CS1353-SOFTWARE ENGINEERING

TWO MARKS

1. **What is software engineering?**
Software engineering is a discipline in which theories, methods and tools
are applied to develop professional software.

2. What is Software ?
Software is nothing but a collection of computer programs that are related
documents that are indented to provide desired
features,functionalities and better
performance.

3. What are the characteristics of the software? ¾_Software is engineered,not manufactured. ¾_Software does not wear out. ¾_Most software is custom built rather than being assembled from components.

4. **What are the various categories of software?** ¾_System software ¾_Application software ¾_Engineering/Scientific software ¾_Embedded software ¾_Web Applications ¾_Artificial Intelligence software

5. **What are the challenges in software?** ¾_Copying with legacy systems. ¾_Heterogeneity challenge ¾_Delivery times challenge

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

7. What are the fundamental activities of a software process?
¾_Specification ¾_Design and implementation ¾_Validation ¾_Evolution

8. What are the umbrella activities of a software process?
¾_Software project tracking and control. ¾_Risk management. ¾_Software Quality Assurance. ¾_Formal Technical Reviews.

Notesengine.com

¾_Software Configuration Management. ¾_Work product preparation and production. ¾_Reusability management. ¾_Measurement.

9. What are the merits of incremental model?

3

i. The incremental model can be adopted when tere 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.

10. List the task regions in the Spiral model. ¾_**Customer communication –** In this region it is suggested to establish customer communication. ¾_**Planning –** All planning activities are carried out in order to define resources timeline and other project related activities. ¾_**Risk analysis –** The tasks required to calculate technical and management risks. ¾_**Engineering –** In this the task region,tasks required to build one or more representations of applications are carried out. ¾_**Construct and release –** All the necessary tasks required to construct,test,install the applications are conducted. ¾_**Customer evaluation –** Customer's feedback is obtained and based on the customer evaluation required tasks are performed and implemented at installation stage.

11. What are the drawbacks of spiral model?

i. 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.
ii. It demands considerable risk assessment.If the risk assessment is done
properly then only the successful product can be obtained.

12. What is System Engineering?

System Engineering means designing,implementing,deploying and operating systems which include hardware,software and people.

13**. List the process maturity levels in SEIs CMM.**

Level 1:Initial – Few processes are defined and individual efforts are taken.
Level 2:Repeatable – To track cost schedule and functionality basic 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.

14. **What is an effector process?**

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

15. Define the computer based system.

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

16. What does Verification represent?

4
Verification represents the set of activities that are carried out to confirm
that the software correctly implements the specific functionality.

17. What does Validation represent?

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

18. What are the steps followed in testing?

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 subsystem and the whole subsystem is tested.
iv. System testing – The whole system is tested in this system.

Notesengine.com

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.

19. What is the use of CMM?

Capability Maturity Model is used in assessing how well an organisation's
processes allow to complete and manage new software projects.

20. Name the Evolutionary process Models.

i. Incremental model
ii. Spiral model
iii. WIN-WIN spiral model
iv. Concurrent Development

21. What is requirement engineering?

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

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

i. Source traceability – These are basically the links from requirement to
stakeholders who propose these requirements.

ii. Requirements traceability – These are links between dependant requirements.

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

23. Define software prototyping.

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

24. What are the benefits of prototyping?

i. Prototype serves 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.

25. What are the prototyping approaches in software process?

5

Notesengine.com

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.

26. What are the advantages of evolutionary prototyping?

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.

27. What are the various Rapid prototyping techniques?

i. Dynamic high level language development.
ii. Database programming.
iii. Component and application assembly.

28. What is the use of User Interface prototyping?

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

Notesengine.com

29. What are the characteristics of SRS?

i. Correct – The SRS should be made up to date when appropriate requirements are identified.
ii. Unambiguous – When the requirements are correctly understood then
only it is possible to write an unambiguous software.
iii. Complete – To make SRS complete,it shold 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.

