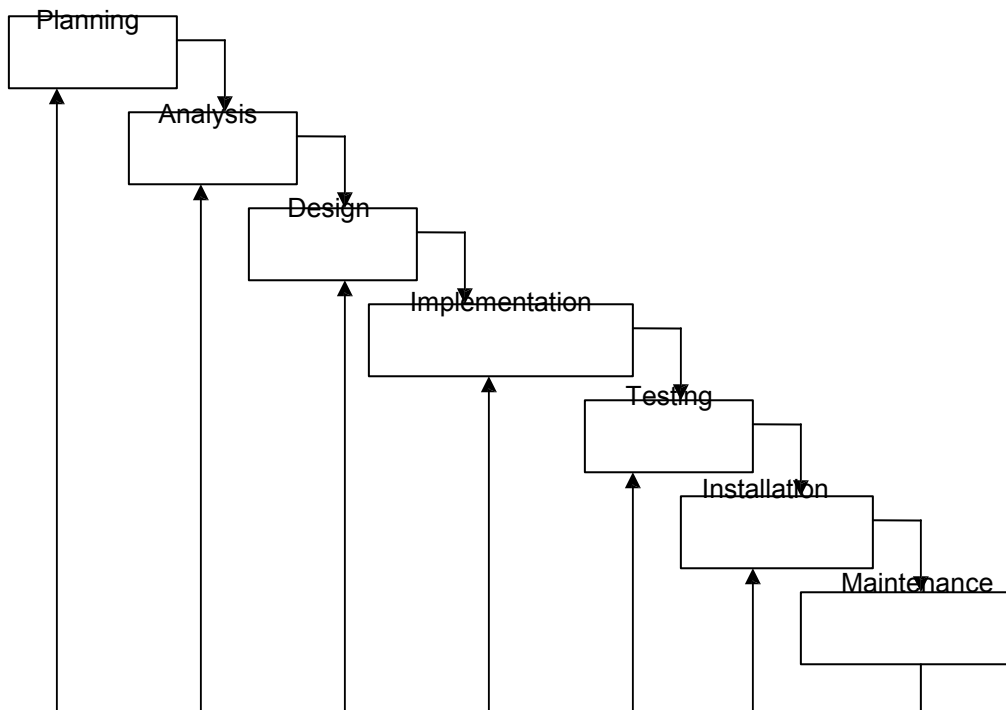# Software Engineering Question Bank

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

System Development Life Cycle (SDLC) is the overall process of developing information systems through a multi-step process from investigation of initial requirements through analysis, design, implementation and maintenance.

2) Draw a diagram for pure waterfall life cycle.

3) Explain the different phases involved in waterfall life cycle. (8 mks) Phase I –

Modeling Phase

In this phase we view the software product as part of a larger system or organization where the product is required. This is basically a system view where all the system elements are created.

Phase II – Software Requirements Analysis

Here we have a phase where the requirements are gathered. The information domain for the software is understood. The function, behaviour, performance and interfacing of the software are determined. The requirements of the software and the customer are decided upon.

Phase III – Design

This determines the data structures, the software architecture, the interface representations and the procedural (algorithmic) detail that goes into the software.

Phase IV – Code Generation

Here the actual programming is done to obtain the machine code; it is an implementation of the design.

Phase V – Testing

The testing is a process that goes hand in hand with the production of the machine code. There are a number of testing strategies. First unit testing is done and then integration testing. Alpha testing is to see if the software is as per the analysis model whereas beta testing is to see if the software is what the customer wanted.

Phase VI – Installation

The software is released to the customer.

Phase VII - Maintenance

This is the largest phase of the software life cycle. Maintenance can be of different types: to modify the software as the requirements of the customer evolve, to remove the residual bugs in the software etc.

4) **What is feasibility study? What are the contents we should contain in the feasibility report? (5 mks)**
A feasibility study is an initial look at an existing information processing system to decide how it might be computerized or improved.

The contents that a feasibility report are:
- A statement of purpose of the system.
- A definition of system scope.
- A list of deficiencies of the current system.
- A statement of user requirements.
- The cost and benefits of development.
- A conclusion and recommendations.

