diagrams?

Ans. The contents of HIPO diagrams are:

Visual table of contents,

Set overview diagrams,

Set of details diagrams.

Q.63 Explain software Requirement Specification.

Ans. Software Requirement Specification includes;

- Information Description
- Behavioural Description
- Bibliography and Appendix
- Functional Description
- Validation criteria
- Preliminary user's Manual

Q.64 What is Requirement Engineering ?

Ans. Requirement engineering is the process of establishing to services that the customer required from the system and constraints under which it operates and is developed.

Q.65 What are the characteristics of SRS?

Ans. The characteristics of SRS are as follows:

i. **Correct:** The SRS should be made up the date when appropriate requirements are identified.

ii. **Unambiguous:** When the requirements are correctly understood then only it is possible to write unambiguous software.

iii. **Complete**: To make SRS complete, its hold be specified what a software designer wants to create software.

iv. **Consistent:** It should be consistent with reference to the functionalities identified.

v. **Specific:** The requirements should be mentioned specifically.

vi. **Traceable:** What is the need for mentioned requirement? This should be correctly identified.

Q.66 What are the objectives of Analysis modeling ?

Ans. The objectives of analysis modeling are:

i. To describe what the customer requires.

ii. To establish a basis for the creation of software design.

iii. To devise a set of valid requirements after which the software can be build.

Q.67 What is ERD?

Ans. Entity Relationship Diagram is the graphical representation of the object relationship pair. It is mainly used in database application.

Q.68 What is DFD?

Ans. Data Flow Diagram depicts the information flow and the transforms that are applied on the data as it moves from input to output.

Q.69 What does Level 0 DFD represent?

Ans. Level-0 DFD is called as fundamental system model or context model. In the context model the entire software system is represented by a single bubble with input and output indicated by incoming and outgoing arrows.

Q.70 What is a state transition diagram?

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

Q.71 Define Data Dictionary.

Ans. The data dictionary can be defined as an organized collection of all the data elements of the system with precise and rigorous definitions so that user and system analyst will have a common understanding of inputs, outputs, components of stores and intermediate calculations.

Q.72 What are the elements of analysis model?

Ans. The elements of analysis model are:

(i) Data Dictionary (ii) Entity Relationship Diagram

(iii) Data flow Diagram (iv) State Transition Diagram

(v) Control Specification (vi) Process Specification

Q.73 What are the elements of design model

Ans. The elements of design model are;

(i) Data Design (ii) Architectural design

(ii) Interface design (iv) Component-level design.

Q.74 What are the dimensions of requirements gathering?

Ans. The dimensions of requirements gathering are:

- Responsibilities: Commitments on either side Requirement form the basis for the success of further in a project.
- Current system requirements

  (i)Functionality requirements

  (ii) Performance requirements

  (iii) Availability needs

  (iv) Security

  (v) Environmental definitions
- Targets

- Acceptance criteria
- Ongoing needs: Documentation
  Training
  Ongoing support

Q.75 List the skill sets required during the requirements phase.

Ans. The skill sets required during the requirements phase are:

- Availability to look the requirements
- domain expertise
- Storing interpersonal skills
- Ability to tolerate ambiguity
- Technology awareness
- Strong negotiation skills
- Strong communication skills

Q.76 What is the primary objective of project closure ?

Ans. Evaluating effectiveness of the original project goals and providing to improve the system.

Q.77 What are the dimensions of requirements gathering?

Ans. The dimensions of requirements gathering are:

- Responsibilities
- Targets
- Current system needs
- Ongoing needs

Q.78 Give the classifications of system requirements.

Ans. The classification of system requirements are:

- Functionality Requirements
- Availability needs
- Environmental definition

Performance requirements

Security

Q.79 List some of the skills essential for requirements gathering phase.

Ans. The skills essential for requirements gathering phase are:

- Ability to see from customers point of view
- Technology awareness
- Domain expertise
- Strong interpersonal skills
- Strong communication skills

Q.80 What does P-CMM model stand for?

Ans. P-CMM stand for people CMM.

Q.81 What are the components of the Cost of Quality?

Ans. Components of the Cost of Quality are:

- Quality Costs.
- Prevention Costs.
- Appraisal Costs.

Q.82 What is Software Quality Control?

Ans. Software Quality Control involves series of inspections, reviews and tests which are used throughout software process to ensure each work product meets requirements placed upon it.

Q.83 What is Software Quality Assurance?

Ans. Software Quality Assurance is a set of auditing and reporting functions that assess effectiveness and completeness of quality control activities.

Q.84 What steps are required to perform Statistical SQA?

Ans. Following steps are required to perform Statistical SQA:

- Information about software defects is collected and categorized.
- Attempt is to trace each defect.
- Using Pareto principal, isolate 20%
- Once vital causes are identified, correct problems can be enforced to overcome.

Q.85 Define SQA Plan

Ans. SQA Plan provides road map for instituting SQA and it serves as template for SQA activities that are instituted for each software project.

Q.86 How do we define Software Quality?

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

Q.87 Define the terms:

(a) Quality of Design (b) Quality of Conformance

Ans. **a)** Quality of Design means characteristics, designer specify for an item.

b) Quality of Conformance means Degree to which design specifications are followed during manufacturing.

Q.88 How the Registration process of ISO 9000 certification is done?

Ans. Registration process of ISO 9000 certification has the following stages:

(i) Application. (ii) Pre-Assessment.

(iii) Documents Review and Adequacy of audit.

(iv) Compliance Audit (v) Registration

(vi) Continued Surveillance

Q.89 What are the factors of Software Quality?

Ans. Factors of Software Quality are:

- Portability
- Reusability
- Maintainability
- Usability
- Correctness

Q.90 List the process maturity levels in SEIs CMM.

Ans. The process maturity levels in SEIs CMM are:

Level 1: Initial: Few processes are defined and individual efforts are taken.

Level 2: Repeatable: To track cost schedule and functionality basis project management processes are established.

Level 3: **Defined**: The process is standardized, documented and followed.

Level 4: **Managed:** Both the software process and product are quantitatively understood and controlled using detailed measures.

Level 5: Optimizing: Establish mechanisms to plan and implement change.

Q.91 What does Verification represent?

Ans. Verification represents that set of activities that are carried out to confirm that the software correctly implements the specific functionality.

Q.92 What does Validation represent?

Ans. Validation represents the set of activities that ensure that the software that has been built is satisfying the customer's requirements.

Q.93 What is the use of CMM?

Ans. Software Quality means Conformance to explicitly stated functional and performance requirements, explicitly documented development standards, Implicit characteristics expected for professionally developed software.

Q.94 Distinguish between verification and validation.

- **Ans.** Verification refers to the set of activities that ensure that software correctly implements a specific function.
- Validation refers to a different set of activities that ensure that the software that has been built is traceable to the customer requirements. According to Boehm,
- Verification: Are we building the product right?
- Validation: Are we building the right product?

Q.95 What are the conditions that exists after performing validation testing?

Ans. After performing the validation testing their exists two conditions:

- The function or performance characteristics are according to the specifications and are accepted.

- The requirement specifications are derived and the deficiency list is created. The deficiencies then can be resolved by establishing the proper communication with the customer.