30. What are the objectives of Analysis modeling?
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
built.
31. What is data modeling?
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.
32. What is a data object?
Data object is a collection of attributes that act as an aspect, characteristic,
quality, or descriptor of the object.
33. What are attributes?
Attributes are the one, which defines the properties of data object.
34. What is cardinality in data modeling?
6
Cardinality in data modeling, cardinality specifies how the number of occurrences of one object is related to the number of occurrences of another
object.
Notesengine.com
35. What does modality in data modeling indicates?
Modality indicates whether or not a particular data object must participate
in the relationship.
36. What is ERD?
Entity Relationship Diagram is the graphical representation of the object
relationship pair.It is mainly used in database applications.
37. What is DFD?
Data Flow Diagram depicts the information flow and the transforms that
are applied on the data as it moves from input to output.
38. What does Level0 DFD represent?
Level0 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.

39. What is a state transition diagram?

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 event.

40. **Define Data Dictionary**.

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.

41. What are the elements of Analysis model?

i. Data Dictionary
ii. Entity Relationship Diagram
iii. Data Flow Diagram
iv. State Transition Diagram
v. Control Specification
vi. Process specification

Notesengine.com

42. What are the elements of design model?

i. Data design
ii. Architectural design
iii. Interface design
iv. Component-level design

43. Define design process.

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

44. List the principles of a software design.

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.

7

45. What is the benefit of modular design?

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

46. What is a cohesive module?

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.

47. What are the different types of Cohesion?

i. Coincidentally cohesive –The modules in which the set I\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.

48. What is Coupling?

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

49. What are the various types of coupling?

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

Notesengine.com

modules.
iv. **Content coupling** – Content coupling occurs when one module makes
use of data or control information maintained in another module.

50. What are the common activities in design process?

i. **System structuring** – The system is subdivided into principle subsystems components and communications between these subsystems
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.

51. What are the benefits of horizontal partitioning?

i. Software that is easy to test.
ii. Software that is easier to maintain.
iii. Propagation of fewer sideeffects.
8
iv. Software that is easier to extend.

52. What is vertical partitioning?

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

53. What are the advantages of vertical partitioning?

i. These are easy to maintain changes.
ii. They reduce the change impact and error propagation.

54. What are the various elements of data design?

i. Data object – The data objects are identified and relationship among
various data objects can be represented using ERD or data dictionaries.
ii. **Databases –** Using software design model, the data models are translated
into data structures and data bases at the application level.
iii. **Data warehouses –** At the business level useful information is identified
from various databases and the data warehouses are created.

55. List the guidelines for data design.

Notesengine.com

i. Apply systematic analysis on data.
ii. Identify data structures and related operations.
iii. Establish data dictionary.
iv. Use information hiding in the design of data structure.
v. Apply a library of useful data structures and operations.

56. Name the commonly used architectural styles.

i. Data centered architecture.
ii. Data flow architecture.
iii. Call and return architecture.
iv. Object-oriented architecture.
v. Layered architecture.

57. What is Transform mapping?

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.

58. What is a Real time system?

Real time system is a software system in which the correct functionalities
of the system are dependent upon results produced by the system and the time at
which these results are produced.

Notesengine.com

59. What is SCM?

Software Configuration Management is a set of activities carried out for
identifying, organizing and controlling changes throughout the lifecycle of
computer software.

60. What is SCI?

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

61. Define software testing?

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

62. What are the objectives of testing?

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

9

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.

63. **What are the testing principles the software engineer must apply while performing the software testing?**

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.

63. What are the two levels of testing?

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.

Notesengine.com

64. What are the various testing activities?

i. Test planning
ii. Test case design
iii. Test execution
iv. Data collection
v. Effective evaluation

65. Write short note on black box testing.

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

66. What is equivalence partitioning?

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.

67. What is a boundary value analysis?

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.

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

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 10
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.

69. What is cyclomatic complexity?

Cyclomatic complexity is a 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.