5) **What are the purposes of Data Flow diagrams, Entity-Relationship diagrams? Give an example diagram of each. (10 mks)**

Data Flow Diagrams
Data Flow diagrams are a means of representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data processes, and data sources/destinations.
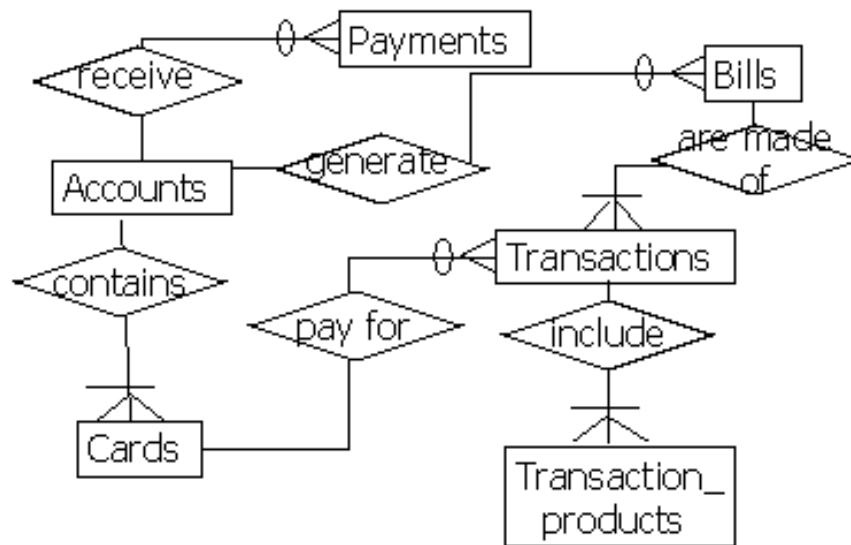
The purpose of data flow diagrams is to provide a semantic bridge between users and systems developers. The diagrams are:

- Graphical - Eliminating thousands of words;
- Logical representations - Modeling WHAT a system does, rather than physical models showing HOW it does it
- Hierarchical - showing systems at any level of detail; and
- Jargonless - allowing user understanding and reviewing.
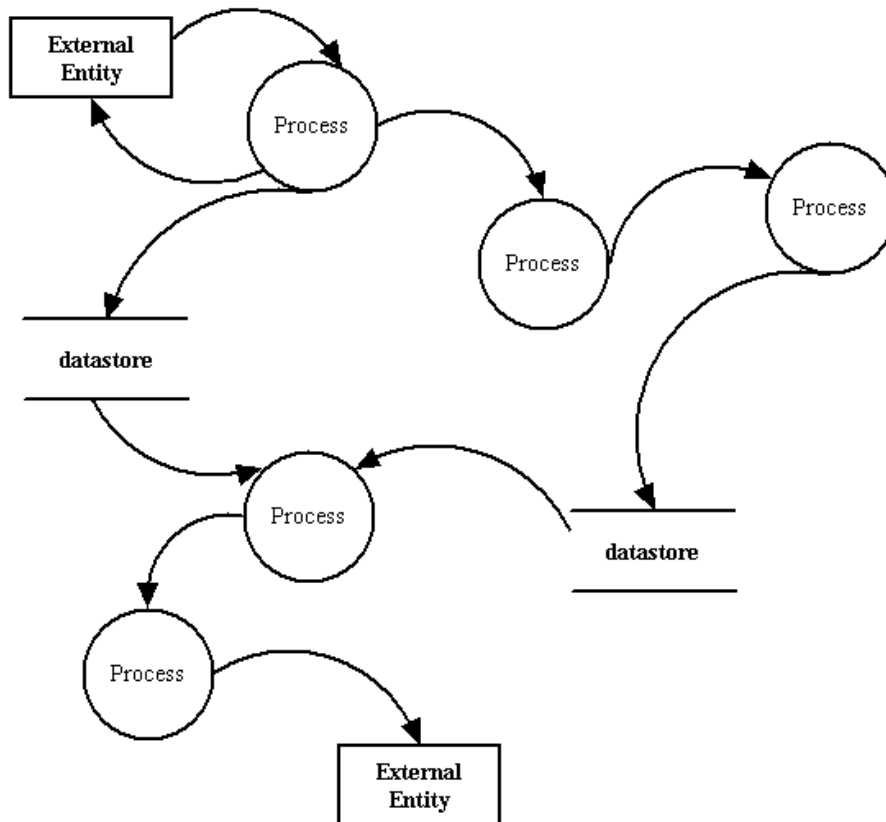
Entity Relationship Diagram
E-R Diagram is a graphical representation of the data layout of a system at a high level of abstraction.
It defines data elements and their inter-relationships in the system