Q.96 Explain the Capability Maturity Model.

Ans. CMM model strives to achieve predictability and consistency as a precursor to continuous improvements by following a set of process in a well defined framework.

- Level 1 is Initial Level
- Level 2 is repeatable which helps in achieving repeatability of performance and quality should the organizations undertake a similar project again.
- Level 3 is defined level.
- Level four is measured level.
- Level 5 is optimistic level, here people always work towards a target.

Q.97 Name any two process models.

Ans. The ISO-9001 Model.

The capability maturity model.

Q.98 Define Quality.

Ans. Quality is about transforming as many of the implied requirements of the customer into stated requirements and meeting all the stated requirements.

Q.99 What is Quality control?

Ans. Quality control refers to testing a product after a given phase to find out if it has any defects.

Q.100 Define Quality Assurance.

Ans. Quality Assurance focuses on prevention of defects from the very start, and it is pro-active.

Q.101 What are the five major areas of SQA?

Ans. Requirements fidelity. Process compliance, change control minimizing the gap between defect injection and detection, and product quality.

Q.102 Name some tools of software quality assurance?

Ans. Inspection and review, Audits, Cross SQAs information sharing, Defect classification and Analysis tools.

Q.103.What is meant by Cardinality and Modality?

Ans. **Cardinality:** The number of occurrence of one object related to the number occurrence of another object is called Cardinality. Cardinality is of following types-

- One to One [1: 1]
- One to Many [1: N]
- Many to Many [M : N]

Modality: Means whether or not a particular Data Object must participate in the relationship.

Q.104. What is System Modeling?

Ans. System Modeling defines the important element in System Engineering Process.

- It defines Process in each view to be constructed
- It represents Behavior of the Process
- It explicitly defines exogenous and endogenous inputs.

Q.105. List out the Factors of Data Modeling?

Ans. Factors of Data Modeling are:

- Data Objects
- Data Attributes
- Relationship
- Cardinality and Modality.

Q.106. What are the Selection Characteristic for Classes?

Ans. Selection Characteristic for Classes are:

- Retained Information
- Needed Services
- Multiple Attribute
- Common Attribute
- Common Operations
- Essential Requirements.

Q.107. Define the terms in Software Designing:

(a) Abstraction (b) Modularity

Ans. (a) Abstraction is of two levels:

1. Highest Level: Solution is stated in broad term using language of problem environment.

2. Lower Level: More detailed description of solution is provided.

(b) Modularity:

- Software is divided into separately named and addressable components, called Modules that are integrated to satisfy problem requirements.

Q.108. How the Architecture Design can be represented?

Ans. Architectural Design can be represented by one or more different models. They are:

1. Structural Models 2. Framework Models

3. Dynamic Models 4. Process Models.

Q.109. What is the advantage of Information Hiding?

Ans. Advantage of Information Hiding is that during testing and maintenance phase if changes are required then they are done in particular module without affecting other module.

Q.110. What types of Classes does the designer create?

Ans. The designer create the following types of classes

• User interface Classes • Business Domain Classes

• Process Classes • Persistent Classes

• System Classes.

Q.111. What is Coupling?

Ans. Coupling is the quantitative measure of degree to which classes are connected to one another. Coupling should be kept as low as possible.

Q.112. What is Cohesion?

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

Q.113. Define Refactoring.

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

Q. 114. What are the five types of Design Classes?

Ans. Five Types of Design Classes are:

• User Interface Classes • Business Domain Classes

• Process Classes Persistent • Classes System Classes.

Q.115. What are the different types of Design Model?

Ans. Different types of Design Model are:

- Process Dimension: Indicate evolution of Design model as design tasks executed as part of software process.
- Abstraction Dimension: Represent level of detail as each element of analysis model is transformed into design equivalent.

Q. 116. List out the different elements of Design Model?

Ans. Different Elements of Design Model are:

• Data Design Elements • Architectural Design Elements

• Interface Design Elements

• Component Level Design Elements

• Deployment Level Design Elements.

Q.117. What types of Design Patterns are available for the software Engineer?

Ans. Types of Design Patterns are:

• Architectural patterns • Design Patterns

• Idioms.

Q.118. What is the Objective of Architectural Design?

Ans. Objective of Architectural Design is to model overall software structure by representing component interfaces, dependencies and relationships and interactions.

Q. 119. What are the different types of Cohesion?

Ans. Different Types of Cohesion are:

• Functional • Layer

• Communicational • Sequential

• Procedural • Temporal

• Utility.

Q. 120. What are the different types of Coupling?

Ans. Different Types of Coupling are:

• Content Coupling • Common Coupling

• Control Coupling • Stamp Coupling

• Data Coupling • Routine Call Coupling

• Type Use Coupling • Inclusion or Import Coupling

• External Coupling.

Q. 121. What is Program Design Language [PDL]?

Ans. Program Design Language [PDL] is also called Structured English or Pseudocode. PDL is a language that uses the vocabulary of one language and overall syntax another.

Q.122. Define Data Objects.

Ans. Data Objects are:

- External entity, occurrence or event, role, organizational unit, place or structure
- • Encapsulates Data only.

Q.123. What is the need for modularity?

Ans. Need for modularity is as it is easier to solve a complex problem. We can achieve reusability too. Best effort and complexity reduces.

Q.124. What are the five criteria that are used in modularity?

Ans. The five criteria that are used in modularity are:

- Modular Decomposability
- Modular Composability
- Modular Understandability
- Modular Continuity
- Modular Protection.

Q.125. What is Software Architecture?

Ans. Software Architecture means the overall structure of the software and the ways in which that software provides conceptual integrity for the system.

Q.126. What are the models used for Architectural design?

Ans. The models used for Architectural design are:

- Structural models
- Framework models
- Dynamic models
- Process models
- Functional models

Q.127. List the coupling factors.

Ans. Coupling factors are:

- Interface complexity between modules.
- Reference to the module the data pass across the interface.

Q.128. Define Stamp coupling.

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

Q.129. Define common coupling.

Ans. When a number of modules reference a global data area, then the coupling is called common coupling.

Q.130. Define temporal cohesion.

Ans. When a module contains tasks that are related by the fact that all must be execute within the same span of time, then it is termed as temporal cohesion.

Q.130. What do you mean by common coupling?

Ans. When a number of modules reference a global data area, then the coupling is called common coupling.

Q.131.Compare Data Flow Oriented Design with data structure oriented designs.

Ans. *Data flow oriented design*: Used to represent a system or software at any level of abstraction.

Data Structure oriented design: It is used for representing information hierarchy using the three constructs for sequence, selection and repetition.

Q.132. Define Architectural Design and Data Design.

Ans. Architectural Design means to develop a modular program structure and represent the relationships between modules.

Data Design means to select the logical representations of data objects, data storage and the concepts of information hiding and data abstraction.

Q.133. Explain the types of coupling and cohesion.

Ans. Coupling is the measure of the relative interdependence among modules.

Types : Data coupling, Stamp coupling, Control coupling, External coupling, Common coupling, Content coupling.

Cohesion is the measure of the relative function strength of a module.

Types : Coincidentally cohesive, Logically cohesive, Temporal cohesion, Procedural cohesion, Communicational cohesion, High cohesion, Sequential cohesion.