70. How to compute the cyclomatic complexity?

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.

71. Distinguish between verification and validation.

¾_Verification refers to the set of activities that ensure that software correctly implements a specific function. ¾_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, ¾_Verification:" Are we building the product right?" ¾_Validation:" Are we building the right product?"

72. What are the various testing strategies for conventional software?

i. Unit testing
ii. Integration testing.
iii. Validation testing.
iv. System testing.

Notesengine.com

73. Write about drivers and stubs.

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

11

¾_The " **driver**" is a program that accepts the test data and prints the relevant results. ¾_The " **stub**" is a subprogram that uses the module interfaces and performs the minimal data manipulation if required.

74. What are the approaches of integration testing?

The integration testing can be carried out using two approaches.
1. The non-incremental testing.
2. Incremental testing.

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

Advantages: ¾_This approach is simple.

Disadvantages: ¾_It is hard to debug. ¾_It is not easy to isolate errors while testing. ¾_In this approach it is not easy to validate test

results. ¾_After performing testing, it is impossible to form an integrated system.

76. What are the benefits of smoke testing? ¾_Integration risk is minimized. ¾_The quality of the end-product is improved. ¾_Error diagnosis and correction are simplified. ¾_Assessment of program is easy.

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

After performing the validation testing there exists two conditions. ¾_The function or performance characteristics are according to the specifications and are accepted. ¾_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.

78. **Distinguish between alpha and beta testing**. ¾_Alpha and beta testing are the types of acceptance testing. ¾_**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. ¾_**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.

79. What are the various types of system testing?

1. **Recovery testing** – is intended to check the system' s 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.

80. Define debugging.

12

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

81. What are the common approaches in debugging? ¾_**Brute force method:** The memory dumps and run-time tracks

Notesengine.com

are examined and program with write statements is loaded to obtain clues to error causes. ¾_**Back tracking method:** The source code is examined by looking backwards from symptom to potential causes of errors. ¾_**Cause elimination method:** This method uses binary partitioning to reduce the number of locations where errors can exists.

82. Write about the types of project plan. ¾_**Quality plan –** This plan describes the quality procedures and standards that will be used in a project. ¾_**Validation plan –** This plan describes the approach, resources and schedule required for system validation. ¾_**Configuration management plan –** This plan focuses on the configuration management procedures and structures to be used. ¾_**Maintenance plan –** The purpose of maintenance plan is to predict the maintenance requirements of the system, maintenance cost and efforts required. ¾_**Staff development plan –** This plan describes how to develop the skills and experience of the project team members.

83. Define measure.

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

84. Define metrics.

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

85. What are the types of metrics? ¾_**Direct metrics –** It refers to immediately measurable attributes. Example – Lines of code, execution speed. ¾_**Indirect metrics –** It refers to the aspects that are not immediately quantifiable or measurable. Example – functionality of a program.

86. What are the advantages and disadvantages of size measure?

Advantages: ¾_Artifact of software development which is easily counted. ¾_Many existing methods use LOC as a key input. ¾_A large body of literature and data based on LOC already exists.

Disadvantages: ¾_This method is dependent upon the programming language. ¾_This method is well designed but shorter program may get

Notesengine.com

suffered. ¾_It does not accommodate non procedural languages. ¾_In early stage of development it is difficult to estimate LOC.
13

87. Write short note on the various estimation techniques.

¾_**Algorithmic cost modeling –** the cost estimation is based on the size of the software. ¾_**Expert judgement –** The experts from software development and
the application domain use their exoerience to predict software costs. ¾_**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. ¾_**Parkinson's law –** The cost is determined by available resources
rather than by objective assessment. ¾_**Pricing to win –** The project costs whatever the customer ready to
spend it.

88. What is COCOMO model?

COnstructive COst MOdel is a cost model, which gives the estimate of
number of man-months it will take to develop the software product.

89. Give the procedure of the Delphi method.

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 coordinator 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,repeating steps 5 and 6 until the co-ordinator is satisfied with the overallprediction synthesized from experts.