Payments

Bills

receive

generate

are made of

Accounts

Transactions

contains

pay for

include

Cards

Transaction_products

6) **What is data modeling? Give 5 examples for data modeling. (5 mks)**
Data modeling is the act of exploring data-oriented structures. Like other modeling artifacts data models can be used for a variety of purposes, from high-level conceptual models to physical data models. From the point of view of an object-oriented developer, data modeling is conceptually similar to class modeling. With data modeling you identify entity types whereas with class modeling you identify classes. Data attributes are assigned to entity types just as you would assign
attributes and operations to classes.
Examples for data modeling include:
* Entity-Relationship diagrams
* Entity-Definition reports
* Entity and attributes report
* Table definition report
* Relationships, inheritance, composition and aggregation.

7) **What is the difference between SRS document and design document? What are the contents we should contain in the SRS document and design document.**

SRS Document
SRS document is a contract between the development team and the customer. Once the SRS document is approved by the customer, any subsequent controversies are settled by referring the SRS document. SRS document defines the customer's requirements in terms of Functions, performance, external interfaces and design constraints.
SRS Includes:
* Functional
* Non functional
* User
* Interface
* System

Design Document
The purpose of a design is to describe how the enhancements will be incorporated into the existing project. It should contain samples of the finished product. This could include navigational mechanism screenshots, example reports, and component diagrams.

Design Includes:
* E-R Diagrams
* Data flow diagrams
* Data Dictionary

8) **Explain all the phases involved in the implementation phase. (10 mks) Conduct system Test**
In this test software packages and in – house programs have been installed and tested, we need to conduct a final system test. All software packages, custom- built programs, and many existing programs that comprise the new system must be tested to ensure that they all work together
This task involves analysts, owners, users, and builders.

Prepare Conversion Plan
On successful completion of system test, we can begin preparations to place the new system into operation. Using the design specifications for the new system, the system analyst will develop a detailed conversion plan. This plan will identify Database to be installed, end – user training and documentation that needed to be developed, and a strategy for converting from the old system to the new system. The conversion plan may include one of the following commonly used installation strategies
1) Abrupt Cut-over
2) Parallel Conversion
3) Location Conversion
4) Staged Conversion

Install Databases

In the previous phase we built and tested the database. To place the system into operation we need fully loaded databases. The purpose of this task is to populate the new systems databases with existing database from the old system.
System builders play a primary role in this activity.

Train Users

Converting to a new system necessitates that system users be trained and provided with documentation that guides them through using the new system. Training can be performed one on one; however group training is preferred. This task will be completed by the system analysts and involves system owners and users.

Convert to New System

Conversion to the new system from old system is a significant milestone. After conversion, the ownership of the system officially transfers from the analysts and programmers to the end users.
The analyst completes this task by carrying out the conversion plan Recall that the conversion plan includes detailed installation strategies to follow for converting from the existing to the new production information system. This task involves the system owners, users, analysts, designers, and builders.

9) **List and explain different types of testing done during the testing phase. (10 mks) Unit**
Involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code
flow should be validated. Unit testing involves the use of debugging technology and testing
techniques at an application component level and is typically the responsibility of the developers, not the QA staff.

Integration

As the system is integrated, it is tested by the system developer for specification compliance.
•Concerned with testing the system as it is integrated from its components
•Integration testing is normally the most expensive activity in the systems integration process
•Should focus on:
•Interface testing where the interactions between sub-systems and components are tested
•Property testing where system properties such as reliability, performance and usability are tested

System

Testing the system as a whole to validate that it meets its specification and the objectives of its users. The testing of a complete system prior to delivery. The purpose of system testing is to identify defects that will only surface when a complete system is assembled. That is, defects that cannot be attributed to individual components or the interaction between two components. System testing includes testing of performance, security, configuration sensitivity, startup and recovery from failure modes. Involves test cases designed to validate that an application and its supporting hardware/software components are properly processing business data and transactions. System testing requires the use of regression testing techniques to validate that business functions are meeting defined requirements.

Black Box

This is testing without knowledge of the internal workings of the item being tested. For example, when black box testing is applied to software engineering, the tester would only know the "legal" inputs and what the expected outputs should be, but not how the program actually arrives at those outputs. It is because of this that black box testing can be considered testing with respect to the specifications, no other knowledge of the program is necessary. For this reason, the tester and the programmer can be independent of one another, avoiding programmer bias toward his own work.