Q.134. What is data modeling?

Ans. Data modeling is the basic step in the analysis modeling. In data modeling the data objects are examined independently of processing. The data model represents how data are related with one another.

Q.135. What is a data object?

Ans. Data object is a collection of attributes that act as an aspect, characteristic, quality, or descriptor of the object.

Q.136. What are attributes?

Ans. Attributes are the one, which defines the properties of data object.

Q.137. What is cardinality in data modeling?

Ans. Cardinality in data modeling specifies how the number of occurrences of one object is related to the number of occurrences of another object.

Q.138. What does modality in data modeling indicates?

Ans. Modality indicates whether or not a particular data object must participate in the relationship.

Q.139. Define design process.

Ans. Design process is a sequence of steps carried through which the requirements are translated into a system or software model.

Q.140. List the principles of a software design.

Ans. The principle of a software design includes:

i.) The design process should not suffer from “tunnel vision”.

ii.) The design should be traceable to the analysis model.

iii.) The design should exhibit uniformity and integration.

iv.) Design is not coding.

v.) The design should not reinvent the wheel.

Q.141. What is the benefit of modular design?

Ans. Changes made during testing and maintenance becomes manageable and they do not affect other modules.

Q.142. What is a cohesive module?

Ans. A cohesive module performs only “one task” in software procedure with little interaction with other modules. In other words cohesive module performs only one thing.

Q.143. What are the different types of Cohesion?

Ans. Different types of cohesion are

(i) **Coincidentally Cohesive :** The modules in which the set of tasks are related with each other loosely then such modules are called coincidentally cohesive.

(ii) **Logically Cohesive :** A module that performs the tasks that are logically related with each other is called logically cohesive.

(iii) **Temporal Cohesion :** The module in which the tasks need to be executed in some specific time span is called temporal cohesive.

(iv) **Procedural Cohesion :** When processing elements of a module are related with one another and must be executed in some specific order then such module is called procedural cohesive.

(v) **Communicational cohesion** : When the processing elements of a module share the data then such module is called communicational cohesive.

Q.144. What is Coupling?

Ans. Coupling is the measure of interconnection among modules in a program structure. It depends on the interface complexity between modules.

Q.145. What are the various types of coupling?

Ans. Various types of coupling are:

(i) Data Coupling : The data coupling is possible by parameter passing or data interaction.

(ii) Control Coupling : The modules share related control data in control coupling.

(iii) Common Coupling : The common data or a global data is shared among modules.

(iv) Content Coupling: Content coupling occurs when one module makes use of data or control information maintained in another module.

Q.146. What are the common activities in design process?

Ans. The common activities in design process are:

(i) **System structuring:** The system is subdivided into principle sub-systems components and communications between these sub-systems are identified.

(ii) **Control modeling**: A model of control relationships between different parts of the system is established.

(iii) **Modular decomposition** : The identified subsystems are decomposed into modules.

Q.147. What are the various elements of data design?

Ans. The various elements of data design are:

(i) **Data object**: The data objects are identified and relationship among various data objects can be represented using ERD or data dictionaries.

(ii) Data warehouses: At the business level useful information is identified from various databases and data warehouse are created.

Q.148 List the guideline for data design.

Ans. The guidelines for data design are.

(i) Apply systematic analysis on data.

(ii) Identify data structures and related operation.

(iii) Establish data dictionary

(iv) Use information hiding in the design of data structure

(v) Apply a library of useful data structures and operations.

Q.149 What is the Objective of the project Planning Process?

Ans. Objective of the Project Planning Process is to provide framework that enable manager to make reasonable estimates of resources, cost and schedule.

Q.150 What are the Decomposition Techniques?

Ans. Decomposition Techniques are:

• Software Sizing • Problem — Based Estimation

• Process — Based Estimation • Estimation with Use Cases.

• Reconciling Estimates.

Q.151.How do we compute the "Expected Value" for Software Size?

- **Ans.** Expected value for estimation variable (size), S, can be compute as Weighted Average of Optimistic (opt) most likely (m), and Pessimistic (pess) estimates.

- S = (opt + 4 m + pess) / 6

Q.152. What is an Object Point?

Ans. Object Point means: Count is determined by multiplying original number of object instances by weighting factor and summing to obtain total object point count.

Q.153. List out the basic principles of software project scheduling?

Ans. Basic Principles of Software Project Scheduling are:

• Compartmentalization

• Interdependency

• Time Allocation

• Effort Validation

• Defined Responsibilities Defined Outcomes

• Defined Milestones.

Q.154. Write the objective of project planning?

Ans. It is to provide a framework that enables the manager to make reasonable estimates of resources, cost and schedule.

Q.155. What is FP? How it is used for project estimation?

Ans. Function Point: It is used as the estimation variable to size each element of the software. It requires considerably less details. It is estimated indirectly by estimating the number of inputs, outputs, data files, external interfaces.

Q.156. What is LOC? How it is used for project estimation?

Ans. LOC: Lines of Code. It is used as estimation variable to size each element of the software. It requires considerable level of detail.

Q.157. Write the formula to calculate the effort in persons-months used in Dynamic multi variable Model?

Ans. Software Equation: E = [LOC’ B°333/P]3 ‘(l/tj Where E is effort in person-months, t is project duration, B is special skills factor, P is productivity parameter.

Q.158. What is called as an object point?

Ans. It is an indirect software measure that is computed using counts of the number of screens, reports and components.

Q.159. Define measure.

Ans. Measure is defined as a quantitative indication of the extent, amount, dimension, or size of some attribute of a product or process.

Q.160. Define metrics.

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

Q. 161. What are the types of metrics?

Ans. The types of metrics are:

- **Direct metrics :** It refers to immediately measurable attributes. Example : Lines of code, execution speed.
- Indirect metrics: It refers to the aspects that are not immediately quantifiable or measurable.

  Example : Functionality of a program.

Q.162. What are the advantages and disadvantages of size measure?

Ans. The advantages and disadvantages of size measure are:

Advantages:

- Artifact of software development which is easily counted.
- Many existing methods use LOC as a key input.
- A large body of literature and data based on LOC already exists.

Disadvantages:

- This method is dependent upon the programming language.
- This method is well designed but shorter program may get suffered.
- It does not accommodate non procedural languages.
- In early stage of development it is difficult to estimate LOC.

Q.163. Write short notes on the various estimation techniques.

Ans. The various estimation techniques are:

- **Algorithmic cost modeling:** The cost estimation is based on the size of the software.
- **Expert Judgement :** The experts from software development and the application domain use their experience to predict software costs.
- **Estimation by analogy** : The cost of a project is computed by comparing the project to a similar project in the same application domain and then cost can be computed.
- **Parkinson's law** : The cost is determined by available resources rather than by objective assessment.
- **Pricing to win :** The project costs whatever the customer is ready to spend on it.

Q.164. What is COCOMO model?

Ans. Constructive Cost Model is a cost model, which gives the estimate of number of man-months it will take to develop the software product.

Q.165. Give the procedure of the Delphi method.