90. What is the purpose of timeline chart?

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.

91. What is EVA?

Notesengine.com

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

92. What are the metrics computed during error tracking activity? ¾_Errors per requirement specification page. ¾_Errors per component-design level ¾_Errors per component-code level ¾_DRE-requirement analysis ¾_DRE-architectural analysis ¾_DRE-component level design ¾_DRE-coding.

93. Why software change occurs?

14

Software change occurs because of the following reasons. ¾_New requirements emerge when the software is used. ¾_The business environment changes. ¾_Errors need to be repaired. ¾_New equipment must be accommodated. ¾_The performance or reliability may have to be improved.

94. Write about software change strategies.

The software change strategies that could be applied separately or together
are: ¾_**Software maintenance –** The changes are made in the software
due to requirements. ¾_**Architectural transformation –** It is the process of changing one
architecture into another form. ¾_**Software re-engineering –** New features can be added to existing
system and then the system is reconstructed for better use of it in future.

95. What is software maintenance?

Software maintenance is an activity in which program is modified after it
has been put into use.

96. Define maintenance.

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.

97. What are the types of software maintenance? ¾_Corrective maintenance – Means the maintenance for correcting

the software faults. ¾_**Adaptive maintenance –** Means maintenance for adapting the
change in environment. ¾_**Perfective maintenance –** Means modifying or enhancing the
system to meet the new requirements. ¾_**Preventive maintenance –** Means changes made to improve
future maintainability.

98. What is architectural evolution?

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

99. How 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).

100. What are the types of static testing tools?

There are three types of static testing tools. ¾_**Code based testing tools –** These tools take source code as input
and generate test cases. ¾_**Specialized testing tools –** Using this language the detailed test
specification can be written for each test case.

¾_**Requirement-based testing tools –** These tools help in designing
the test cases as per user requirements.

Notesengine.com

SIXTEEN MARKS

1. **Explain iterative waterfall and spiral model for software life cycle and**
various activities in each phase.

Answer: Iterative waterfall model

The iterative waterfall model is as shown in the following figure.

· Requirement gathering phase in which all requirements are identified.
· The deign phase is responsible for creating architectural view of the software.
· The implementation phase in which the software design is transformed
into coding.
· Testing is a kind of phase in which the developed software component is

fully tested.
· Maintenance is an activity by which the software product can be maintained.
Requirements
Design
Implementation
Testing
Maintenance
16
SPIRAL MODEL
· The spiral model is divided into number of frame works. These frameworks are
denoted by task regions.
· Usually there are six task regions. In spiral model project entry point axis is
defined.
· The task regions are:
Customer communication
Planning
Risk analysis.
Engineering.
Construct and release.
Customer evaluation.
Drawbacks
· It is based on customer communication.
· It demands considerable risk assessment**.**
2. Explain about the incremental model.
· Have same phases as the waterfall model.
· Phases are
Analysis.
Design.
Code.
Test.
· Incremental model delivers series of releases to customers called as increments.
· The first increment is called as core product. Here only the document
processing facilities are available.
· Second increment, more sophisticated document producing and processing

Notesengine.com

facilities are available.
· Next increment spelling and grammar checking facilities are given.
17
Merits
· This model can be adopted when there is less number of people involved in the project.
· Technical risks can be managed with each increment.
· For a very small time span, at least core product can be delivered to the customer.
RAD Model
· Rapid Application Development Model is the type of incremental model.
· Achieves the high speed development using component based construction.
Phases
· Business modeling
· Data modeling
· Process modeling
· Application generation.
· Testing and turnover**.**
3. Explain in detail about the software process.
· It is defined as the structured set of activities that are required to develop
the software system.
Fundamental activities
· Specification
· Design and implementation
· Validation
· Evolution
Common Process Framework
· **Process framework activities**
Communication
Planning
Modeling
Construction
Deployment.
· **Task Sets**
Defines the actual work to achieve the software objective.
· **Umbrella activities**
Software project tracking and control

Notesengine.com

Risk management
Software quality assurance
Formal technical reviews
Software configuration management
Work product preparation and production
Reusability management.
Measurement.
18

Capability Maturity Model(CMM)

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

Level 2:Repeatable – To track cost schedule and functionality basic 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.