White Box

Also known as *glass box*, *structural*, *clear box* and *open box testing*. White Box is a software testing technique whereby explicit knowledge of the internal workings of the item being tested are used to select the test data. Unlike Black Box testing, white box testing uses specific knowledge of programming code to examine outputs. The test is accurate only if the tester knows what the program is supposed to do. He or she can then see if the program diverges from its intended goal. White box testing does not account for errors caused by omission, and all visible code must also be readable.

10) **List and explain all the phases involved in the construction phase. (10 mks) Build and Test Networks**
   * In many cases new or enhanced applications are built around existing networks. If so there is no problem.
   * However if the new application calls for new or modified networks they must normally be implemented before building and testing databases and writing or installing computer programs that will use those networks.
   * This phase involves analysts, designers and builders
   * A network designer and network administrator assume the primary responsibility for completing this task.

Build and Test Databases
- This task must immediately precede other programming activities because databases are the resource shared by the computer programs to be written. If new or modified databases are required for the new system, we can now build and test those databases.
- This task involves system users, analysts, designers, and builders.
- The same system specialist that designed the database will assume the primary responsibility in completing this task

Install and Test New Software Packages
- Some systems solutions may have required the purchase or lease of software packages. If so, once networks and databases for the new system have been built, we can install and test the new software.
- This activity typically involves systems analysts, Designers, builders, vendors and consultants.

Write and Test New Programs
- In this phase we are ready to develop any programs for the new system. Prototype programs are frequently constructed in the design phase. However, these prototypes are rarely fully functional or incomplete.
- This task involves the system analysts, designers and builders.

11) **What is data conversion? Why is it necessary?**

Data Conversion is the changing of the data structure to accommodate new or different needs for the data. Different operating systems have different application software, and each application normally has its own internal way of saving data. There are some standards such as CSV files for databases and
RTF files for word processing text, however, these are few and far between and often only save the basic information rather than the full structure.

12) **What is change management?**

Computer based systems are dynamic. As the business Environment changes, there is a need of some changes to the information system. The changes occur not only during the study, design, and
development phases of the life cycle of the system. In this process there are two elements that are essential to the management of change.
- The performance review board, which can make management–level decisions about system modifications.
- Baseline documentation, which can be referred to, to determine the extent and impact of proposed modifications.

13) **What is user acceptance testing? Explain different testings in user acceptance testing. Why is it necessary?**

User Acceptance Testing is a phase of software development in which the software is tested in the "real world" by the intended audience.

Different testings are:
Alpha Testing
Alpha testing is the software prototype stage when the software is first able to run. It will not have all the intended functionality, but it will have core functions and will be able to accept inputs and generate outputs. An alpha test usually takes place in the developer's offices on a separate system.

Beta Testing
The beta phase of software design exposes a new product, which has just emerged from in-house (alpha) testing, to a large number of real people, real hardware, and real usage. Beta testing is not a method of getting free software long-term, because the software expires shortly after the testing period.

User acceptance testing is used to know if the system is working or not (both clients & in-house)

14) **What are functional and non-functional requirements? Functional**
- How the system should react to the particular inputs
- How the system should behave to the particular situations
- What the system should not do

Non functional
- Constraints on the services or functions
- Time constraints
- Constraints on the development process

15) **Explain the steps involved in the prototyping**
1. Define the goal and purpose of the prototyping.
2. Make plans for iterations (number, range) and evaluations (dates).
3. Transform the conceptual design to a first outline of the user interface and a first synopsis for the users' information.
4. Design the paper prototype.

5. Let domain experts review the paper prototype regarding completeness and correctness.
6. Test the prototype's usability.
7. Analyze the test results.
8. Decide on changes and additions regarding the conceptual model.
9. Design an executable prototype with users' information. Hold design meetings regularly, during which the following are documented:
    - implemented changes and additions.
    - unresolved requirements and problems.
10. Review and test the usability of the prototypes.
11. Analyze the test results.
12. Decide on changes and additions regarding the conceptual model
13. Design the next prototype, and so on.

16) **What is cardinality? Give examples (5 mks)**
   Cardinality is the number of occurrences that may exist between occurrences of two related entity types. The cardinalities between a related pair of entity types are: one to one, one to many, or many to many.