Ans. The procedure of the Delphi method consists of:

1. The co-ordinator presents a specification and estimation form to each expert.
2. Co-ordinator calls a group meeting in which the experts discuss estimation issues with the co-ordinator and each other.
3. Experts fill out forms anonymously.
4. Co-ordinator prepares and distributes a summary of the estimates.
5. The Co-ordinator then calls a group meeting. In this meeting the experts mainly discuss the points where their estimates vary widely.
6. The experts again fill out forms anonymously.
7. Again co-ordinator edits and summarizes the forms, repeats steps 5 and 6 until the co-ordinator is satisfied with the overall prediction synthesized from experts.

Q.166. What is the purpose of timeline chart?

Ans. The purpose of the timeline chart is to emphasize the scope of the individual task. Hence set of tasks are given as input to the timeline chart.

Q.167 What is EVA?

Ans. Earned Value Analysis is a technique of performing quantitative analysis of the software project. It provides a common value scale for every task of the software project. It acts as a measure for software project progress.

Q.168. What is Metrics?

Ans. Metrics in a project management context is about measurements. Measuring your progress in order to know where you are and what mid course corrections you need to take to achieve your goals.

Q.168. What are the formal models of size estimation?

Ans. The formal models of size estimation are:

- Lines of code
- Function points

Q.169. What is variance?

Ans. For any estimate the metric to determine its effectiveness is the variance.

Q.170. How to define variance?

Ans. Variance = (planned-Actual)/Planned * 100

Q.171. What is meant by activity schedule?

Ans. Activity schedule indicates the planned start and completion dates for each activity.

Q.172. What is mean by resource schedule?

Ans. Resource schedule shows the date on which each resource will be required and the level of that requirements are identified.

Q.173. What are the various types of charts used in visualizing progress?

Ans. The various types of charts used in visualizing progress are:

- Gannt chart,
- Bar chart
- Slip chart
- Time line chart.

Q.174. What is review point?

Ans. Project level progress reviews that generally take place at particular points during the life cycle of project is known as review point.

Q.175. What are the Basic Principles of Software Testing?

Ans. Basic Principles of Software Testing are:

- Traceable to Customer Requirements
- Planned long before Testing begins
- Pareto Principles applied to software testing
- Begin small and progress towards testing.
- Exhaustive testing is not possible.
- Conducted by independent third party.

Q.176. List out the Characteristics of Testability of Software?

Ans. Characteristics of Testability of Software?

- Operability
- Controllability
- Simplicity
- Understandability
- Observability
- Decomposability
- Stability

Q.176. List out Various Methods for Finding Cyclomatic Complexity?

Ans. Various Methods for finding Cyclomatic Complexity are:

- Number of Regions
- Cyclomatic Complexity V(G) , for flow Graph

$$V(G)=E-N+2$$

- Cyclomatic Complexity

$$V(G)=P+2$$

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

Q.11. Define Smoke Testing ?

Ans. Smoke Testing is Integration Testing and Commonly used when software products are being developed.

Q.11. What are the attributes of Good Test?

Ans. Attributes of Good Test are:

- High Probability of Finding errors
- “Best of Breed”
- Not Redundant
- Neither too Simple nor too complex

Q.11. Define White Box Testing?

Ans. White Box Testing is also called Glass Box Testing in this test case design uses control Structure of Procedural Design to derive test cases

Q.11. Define Basic path Testing

Ans. Basic Path Testing:

- White Box Testing
- Enable test case designer to derive a logical complexity measure of a procedural design.
- Use this measure as a Guide for defining a basis set of execution paths.

Q.11. Define the terms:

a) Graph Matrices.

b) Connection Matrices.

Ans. Graph Matrices:

- To develop software tool the data structure used is graph Matrix.
- Square Matrix
- Size equals number of nodes on the Flow graph

Connection Matrices:

- It Link Weight = 1= > Connection Exists
- It Link Weight=1=>Connection Does not Exists.

Q.11. What is Behavioural Testing?

Ans. Behavioral Testing :

- Also know as Black Box Testing.
- Focuses on Functional Requirement of software.
- Enables Software engineer to derive set of input condition that fully exercise all functional requirement of a software.

Q.11. What are the benefits of Smoke Testing?

Ans. 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.

Q.11. What errors are commonly found during Unit Testing?

Ans. Errors commonly found during Unit Testing are:

- Misunderstood or incorrect arithmetic precedence
- Mixed Mode Operations
- Incorrect Initializations
- Precision Accuracy
- Incorrect Symbolic representation of expression.

Q.177.What problems may be encountered when Top-Down Integration is chosen?

Ans. Following problems may be encountered when Top Down Integration is chosen:

- Develop stubs that perform limited functions that simulate the actual module.
- Integrate the software from the bottom of the hierarchy upward.

Q. 178. What are the Steps in Bottom-Up Integration?

Ans. Steps in Bottom-Up Integration are:

- Low level components are combined into clusters perform specific software sub function.
- Driver is written to coordinate test case input and output.
- Cluster is tested.
- Drivers are removed and clusters are combined moving inward in program structure.

Q. 179. What is Regression Testing?

Ans. Regression Testing means Re-execution of subset of tests that have already been conducted and to ensure changes have not propagated unintended side effects.

Q. 180. What are the Characteristics of "Critical Module"?

Ans. Characteristics of "Critical Module" are

- Addresses several software requirements
- Has High Level of Control
- Complex or error prone
- Has Definite Performance Requirements.

Q. 181. What are the Properties of Connection Matrices?

Ans. Properties of Connection Matrices are:

- Probability that link will execute
- Processing time expended during traversal of link
- Memory required during traversal of link
- Resource required during traversal of link.

Q. 182. What is Flow Graph Notation?

Ans. Flow Graph Notation means Simple notation for representing Control Flow. It is drawn only when Logical Structure of component is complex.

Q.183. Define Cyclomatic Complexity?

Ans. Cyclomatic Complexity is a Quantitative measure of Logical Complexity and it is the number of Independent Paths in the basis set program.

Q. 184. What is Equivalence Partition?

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

Q.185. List out the possible errors of Black Box Testing?

Ans. Errors of Black Box Testing are:

- Incorrect or Missing Functions
- Interface Errors
- Errors in Data Structures or external databases
- Behavioural or Performance errors
- Initialization or Termination errors.

Q.186. What are the steps followed in testing?

Ans. The steps followed in testing are:

i.) **Unit testing :** The individual components are tested in this type of testing.

ii.) **Module testing:** Related collection of independent components are tested.

iii.) **Sub-system testing** : This is a kind of integration testing. Various modules are integrated into a sub-system and the whole sub-system is tested.

iv.) **System testing**: The whole system is tested in this system.

v.) **Acceptance testing :** This type of testing involves testing of the system with customer data if the system behaves as per customer need then it is accepted.

Q.187. Define software testing? -

Ans. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and coding.

Q.188. What are the objectives of testing?

Ans. The objectives of testing are:

(i) Testing is a process of executing a program with the intend of finding an error.

(ii) A good test case is one that has high probability of finding an undiscovered error.

(iii) A successful test is one that uncovers as an-yet undiscovered error.

Q.189. What are the testing principles the software engineer must apply while performing the software testing?

Ans. The testing principles to be applied are:

(i) All tests should be traceable to customer requirements.

(ii) Tests should be planned long before testing begins.