4. Explain in detail about the life cycle process.

Fig: System engineering process

· System engineering process follows a waterfall model for the parallel development of different parts of the system.

System requirements definition

· Three types of requirements

Abstract functional requirements.

System properties.

Undesirable Characteristics.

· System objectives

· System requirement problem.

The system design process

Process steps

· Partition requirements

· Identify sub-systems.

· Assign requirements to sub-systems.

· Specify sub-system functionality.

· Define sub-system interfaces.

Requirement

Definition

definition

Notesengine.com

System
Design
Sub-system
Design
System
Integration
System
decommissioning
System
evolution
System
Installation
19

Sub-System development process

· After system design it starts.
· Involve use of COTS (Commercial-Off-The-Shelf).

System Integration

· It is the process of putting hardware, software and people together to make a
system.

System Installation

Issues are

· Environmental assumptions may be incorrect.
· There may be human resistance to the introduction of anew system.
· System may have to coexist with alternative systems for some period.
· There may arise some physical installation problems (e.g. cabling problem).
· Operator training has to be identified.

System evolution

· The lifetime of large systems is too long. They must evolve to meet change
requirements.
· The evolution may be costly.
· Existing systems that must be maintained are sometimes called as legacy systems.

System Decommissioning

· Taking the system out of service after its useful lifetime is called as System
Decommissioning.

Notesengine.com

5. Explain the prototyping approaches in software process.
Two approaches
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.
Evolutionary prototyping
Objective:
· The principal objective of this model is to deliver the working system to the
end-user.
· Example-AI systems.
Advantages
· Fast delivery of the working system.
· User is involved while developing the system.
· More useful system can be delivered.
· Specification, design and implementation work is co-ordinated manner.
Problems
· Management problems
· Maintenance problem
· Verification
20
Incremental Development
· After designing the overall architecture the system is developed and delivered in series of increments.
Throw-away prototyping
Objective:
· The principal objective of this model is to validate or to derive the system
requirements.
· It is developed to reduce requirement risks.
Advantages
· Requirement risks are very less.

Notesengine.com

Problems

· It can be undocumented.

· Changes made during the software development proceed may degrade
the system structure.

· Sometimes organizational quality standard may not be strictly applied.

6. Explain about rapid prototyping techniques.

Executable specification languages.

· Used to animate the system specification.

· It is expressed in a formal, mathematical language to provide a system
prototype.

Very high level languages.

· These are programming languages which include powerful data management facilities.

· They simplify program development.

Application generators and fourth-generation languages.

· These are successful languages because there is a great deal of communality across data processing applications.

7. Explain in detail about data modeling.

· Data modeling makes use of the ERD.

· Consists of 3 interrelated information.

The data object.

Attributes.

Relationships.

Cardinality and Modality

· Cardinality is the specification of the number of occurrences of one object
that can be related to the number of occurrences of another object.

· One-to-one cardinality.

· One-to-many cardinality.

· Many-to-Many cardinality.

· Modality of a relation is 0 if there is no explicit relationship or relation is
optional.

· Modality is 1 if an occurrence of relationship is mandatory.

Entity/Relationship Diagrams

· Components are
Data Objects.

Notesengine.com

21
Attributes.
Relationships.
Various type indicators.

8. Explain in detail about Functional Modeling.

· This model describes the computations that take place within a system.
· This model is useful when the transformation from the inputs to outputs is
complex.
· The functional model of a system can be represented by a data Flow
Diagram(DFD).