17) **What is a class and object? Give the diagrams and representation of class and object.** A class is a template for defining objects that specifies variables, and procedures that operate on those variables. A class is represented by a rectangle. The following diagram shows a typical class in a class diagram:
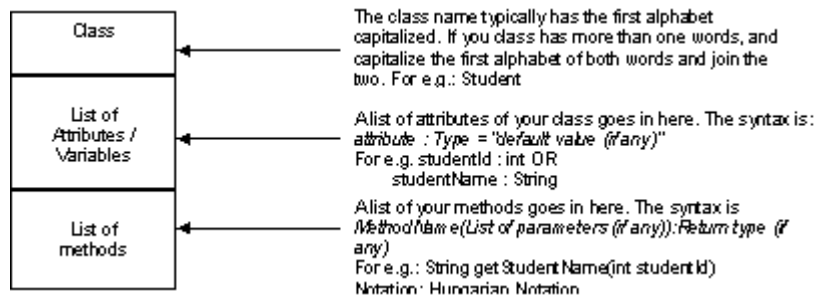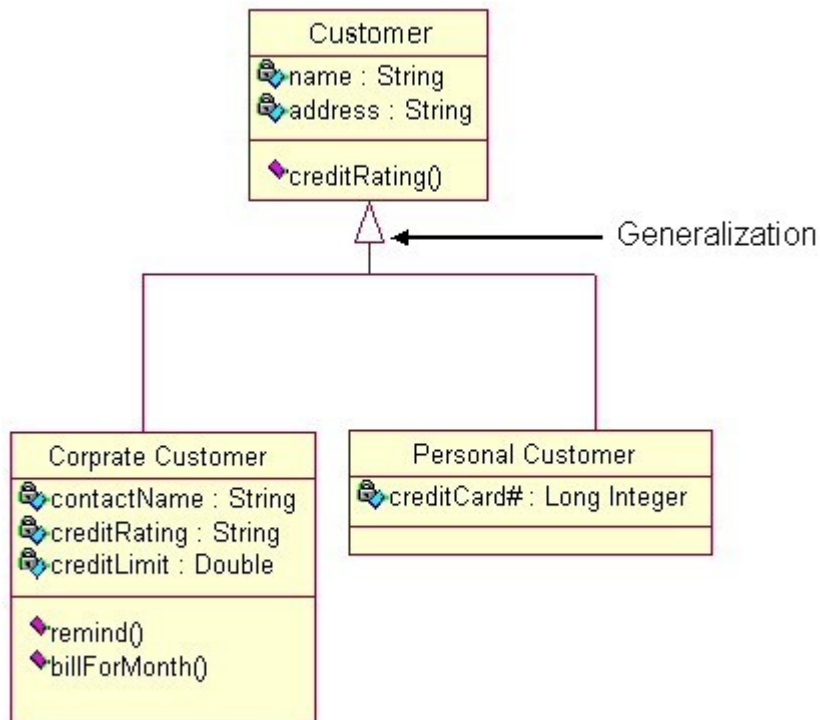

Objects are members, or instances, of a class formed by assigning specific values to the variables in the class.
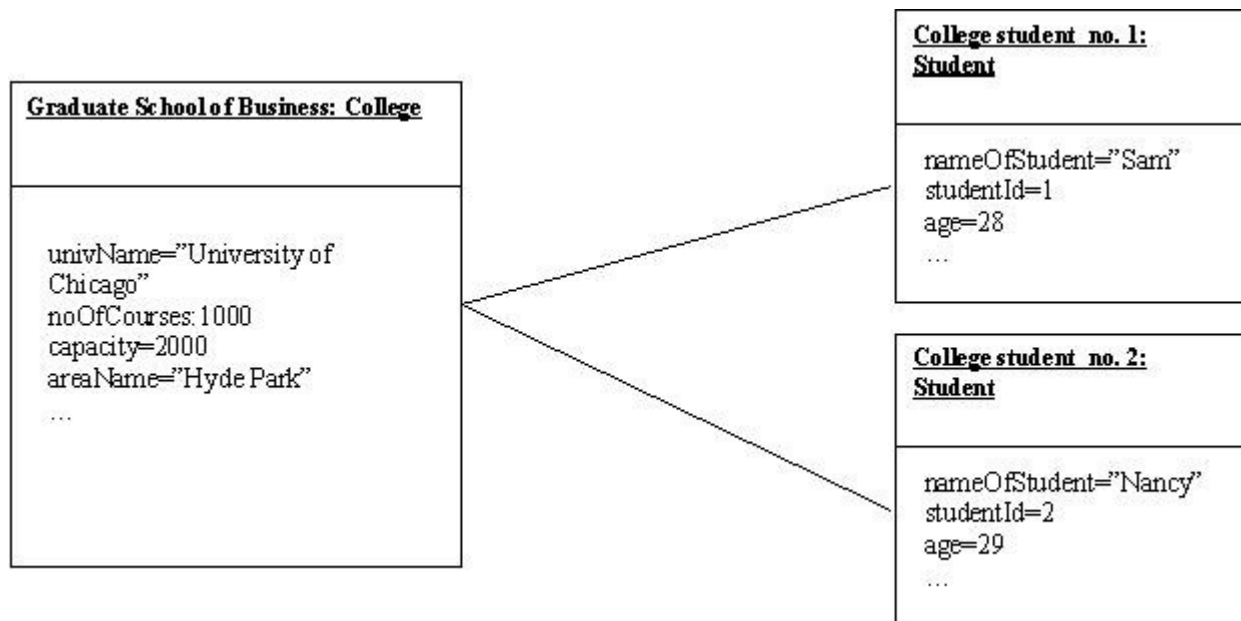
18) **What is generalization? Give an example of generalization**
   A generalization is an object class, which is a superset of another object class (or classes). Generalization models the "is a" relationship set since members of the specialization class (or classes) are always members of the generalization class. This means that members of the specialization class have all of the same properties of the generalization class including relationships with other objects as well as behaviour.