(iii) The pareto principle can be applied to software testing-80% of all errors uncovered during testing will likely be traceable to 20% of all program modules.

(iv) Testing should begin “in the small” and progress toward testing “in the large”.

(v) Exhaustive testing is not possible.

(vi) To be most effective, an independent third party should conduct testing.

Q.190. What are the two levels of testing?

Ans. The two levels of testing are

(i) **Component testing:** Individual components are tested. Tests are derived from developer’ s experience.

(ii) **System Testing** The group of components are integrated to create a system or sub-system is done. These tests are based on the system specification.

Q.191. What are the various testing activities?

Ans. The various testing activities are:

(i) Test planning (ii) Test case design

(iii Test execution (iv) Data collection

(v) Effective evaluation

Q.192. Write short note on black box testing.

Ans. The black box testing is also called as behavioral testing. This method fully focuses on the functional requirements of the software. Tests are derived that fully exercise all functional requirements.

Q.193. What is equivalence partitioning?

Ans. Equivalence partitioning is a black box technique that divides the input domain into classes of data. From this data test cases can be derived. Equivalence class represents a set of valid or invalid states for input conditions.

Q.194. What is a boundary value analysis?

Ans. A boundary value analysis is a testing technique in which the elements at the edge of the domain are selected and tested. It is a test case design technique that complements equivalence partitioning technique. Here instead of focusing on input conditions only, the test cases are derived from the output domain.

Q.195. What are the reasons behind to perform white box testing?

Ans. There are three main reasons behind performing the white box testing.

1. Programmers may have some incorrect assumptions while designing or implementing some functions. Due to this there are chances of having logical errors in the program. To detect and correct such logical errors procedural details need to be examined.

2. Certain assumptions on flow of control and data may lead programmer to make design errors. To uncover the errors on logical path, white box testing is must.

3. There may be certain typographical errors that remain undetected even after syntax and type checking mechanisms. Such errors can be uncovered during white box testing.

Q.196. What is cyclomatic complexity?

Ans. Cyclomatic complexity is software metric that gives the quantitative measure of logical complexity of the program.

The Cyclomatic complexity defines the number of independent paths in the basis set of the program that provides the upper bound for the number of tests—that must be conducted to ensure that all the statements have been executed at least once.

Q.197. How to compute the cyclomatic complexity?

Ans. The cyclomatic complexity can be computed by any one of the following ways:-

1. The numbers of regions of the flow graph correspond to the cyclomatic complexity.

2. Cyclomatic complexity, V(G), for the flow graph, G, is defined as:

$$V(G) = E — N +2,$$

   E — number of flow graph edges,

   N — number of flow graph nodes

3. V(G) = P + 1

   Where P is the number of predicate nodes contained in the flow graph.

Q.198. What are the various testing strategies for conventional software?

Ans. The various testing strategies are:

(i) Unit testing (ii) Integration testing

(iii) Validation testing (iv) System testing.

Q.199. Write about drivers and stubs.

Ans. Drivers and stub software need to be developed to test incompatible software.

- The “**driver**” is a program that accepts the test data and prints the relevant results.
- The “**stub**” is a sub-program that uses the module interfaces and performs the minimal data manipulation if required.

Q.200. What are the approaches of integration testing?

Ans. The integration testing can be carried out using two approaches.

1. The non-incremental testing.

2. Incremental testing.

Q.201. What are the advantages and disadvantages of big-bang?

Ans. The advantages and disadvantages of big bang are:

Advantages:

- This approach is simple.

Disadvantages:

- It is hard to debug.
- It is not easy to isolate errors while testing.
- In this approach it is not easy to validate test results.
- After performing testing, it is impossible to form an integrated system.

Q.202. What are the benefits of smoke testing?

Ans. The benefits of smoke testing are:

- Integration risk is minimized.
- The quality of the end-product is improved.
- Error diagnosis and correction are simplified.
- Assessment of program is easy.

Q.203. Distinguish between Alpha and Beta testing.

Ans. Alpha and Beta testing are the types of acceptance testing.

- **Alpha test :** The alpha testing is attesting in which the version of complete software is tested by the customer under the supervision of developer. This testing is performed at developer's site.
- **Beta test :** The beta testing is a testing in which the version of the software is tested by the customer without the developer being present. This testing is performed at customer's site.

Q.203. What are the types of static testing tools?

Ans. There are 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 the as per user requirements.

Q.204. What are the activities that make up testing?

Ans. The activities that make up testing are:

• Test specification • Test design

• Test development • Test registration

• Test execution • Test maintenance.

Q.205. List the various types of testing done during product life cycle.

Ans. The various types of testing done during product life cycle are:

• White box testing • Black box testing

• Integration testing • System testing

• Installation testing • Regression testing

• Acceptance testing

Q.206. What is testing?

Ans. Testing refers to the activities that are carried out to ensure that the final software product meets the requirements that the product is intended to satis1r.

Q.207. What are the activities that make up testing?

Ans. The activities that make up testing are:

• Test specification • Test design

• Rest development • Test Registration

• Test execution • Test maintenance

Q.208. What is done in test design step?

Ans. The details of the layout, tooling and standards required for test development are designed in this stage.

Q.209. List some of the testing done during SDLC.

Ans. White box testing, black box testing, integration testing, system testing, installation testing. Regression testing, Acceptance testing.

Q.210. What is functionality testing?

Ans. It is a black box testing which exercises the basic functionality of the product from an external; perspective.

Q.211. What is interface testing?

Ans. Testing which tests the interfaces among modules.

Q.212. What do system testing do?

Ans. System testing focus on stressing the system under extreme conditions and ensuring that if there is any failure, it is well managed.

Q.213. What are the steps carried out in installation testing?

Ans. The steps carried out in installation testing are:

• Packaging • Documenting

• Installing • Verifying.

Q.214. Define regression testing.

Ans. Regression tests are defined as those tests that are run to verify that problems do not resurface or regress.

Q.215 Explain Integrated testing team model?

Ans. There in one project manage who manages both the development and the testing functions.

Q.216 What are the objective of Formal Technical Reviews.

Ans. The Objective of Formal Technical Reviews are:

- Uncover errors in function, logic and implementation for representation of software.
- Software represented according to predefined standard.

- Verify software under review meets requirements
- Achieve software developed in Uniform Manner.
- Make projects more manageable.

Q.217 What are the various types of system testing?

Ans. Various types of system testing are:

1. **Recovery testing**- is intended to check the system' a ability to recover from failures.
2. **Security testing**- verifies that system protection mechanism prevent improper penetration or data alteration.
3. **Stress testing**- Determines breakpoint of a system to establish maximum service level.
4. **Performance testing**- evaluates the run time performance of the software, especially real- time software.

Q.218 Define debugging.

Ans. Debugging is defined as the process of removal of defect. It occurs as a consequence of successful testing.

Q.219 What are the common approaches in debugging?

Ans. The common approaches tin debugging are:

- **Brute force method:** The memory dumps and run- time tracks are examined and program with write statements in loaded to obtain clues to error causes.
- **Back tracking method:** The source code is examined by looking backwards from symptom to potential causes or errors.
- **Causes eliminations method:** This method uses binary partitioning to reduce the number of location where errors can exists.

Q.220 Write about the types of project plan.

Ans. The types of project plan are.