Data Flow Diagrams/Data Flow Graph/Bubble chart

· A DFD is a graphical representation that depicts the information flow and
the transforms that are applied as the data move from input to output.
· Level 0 DFD also called as fundamental system model or context model
represents the entire software as a single bubble with input and output data
indicated by incoming and outgoing arrows.
· Level 1 DFD contains 5 or 6 bubbles. Each bubbles can be refined at
Layers to depict more details.

Notesengine.com

Extensions to Real Time Systems

· Ward and Meller extensions
· Hatley and Pirbhai extension.

9. Explain in detail about Structural Modeling.

· Structural model includes a detail refinement of ERD,data flow model and
control flow model.
· Creating an ERD.
· Example: Safe Home Security System.
· Developing relationships and cardinality/Modality.
· Creating a data flow model using the guidelines.
· Creating a control flow model which describes the structural connection of
Processes
Control flows

Control stores.
· State automation
· Process activation table.

10. Explain in detail the design concepts.

Abstraction

· Functional abstraction
· Data abstraction
· Control abstraction

Information hiding

· Each module in the system hides the internal details of its processing activities and modules communicate only through over defined interfaces.

22

Structure

· It permits the decomposition of a large system into smaller, more manageable units with well defined relationships to the other units in a system.
· Network is the most general form of structure.

Hierarchical Structures/Structure Charts

· It depicts the structure of subroutines in a system, the data passed between routines, can be indicated on the arcs connecting routines.

Modularity

· Modular system consists of well-defined, manageable units with well defined interfaces among units.

Concurrency

· Independent processes that can be activated simultaneously if multiple processors are available.

Coupling and Cohesion

· **Data coupling –** The data coupling is possible by parameter passing or data interaction.
· **Control coupling –** The modules share related control data in control coupling.
· **Common coupling –** The common data or a global data is shared among modules.
· **Content coupling** – Content coupling occurs when one module makes use of data or control information maintained in another module.

11. Explain the design principles.

· The design process should not suffer from tunnel vision.
· The design should be traceable to the analysis model.

Notesengine.com

· Design should not reinvent the wheel.
· The design should minimize the intellectual distance between the software
and problem as it exists in the real world.
· The design should be structured to degrade gently, even when aberrant
data, events or operating conditions are encountered.
· Design is not coding, coding is not design.
· The design should be assessed for quality as it is being created, not after
the fact.
· The design should be reviewed to minimize conceptual (semantic) errors.

12. Explain the design steps of the transform mapping.

· Review the fundamental model.
· Review and refine the DFD for the software.
· Determine whether the DFD has the transform or transaction mapping.
· Isolate the transform center by specifying incoming and outgoing flow
boundaries.
· Perform first-level factoring.
· Perform second-level factoring.
23
· Refine the first iteration architecture using design heuristics for improved
software quality.

13. Explain the design steps in transaction mapping.

· Review the fundamental model.
· Review and refine the DFD for the software.
· Determine whether the DFD has the transform or transaction mapping.
· Identify transaction center and the flow characteristics along each of the
action paths.
· Factor and refine the transaction structure and the structure of each action
path.
· Refine the first iteration architecture using design heuristics for improved

Notesengine.com

software quality.

14. Explain in detail about the real time systems.

· Hard and soft real time systems.
· Real time and high performance.
· Real-Time control.
· Real time software design

Periodic Stimuli – Occur at predictable time intervals.
Aperiodic Stimuli – Occur regularly

15. Explain the types of software testing.

· Unit testing
· System testing
· Integration testing
· User-acceptance testing
· End-to-End testing
· Regression testing
· Exception testing
· Destructive testing

16. Explain in detail about Black box testing.

· Black box or behavioral testing focuses on the functional requirements
of the software.
· It is applied during the last stage of testing.
· Syntax driven testing is suitable for the specification which are described by a certain grammar.
· Decision table based testing is implemented when the original software
requirement have been formulated in the format of " if-then" statements.
· Liquid level control is the study of a simple control problem which is designed to check the liquid level. It has 2 sensors.
· Cause effect graphs in functional testing.