   A generalization is used when two classes are similar, but have some differences. Look at the generalization below:

Customer
name : String
address : String

creditRating()

Generalization

Corprate Customer
contactName : String
creditRating : String
creditLimit : Double

remind()
billForMonth()

Personal Customer
creditCard# : Long Integer

In this example the classes **Corporate Customer** and **Personal Customer** have some similarities such as name and address, but each class has some of its own attributes and operations. The class **Customer** is a general form of both the Corporate Customer and Personal Customer classes. This allows the designers to just use the Customer class for modules and do not require in-depth representation of each type of customer.

19) Define Software Engineering.
  Software Engineering :
• The Application of systematic, disciplined, quantifier
approach
• To the development, operations, and maintenance of software

20) What is a Process Framework?
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

21) What are the Generic Framework Activities?
Generic Framework Activities :
• Communication
• Planning
• Modeling
• Construction
• Deployment

22) Define Stakeholder.
Stakeholder :
• Anyone who has stake in successful outcome of
Project
• Business Managers, end-users, software engineer, support
people

23)How the Process Model differ from one another?
• Based on flow of activities
• Interdependencies between activities
• Manner of Quality Assurance
• Manner of Project Tracking
• Team Organization and Roles
• Work Products identify an requirement identifier

24) Write out the reasons for the Failure of Water Fall Model?
Reasons For The Failure Of Water Fall Model :

• 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 product reach only at Deployment phase

25) What are the Drawbacks of RAD Model?
Drawbacks of RAD Model :
• 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

26) Why Formal Methods are not widely used?
• Quite Time Consuming and Expensive
• Extensive expertise is needed for developers to apply formal methods
• Difficult to use as they are technically sophisticated maintenance may become risk

27)What is Cross Cutting Concerns?
Cross Cutting Concerns :
• When concerns cut across multiple functions, features and information

28) What are the different Phases of Unified Process?
Different Phases of Unified Process :
• Inception Phase
• Elaboration Phase
• Construction Phase
• Transition Phase
• Production Phase

29) Define the terms :
a) Agility
b) Agile Team
a) Agility :-
• Dynamic, Content Specific, Aggressively Change Embracing and Growth Oriented b)
Agile Team :-
• Fast Team
• Able to Respond to Changes

30) Define the terms:
a) Agile Methods b)
Agile Process
a)Agile Methods :-
• Methods to overcome perceive and actual weakness in conventional software engineering
• To accommodate changes in environment, requirements and use cases
b)Agile Process :-
• Focus on Team Structures, Team Communications, Rapid Delivery of software and it de-emphasis importance of intermediate product

31) What is the Use of Process Technology Tools? Use of Process Technology Tools :
• Help Software Organizations
1. Analyze their current process
2. Organize work task
3. Control And Monitor Progress
4. Manage Technical Quality

32) Define the term Scripts.
Scripts :
• Specific Process Activities and other detailed work
functions that are part of team process

33) What is the Objective of the Project Planning Process?
Objective of the Project Planning Process :
• To provide framework that enables manager to make
reasonable estimates of resources, cost and schedule

34) What are the Decomposition Techniques?
Decomposition Techniques :
• Software Sizing
• Problem – Based Estimation
• Process – Based Estimation
• Estimation With Use – Cases
• Reconciling Estimates