- **Quality plan**: This plan describes the quality procedures and standards that will be used in a project.

- **Validation plant**: This plan describes the approach, resources and schedule required for system validation.

- **Configuration management plan**: This plan focuses on the configuration management procedures and structures to be used.

- **Maintenance plan:** The purpose of maintenance plan is to predict the maintenance requirements of the system, maintenance cost and effort required.

- **Staff development plan:** This plan describes how to develop this skills and experience of the project team members.

Q.221 What are the metrics computed during error tracking activity?

Ans. The metrices computed during error tracking activity are:

- Errors per requirement specification page.
- Errors per component- design level.
- Errors per component- code level.
- DRE- requirement analysis
- DRE- architectural analysis
- DRE- component level design
- DRE- coding .

Q.222 Why software change occurs?

Ans. Software change occurs because of the following reasons:

- New requirements emerge when the software is used.
- The business environment changes.
- Errors need to be repaired.
- New equipment must be accommodated.
- The performance or reliability may have to be improved.

Q.223 Write about software change strategies.

Ans. The software change strategies that could be applied separately or together are:

- **Software maintenance**: The changes are made in the software due to requirements.
- **Architectural transformation**: It is the process of changing one architecture into another form.
- **Software re-engineering**: New feature can be added to existing system and then the system is reconstructed for better use of it in future.

Q.224 Define maintenance.

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

Q.225 What are the types of software maintenance?

Ans. Types of software maintenance are :

- **Corrective Maintenance**: Means the maintenance for correcting the software faults.
- **Adaptive maintenance**; Means maintenance for adapting the change in environment.
- **Perfective maintenance**: Means modifying or enhancing the system to meet the new requirements.
- **Preventive maintenance**: Means changes made to improve future maintainability.

Q.226 Who reviews or approves the change request?

Ans. The configuration control board review and approves the change request.

Q.227 Give some design principles for maintainability.

Ans. Some design principles for maintainability are:

- Module level accountability
- Proper documentation

Q.228 What is done is maintenance phase?

Ans. The maintenance phase deals with the process of evaluating the customer's product change request, making necessary changes to the product.

Q.229 List the activities done in maintenance phase.

Ans. The activities done in maintenance phase are.

- Problem **reporting.**
- Problem resolution.
- Solution distribution.
- Proactive defect prevention.

Q.230 What is problem reporting?

Ans. Once the user finds the difference in using the product behavior and the behaviors that it has to be, the problem reporting takes place.

Q.231. What is the Use of Process Technology Tools?

Ans. Use of Process Technology Tools is to help Software Organizations to:

1. Analyze their current process
2. Organize work task
3. Control And Monitor Progress
4. Manage Technical Quality 5.

Q.232. List out the basic principles of software project scheduling?

Ans. Basic Principles of Software Project Scheduling are:

- Compartmentalization • Interdependency
- Tune Allocation • Effort Validation
- Defined Responsibilities • Defined Outcomes
- Defined Milestones.

Q.233. What Architectures are defined and developed as part of BPE?

Ans. Following Architectures are defined and developed as part of BPE:

- Data Architecture ‘Applications Architecture
- Technology Architecture.

Q.234. What is the Origin of changes that are requested for software?

Ans. Following are the Origin of Changes:

- New Business or Market Condition
- New Customer Needs
- Reorganization or business growth/downsizing
- Budgetary or Scheduling constraints.

Q.235. List out the Elements of SCM?

Ans. Elements of SCM arc:

• Component Elements • Process Elements

• Construction Elements • Human Elements.

Q.236. What are the Features supported by SCM?

Ans. Features supported by SCM are:

- Versioning
- Dependency tracking and change Management
- Requirements tracking
- Configuration Management
- Audit trails.

Q.237. What are the Objectives of SCM Process?

Ans. Objectives of SCM Process are:

- Identify all items, collectively define software configuration
- Manage changes to one or more these items
- Facilitate construction of different version of an application
- Ensure that the software quality is maintained.

Q.238. What are the Issues to be considered for developing tactics for Web App Configuration Management?

Ans. The issues to be considered for developing tactics for Web App Configuration Management are:

- Context
- People
- Stability.

Q.239. Define CASE Tools.

Ans. CASE Tools stands for Computer Aided Software Engineering. It is a System software that provide Automated support for software process activities. It Includes program used to support software process activities such as Requirement Analysis, System Modeling. Debugging and Testing.

Q.24O. What are the type of CASE Tools?

Ans. There are two types of CASE Tools:

- Upper CASE Tools
- Lower CASE Tools.

Q.241. Write a short note on 4GT.

Ans. Fourth Generation Techniques. 4GT encompasses a broad array, of software tools. Each tool enables the software developer to speci1' some characteristics of software at a higher level.

Q.242. List the tools or methods available for rapid prototypiug.

Ans. The tools or methods available for rapid prototyping are:

i. 4GT.

ii. Reusable software components.

iii. Specification and prototyping environment.

Q. 243. What are the aspects of software reuse?

Ans. The aspects of software reuse are:-

1) Software development with reuse

2) Software development for reuse

3) Generator based reuse

4) Application system reuse.

Q.244. Define SCM.

Ans. It. is an umbrella activity that is applied throughout software process. It has a set of tracking and control activities that begin when a software engineering project begins and terminates only when the software project is taken out of operation.

Q.245. List the SCM Activities.

Ans. The SCM Activities are:

i.) Identify a change

ii.) Control change

iii.) Ensure that change is being properly implemented

iv.) Report changes to others who may have an interest.

Q.246. What is meant by software reusability?

Ans. A software component should be designed and implemented so that it can be reused in many different programs.

Q.247. What is CASE?

Ans. CASE: Computer Aided Software Engineering CASE provides the engineer I the ability to automate manual activities and to improve engineering insight.

Q.248. Write the distinction between SCM and Software Support.

Ans. **SCM :** Lt has a set of tracking and control activities that begin when a Software engineering project begins and terminates only when the software project is taken out of operation.

Software support: It has a set of software engineering activities that occur a software has been delivered to the customer and put into operation.

Q.249. What is the difference between basic objects and aggregate objects used software configuration.

Ans. Basic Objects: It represents unit of text. e.g., Section of requirement specifications, Source listing for a component.

Aggregate Objects: Collection of basic: objects. And other aggregate objects e.g Full design specification is.

Q.250. What is software reuse? Explain the various aspects of software reuse.

Ans. Software reuse means a software component should be designed and implement so that it can be reused in many different programs.

The various aspects of software reuse:

- Software development with reuse
- Software development for reuse Generator based reuse
- Application system reuse

Q.251. Describe the various Software Configuration Management Tasks.

Ans. The Various Software Configuration Management tasks are:

- SCM Definition
- Process
- Software Configuration Items
- Version Control
- Configuration Audit
- Activities
- Baselines
- Identification of objects
- Change Control
- Status Reporting

Q.252. What are the different types of CASE tools available In Software Engineering?