17. Explain about the software testing strategies.

· A strategic approach to software testing.
· Verification and Validation. ¾_Verification refers to the set of activities that ensure that software
24
correctly implements a specific function. ¾_Validation refers to a different set of activities that ensure that the
software that has been built is traceable to the customer requirements.

Notesengine.com

According to Boehm, ¾_Verification:" Are we building the product right?" ¾_Validation:" Are we building the right product?"
· Organization for software testing
· A software testing strategy.
· Criteria for completion of testing.

18. Explain in detail about Integration testing.

· It is a systematic technique for constructing the program structure.
· **Incremental integration –** The program is constructed and tested in
small increments.

Top-down integration

· It is an incremental approach.
· Modules are integrated by moving downward through the control hierarchy beginning with the main control module(main program).
· Subordinate modules are incorporated by depth-first or breadth-first manner.

Bottom-up integration

· This testing begins construction and testing with the components at the
lowest levels in the program structure.

Regression testing

· It is the re-execution of some subset of tests that have already been conducted to ensure the changes that have not been propagated unintended side effects.

Notesengine.com

Smoke testing

· It minimizes the integration risk.
· Error diagnosis and correction are simplified.

19. Explain in detail about system testing.

· System testing
· Stress testing
· Security testing.
· Performance testing

20. Explain in detail about SCM.

· Software Configuration Management is an umbrella activity that is applied throughout the software process.

SCM Activities

· Identify change.
· Control change.
· Ensure the change is properly implemented.
· Report change to others.

Need for SCM

25

· When you build computer software change happens, you need to control it effectively.

SCI

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

21. Explain about software cost estimation.

· Major factors are:

Programme ability.

Product Complexity.

Product size.

Available time.

Required reliability.

22. Explain in detail about COCOMO model.

· Constructive Cost Model.

· Software cost estimation gives the estimation of how much months a
man take to develop a software product.

· Application Composition Model.

· Early design stage model

· Post-architecture stage model.

· COCOMO II application composition uses object points.

· NOP=(object point)X[100-%reuse)/100]

NOP-New Object Point.

· Productivity Rate, PROD=NOP/person-Month.

23. Explain in detail about Delphi Method.

Procedure

· The co-ordinator presents a specification and estimation form to each
expert.

· Co-ordinator calls a group meeting in which the experts discuss estimation issues with the coordinator and each other.

· Experts fill out forms anonymously.

· Co-ordinator prepares and distributes a summary of the estimates.

· The Co-ordinator then calls a group meeting.In this meeting the experts mainly discuss the points where their estimates vary widely.

· The experts again fill out forms anonymously.

Vtuesengine.com

· Again co-ordinator edits and summarizes the forms,repeating steps 5
and 6 until the co-ordinator is satisfied with the overall prediction synthesized from experts.

24. Explain in detail about software Maintenance.

· Software maintenance is an activity in which program is modified after it has been put into use.
· 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.

Different aspects of maintenance

· The bug-fixing view
26
· The need-to-adapt view
· The user-support view

Need for software maintenance

· To provide continuity of service
· To support mandatory upgrades.
· To support user requests for improvements.
· To facilitate future maintenance work.

25. Explain about CASE tools.

· Computer Aided Software Engineering.
· Business Process Engineering Tools.
· Process Modeling and Management Tools.
· Project Planning Tools.
· Risk analysis tools.
· Project management tools.
· Requirement tracing tools.
· Metrics and management tools.
· Documentation tools.
· System Software tools.
· Quality Assurance tools.
· Database management tools.
· SCM tools.
· Analysis and design tools.
· PTO/SIM tools.
· Interface design and development tools.
· Prototyping tools.
· Programming tools.
· Web development tools.

Noteengine.com

· Integration and testing tools.
· Static analysis tools.
· Dynamic analysis tools.
· Test management tools.
· Client/server tools.
· Re-engineering tools.

Notesengine.com