35) How do we compute the "Expected Value" for Software Size?
• Expected value for estimation variable(size), S, can be
compute as Weighted Average of Optimistic(Sopt),most
likely(Sm),and Pessimistic(Spess) estimates
• S = (Sopt+4Sm+Spess)/6

36) What is an Object Point?
Object Point :
• Count is determined by multiplying original number of object
instances by weighting factor and summing to obtain total object
point count

37) What is the difference between the "Known Risks" and Predictable Risks" ? Known
Risks :-
• That can be uncovered after careful evaluation of the
project plan, the business, and technical environment in
which the product is being developed
• Example : Unrealistic delivery rate
Predictable Risks :-
• Extrapolated from past project experience
• Example : Staff turnover

38) List out the basic principles of software project scheduling ? Basic
Principles Of Software Project Scheduling :-
• Compartmentalization
• Interdependency
• Time Allocation
• Effort Validation
• Defined Responsibilities
• Defined Outcomes
39)What is meant by Cardinality and Modality ?
Cardinality :-
• The number of occurrence of one object related to the
number of occurrence of another object
• One to One [1 :1]
• One to Many [1 : N]
• Many to Many [M : N]
Modality :-
• Whether or not a particular Data Object must participate in
the relationship

40) What are the Objectives of Requirement Analysis ?
Objectives of Requirement Analysis :
• Describe what customer requires
• Establish a basis for creation of software design
• Define a set of requirements that can be validated once the
software design is built

41) What are the two additional feature of Hayley Pirbhai Model?
• User Interface Processing
• Maintenance and Self test Processing

42) Define System Context Diagram[SCD]?
System Context Diagram[SCD] :
• Establish information boundary between System being
implemented and Environment which system operate
• Defines all external producers, external consumers and
entities that communicate through User Interface

43) Define System Flow Diagram[SFD]?
System Flow Diagram[SFD] :
• Indicates Information flow across SCD region
• Used to guide system engineer in developing system

44) What are the Requirements Engineering Process Functions?
• Inception
• Elicitation
• Elaboration
• Negotiation
• Specification
• Validation
• Management

45)Define SQA Plan.
SQA Plan :
• Provides roadmap for instituting SQA
• Plan serves as template for SQA activities that instituted for
each software project

46) What is Baseline criteria in SCM ?
• Help to control Change
• Specification or product that has been formally
• Reviewed and agreed upon serves as basis for future
development
• That can be change only through formal change control
procedures

47) Define Status Reporting ?
• Also called Configuration Status Reporting
• Is a SCM task that answers
1. What Happened ?
2. Who did it ?
3. When did it happen ?
4. What else will be affected ?

48) What is the Origin of changes that are requested for software?
Origin Of Change :-
• New Business or Market Condition
• New Customer Needs
• Reorganization or business growth/downsizing
• Budgetary or Scheduling constraints

49) List out the Elements of SCM?
Elements of SCM :-
• Component Elements
• Process Elements
• Construction Elements
• Human Elements

50) What are the Features supported by SCM?
Features supported by SCM :
• Versioning
• Dependency tracking and change Management

• Requirements tracking
• Configuration Management
• Audit trails

51) What are the Objectives of SCM Process?
Objectives of SCM Process :
• 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

52) What are the issues to be considered for developing tactics for WebApp Configuration
Management?
• Context
• People
• Scalability

53) Define CASE Tools.
CASE Tools :
• Computer Aided Software Engineering
• It is a System software
• Provide Automated support for software process activities
• Includes program used to support software process
activities
• Such as Requirement Analysis, System Modeling, Debugging and
Testing

54) How do we define Software Quality?
Software Quality :
• Conformance to explicitly stated functional and
performance requirements, explicitly documented
development standards
• Implicit characteristics, expected for professional
developed software

55) Define the terms :
a) Quality of Design
b) Quality of Conformance
Quality of Design :
• Characteristics, designer specify fro an item
Quality of Conformance :
• Degree to which design specifications are followed
during manufacturing

56) What are the Type of CASE Tools?
Types of CASE Tools :-
• Upper CASE Tools
• Lower CASE Tools

57) Define Software Reliability?
Software Reliability :
• Probability of failure-free operation of computer program in a
specified environment for a specified time

58) How the Registration process of ISO 9000 certification is done?
• Registration process of ISO 9000 certification has the
following stages
1. application
2. Pre-assessment
3. Document Review and Adequacy of audit
4. Compliance Audit
5. Registration

6. Continued Surveillance

59) What are the Factors of Software Quality?
Factors of Software Quality :
• Portability
• Usability
• Reusability
• Correctness
• Maintainability

QUESTION BANK (Software Engineering) QUESTIONS

1. What is software engineering?

2. What is the difference between program and software?

3. Write out the reasons for the Failure of Water Fall Model.
4. What are the characteristics of the software?

5. Define the terms :

i. Agility
ii. Agile Team

6. What are the various categories of software?

7. What are the challenges in software?
8. Define software process

9. What are the fundamental activities of a software process?

10.What are the umbrella activities of a software process?