Ans. The different types of CASE tools available in Software Engineering are:

- Business Process Engineering Tools
- Process Modeling and Management Tools
- Project Planning tools
- Risk Analysis Tools

- Project Management Tools
- Requirements Tracing Tools
- Documentation Tools
- System Software Tools
- Quality Assurance Tools
- Database Management Tools
- Software Configuration Management Tools
- Analysis and Design Tools PRO/SIM Tools
- Interface Design and Development Tools
- Prototyping Tools
- Programming Tools
- Web d:velopment Tools
- Integration arid Testing Tools
- Static Analysis Tools
- Dynamic Analysis Tools
- Test Management Tools
- Client/Server Testing Tools
- Re-Engineering Tools

Q.253 What is SCI?

Ans. Software Configuration Item is information that is carried as part of the software engineering process.

Q.254 What is architectural evolution?

Ans. Architectural evolution is the process of changing a system from a centralized architecture to a distributed architecture like client server.

Q.255How the CASE tools are classified? CASE tools can be classified by

a) By function or use

b) By user type (e.g., manager, tester). or

c) By stage in software engineering process (e.g., requirements, test).

Q.256 List the process activities of software configuration management.

Ans. The steps that constitute software configuration management are:

- Initial working
- Base lining

- Change management
- Management of work spaces
- Configuration status accounting
- Configuration audit.

A.257 What is configuration management?

Ans. Configuration management is the combination of software, services and process that enable each developer to recreate and use the exact set of files and environment for a specific software product, version and platform.

Q.258 What is configuration?

Ans. A configuration is a set of related items identifiable by id.

Q.259 What is a configuration item?

Ans. It is an elementary part of the configuration that must be identified or versioned tracked and controlled.

Q.260 What are the steps that constitutes configuration management?

Ans. Initial working, Base lining, Change management, management of workspaces, configuration status accounting, configuration audit.

Q.261. What is Cross Cutting Concerns?

Ans. Cross Cutting Concerns is when concerns cut across multiple functions, feature and information.

Q.262. What are the different Phases of Unified Process?

Ans. Different Phases of Unified Process are:

- Inception Phase
- Elaboration Phase
- Construct ion Phase
- Transition Phase
- Production Phase.

Q.263. What are the two additional feature of Hayley Pirbhai Model?

Ans. Two additional feature of Hayley Pirbhi Model are:

- User Interface Processing
- Maintenance and Self Test Processing

Q.264. List out the Types of Traceability Table?

Ans. Types of Traceability Table are:

- Features Traceability Table
- Source Traceability Table
- Dependency Traceability Table
- Subsystem Traceability Table
- Interface Traceability; Table.

Q.265. What are the Benefits of Analysis Pattern?

Ans. Benefits of Analysis Pattern are:-

- It speeds up development of Analysis model.
- Transformation of Analysis into Design model.

Q.266. Define CRC Modeling?

Ans. CRC Modeling: Class Responsibility Collaborator Modeling is a collection of Standard Index Card and divided into 3 sections

1. Name of class at Top.
2. List of class Responsibilities at Left.
3. Collaborators at Right.

- Classes that cover the Information to complete its responsibilities.

Q.267. Define Swim Lane Diagram?

Ans. Swim Lane Diagram is a variation of activity diagram. It allows us to represent flow of activities and actors responsible for activity.

Q.268. Define steps in Behavioral Modeling.

Ans. Steps in Behavioral Modeling are:

- Evaluate all Use Cases.

- Identify Events.

- Create Sequence for each use cases.

- Build a State Diagram.

- Review Model for Accuracy and Consistency.

Q.269. What are the types of Interface Design Elements?

Ans. Types of Interface Design Elements are

- User Interfaces

- External Interfaces

- Internal Interfaces.

Q.27O. What are the basic design principles of Class-Based Components?

Ans. Basic Design principles of Class-Based Components arc:

- Open-Closed Principle [OCP]

- Liskov Substitution Principle (LSP]

- Dependency Inversion Principle [DIP]

- Interface Segregation Principle [ISP]

- a Release Reuse Equivalency Principle [RREP]

- Common Closure Principle ICCPI

- Common Reuse Principle ICRPI.

Q.271 What should we consider when we name components?

Ans. We should consider the following when we name components:

- Components

- Interface

- Dependencies and Inheritance.

Q.272 What Is Base line criteria in SCM?

Ans. Base line criteria in SCM are:

- Help to control Change.
- Specification or product that has been formally.
- Reviewed and agreed upon serves as basis for future development.
- That can he change only through formal change control procedures.

Q.273 Define Status Reporting?

Ans. It is also called Configuration Status Reporting and it is a SCM task that answers the following:

(a) What Happened? (b) Who did it?

(c) When did it happen? (d) What else will be affected?

Q.274. Write about Democratic Teams in software development, (Egoless Team).

Ans. It is egoless team. All team members participate in all decisions. Group leadership rotates from member to member based on tasks to be performed.

Q. 275. What are the two project scheduling methods?

Ans. The two project scheduling methods are:

a) PERT -Program Evaluation and Review Techniques.

b) CPM-Critical Path Method.

Q.276. What is Aesthetics?

Ans. Aesthetics is a science of art and beauty. These are fundamental to software design whether in art or technology.

Q. 277. Define Configuration Status Reporting.

Ans. Configuration Status Reporting means

• What happened? • Who did it?

• When it happened? • What else will be affected?

Q.278. What is the need for baseline?

Ans. Need for Baseline:

i.) Basis for further development

ii.) Uses formal change control procedure for change

iii.) Helps to control change.

Q.279. What is an effectors process?

Ans. The effectors process is a process that verifies itself. The effector process exists certain criteria.

Q.280. Define the computer based system.

Ans. The computer based system can be defined as a set or an arrangement of element that are organized to accomplish some predefined goal by processing information.

Q.291. What are the various types of traceability in software engineering?

Ans. The various types of traceability in software engineering are:

i. **Source traceability:** These are basically the links from requirement to stake holders who propose these requirements.

ii. **Requirements traceability** : These are links between dependant requirement

iii. **Design traceability** These are links from requirements to design.

Q.292. What are the benefits of horizontal partitioning?

Ans. The benefits of horizontal partitioning are:

i.) Software that is easy to test.

ii.) Software that is easier to maintain.

iii.) Propagation of fewer side effects.

iv.) Software that is easier to extend.

Q.293. What is vertical partitioning?

Ans. Vertical partitioning often called factoring suggests that the control and work should be distributed top-down in program structure.

Q.294. What are the advantages of vertical partitioning?

Ans. The advantages of vertical partitioning are:

(i) These are easy to maintain changes.

(ii) They reduce the change impact and error propagation.

Q. 295. Name the commonly used architectural styles.

Ans. The commonly used architectural styles are

i.) Data centered architecture.

ii.) Data flow architecture.

iii.) Call and return architecture.

iv.) Object-oriented architecture.

v.) Layered architecture.

Q.296. What is Transform mapping?

Ans. The transform mapping is a set of design steps applied on the DFD in order to map the transformed flow characteristics into specific architectural style.

Q.297. What is workspace?

Ans. The workspace replicates the environment in which the developer can build the product under the same conditions in which the corresponding base line is built.

Q.298. Define configuration Audit.

Ans. A Configuration audit means ensuring that configuration management is being followed as per stated guidelines and that all the quality records are in place.

Q.299. What is Pareto Analysis?

Ans. The 80-20 rule is probably one of the most widely observed laws of nature. 80% of the productivity of an organization is contributed by 20% of its people.