11.What are the merits of incremental model?
12. List the task regions in the Spiral model.

13. What are the drawbacks of spiral model?

14. What is System Engineering?
15. List the process maturity levels in SEIs CMM.

16. What is an effectors process?

17. Define the computer based system.
18.What does Verification represent?

19. What does Validation represent?

20.What is the difference between the "Known Risks" and Predictable Risks"?

21. What are the steps followed in testing?

22. Explain about the incremental model. (16)
23. Explain in detail about the software process. (16)

24. Explain in detail about the life cycle process. (16)

25. Explain Spiral model and win-win spiral model in detail?

26. Name the Evolutionary process Models.
27. What are the Objectives of Requirement Analysis?

28. What is requirement engineering?

29. What are the various types of traceability in software engineering?
30. Define software prototyping.

31. What are the Requirements Engineering Process Functions?

32. What are the benefits of prototyping?
33. What are the prototyping approaches in software process?

34. What are the Difficulties in Elicitation?

35. What are the advantages of evolutionary prototyping?
36. What are the various Rapid prototyping techniques?

37. What is the use of User Interface prototyping?

38. What is System Modeling?

39. What are the characteristics of SRS?
40. What are the objectives of Analysis modeling?

41. What is data modeling?.What is a data object?

42. What is cardinality in data modeling?.What does modality in data modeling indicates?

43. What is ERD?What is DFD?

44. What does Level0 DFD represent?

45. What is a state transition diagram?
46. Explain in detail about Functional Modeling.

47. Explain in detail about Structural Modeling. (16)

48. Explain in detail about data modeling. (16)

49. Explain about rapid prototyping techniques. (16)
50. Explain the prototyping approaches in software process. (16)

51. What are the **elements** of Analysis model? What are the elements of design model?

52. How the Architecture Design can be represented?
53. Define design process. List the principles of a software design.

54, What is the benefit of modular design?

55. What is a cohesive module?
56. What are the different types of Cohesion?

57. What is coupling?What are the various types of coupling?

58. What are the common activities in design process?
59. What are the benefits of horizontal partitioning?.What is vertical partitioning?

60. What are the advantages of vertical partitioning?

61. What are the various elements of data design?
62. List the guidelines for data design..Name the commonly used architectural styles.

63. Explain in detail the design concepts. Explain the design principles.

64. Explain the design steps of the transform mapping.

65.What are the testing principles the software engineer must apply while performing the

software testing?

66. Define White Box Testing?

67. What are the two levels of testing?
68. What are the various testing activities?

69. Write short note on black box testing.

70. What is equivalence partitioning?
71. What is Regression Testing?

72. What is a boundary value analysis?

73. What are the reasons behind to perform white box testing?
74. What is cyclomatic complexity?

75. How to compute the cyclomatic complexity?

76. Distinguish between verification and validation.

77.What are the various testing strategies for conventional software?
78. Write about drivers and stubs.

79. What are the approaches of integration testing?

80. What are the advantages and disadvantages of big-bang?
81. What are the benefits of smoke testing?

82. What are the conditions exists after performing validation testing?

83. Distinguish between alpha and beta testing.
84. What are the various types of system testing? Explain the types of software testing.

85. Explain in detail about Black box testing.

86. Explain about the software testing strategies.
87. What are the advantages and disadvantages of size measure?

88. Write short note on the various estimation techniques.

89. What is the Objective of Formal Technical Reviews?
90. What is COCOMO model?.Give the procedure of the Delphi method.

91. What is the purpose of timeline chart?

92. What is EVA?

93. What are the metrics computed during error tracking activity?
94. Why software change occurs?

95. Write about software change strategies.

96. Define CASE Tools.

97. What is software maintenance?

98. Define maintenance.

99. What are the types of software maintenance? What is architectural evolution?

100. How the CASE tools are classified. Explain about software cost estimation.