Q.300. What is fish-bone diagram?

Ans. Fish-Bone diagram is another common tool that is used for getting to the root cause of the defects.

Q.301. What are the components of project plan?

Ans. The WHAT part. The WHAT COST part. The WHEN part. The How part, The WHOM part.

Q.302. What is Work Breakdown Structure?

Ans. Work breakdown structure is the decomposition of the project into smaller and more manageable parts.

Q.303. What are the activities that increase the effect of project tracking?

Ans. The activities that increase the effect of project tracking are

- Status reporting
- Communications
- SPMP updates.

Q.304. What are the methods of communication?

Ans. One to one, Group meetings, Conferencing among teams in various locations, non- work related outings.

Q.305. What does project closure refer to?

Ans. Closure refers to the conclusion of a project or some logical part of the project.

Q.306 What is the primary objective of project closure?

Ans. Evaluating effectiveness of the original project goals and providing to improve the system.

Q.307. Define estimation.

Ans. Estimation is a process of expectations setting which forms the basis of qua the resources required to accomplish certain goals, based on certain clearly assumptions.

Q.308.What are the three phase of estimation?

Ans. The three phases of estimation are:

- Size

• Effort

• Schedule estimate

Q.309. What is second system effect?

Ans. It is a behavior pattern. When one goes from first project to the next, there is a tendency to stereotype the second project to it make it look like the first one.

Q.310. What are the design techniques that one can follow to enhance diagnosability.

Ans. The design techniques that one can follow to enhance diagnosability are:

- Providing enough foot prints
- Making context self contained
- Having self identified data structures.

Q.311. What are the challenges In building global teams?

Ans. Two challenges is building global terms are:

Cultural differences, communication issues, time zone differences and hiring retention.

Q.312. Name some models of global projects.

Ans. Models of global projects are;

Resource model, life cycle model, integrated team model.

Q.313. Define Resource model

Ans. Here one of the team is signified as the primary team, and this directs and allocates work for other teams.

Q.314. Explain integrated team model.

Ans. This is the true globally distributed model. The teams in different locations work together throughout all life cycle phases as peers.

Q.315. What is the effect of internet on project management?

Ans. Reduction in training time and Reduction in software distribution costs.

Q.316. What are the challenges in building global teams?

Ans. Cultural differences, communication issues, time zone issues, hiring and retentions.

Q.317. What are the various categories of resources?

Ans. The various categories of resources are:

- Labour
- Equipment
- Materials
- Space

Q.318. What is cost schedule?

Ans. Cost schedule shows the planned cumulative expenditure cost by the use of resource overtime.

Q.319. What is an Object Point?

Ans. Object Point means: Count is determined by multiplying original number of object instances by weighting factor and summing to obtain total object point count.

Q.320. What is called support risk?

Ans. Support risk is the degree of uncertainty fiat the resultant software will be easy to correct, adapt and enhance.

Q.321. What is RMMM?

Ans. RMMM stands for Risk Mitigation, Monitoring and Management Plan. It is also called Risk Aversion.

Q.322. What are four impacts of the project risk?

Ans. Four impacts of the project risk are:

Catastrophic, Critical, Marginal, Negligible.

Q.321. Explain the common risk tools and techniques.

Ans. There are at least six different ways of identifying the potential risks. These are:

- Examining organizational history

- Preparing checklists
- Information buying
- Framework based risk categorization
- Simulation
- Decision trees.

Q.322. What Is Risk?

Ans. Risks are events that are usually beyond the planner's control.

Q.323. What is Risk management?

Ans. Risk management is the process of anticipating hurdles in carrying out the original plan and providing alternate plans so that the impact on the originally anticipated final outcome is minimal.

Q.324. Give the two Important characteristics of the risk management?

Ans. The two important characteristics of risk management are:

It is proactive. It strives to reduce the impact of uncertainty.

Q.325. What are the three phases of Risk management?

Ans. The three phases of risk management are:

Risk identification, Risk Quantification, and Risk mitigation.

Q.326. What are the ways of Identifying the potential risks?

Ans. The ways of identifying the potential risks are:

Examining organizational history, preparing checklists, information buying, framework based risk categorization, simulation, Decision trees.

Q.327. What are the Dimensions of Risk quantification?

Ans. Probability and the impact of Risk.

Q.328. What Is Risk mitigation?

Ans. Mitigation is a possible means if minimizing or even avoiding the Impact of risk.

Q.329. What are the factors that lead to Risk?

Ans. The factors that lead to Risk are:

- Estimation errors.
- Planning assumptions.
- Business risks.

Q.330. What is meant by hazard?

Ans. Hazard is an event that create a problem for successful completion of a project,

Q.331. What are the various steps under risk analysis?

Ans. The various steps under risk analysis are:

- Risk Estimation.
- Risk identification.
- Risk evaluation.

Q.332. What is risk likelihood?

Ann. The probability of a hazard occurring is known as the risk likelihood.

Q.333. What is risk impact?

Ans. The effect that the resulting problem will have direct impact of the project is impact.

Q.334. Define the terms:

(a) Agility **(b) Agile Team**

Ans. (a) Agility: Dynamic Content Specific, Aggressively Change Embracing and Growth Oriented.

Q.335. Define Software Reliability?

Ans. Software Reliability is the Probability of failure free operation of computer program in a specific environment for a specified time.

Q.336. Name some software testing tools.

Ans. LoadRunner, WinRunner, Silk Test.

Q.337. Name some CASE tools useful during maintenance.

Ans. RIGI, CM.

Q.338. What is the relation between error, fault and failure.

Ans. Error or mistake or bug leads to a fault or defect and a program having fault when executed results in a software failure**.**

Q.339. What are the test points?

Ans. Test points allow data to be inspected or modified at various points in the system.

Q.340. Can we do functional and non-functional testing of websites too?

Ans. Yes. We can do both black and white box testing of websites also.

Q.341. Use cases can do testing also. True/False.

Ans. True. One use case can directly create one test case.

Q.342. What is pair testing?

Ans. Two testers work together on one computer to find defects.

Q.343. What is refactoring?

Ans. A small change to a database schema which improves its design.

Q.345. For a project XYZ, defects reported by the customer are 5 and internal defects reported are 150. Find the defect leakage.

Ans. Defect leakage = 5/(150+5) *100% = 3.22%

Q.346. What is Parkinson's law?

Ans. The effort estimate is equal to the amount of available resources or budget.

Q.347. For a certain project ABC, total defects attributed to all phases is 55 and total size of the project is 180FP. Find the defect injection rate?

Ans. Defect Injection Rate(DIR) = 55/180 = 0.30 Defects/FP.

Q.348. For a project XXX, number of defects are 55 and size is 100FP. Find the defect density.

Ans. Defect Density = 55/100 *100 = 55 defects/100FP.

Q.349. A project PP has 100 nos. Regression test cases, 80 nos. test cases executed during regression testing. Find the percentage of test cases executed.

Ans. % Test cases executed = (80/100) * 100% = 80%

Q.350. For a project bbb, testing efforts= 150hrs. and project efforts = 500hrs. Find % testing efforts.

Ans. % Testing efforts = (150/500) * 100% = 30%