VETRI VINAYAHA COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CS6403 SOFTWARE ENGINEERING
II year/ IV sem CSE
(Regulation 2013)

UNIT 1- SOFTWARE PROCESS AND PROJECT MANAGEMENT

PART A

1.Define 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?

* Software is engineered, not manufactured.
* Software does not wear out.
* Most software is custom built rather than being assembled from components.

4. What are the various categories of software?

* System software
* Application software
* Engineering/Scientific software
* Embedded software

5. What are the challenges in software?

* Copying with legacy systems.
* Heterogeneity challenge
* 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?

* Specification
* Design and implementation
* Validation
* Evolution

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

* Software project tracking and control.
* Risk management.
* Software Quality Assurance.
* Formal Technical Reviews.
* Software Configuration Management.

* Work product preparation and production.
* Reusability management.
* Measurement.

9. What are the merits of incremental model?

i) The incremental model can be adopted when there is 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.

* Customer communication - it is suggested to establish customer communication.
* Planning – All planning activities are carried out
* Risk analysis – The tasks required to calculate technical and management risks.
* Engineering – tasks required to build one or more representations of applications
* Construct and release – tasks required to construct, test, install the applications
* Customer evaluation - tasks are performed and implemented at installation stage based on the customer evaluation.

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. 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.

13. 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".

14. What does Verification represent?

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

15. What does Validation represent?

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

16. What is the use of CMM?

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

17. Name the Evolutionary process Models.

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

18.What is meant by Software engineering paradigm?

The development strategy that encompasses the process, methods and tools and generic phases is often referred to as a process model or software engineering paradigm.

PART B

1. Describe evolutionary process models.

2. Explain the various phases of software development life cycle and identify deliverables at each phase.

3. What is prototyping? Explain the types of prototyping.

4. Explain Component Based Development model in detail

5. ExpalinRADmodel.

6. Explain about Project Scheduling and its methods.

7.How is earned value computed to assess the progress?

8.Explain COCOMO model for estimation.

UNIT – II REQUIREMENTS ANALYSIS AND SPECIFICATION

1. 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.

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

i. Source traceability – These are basically the links from requirement to stakeholders
ii. Requirements traceability – These are links between dependant requirements.
iii. Design traceability – These are links from requirements to design.

3. Define software prototyping.

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

4. 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.

5. What are the prototyping approaches in software process?

i. Evolutionary prototyping – the initial prototype is prepared and it is then refined through number of stages to final stage.

ii. Throw-away prototyping – a rough practical implementation of the system is produced. The requirement problems can be identified from this implementation.

6. 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-ordinate manner.

7. 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 should 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?

8. 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.

9. What is a data object?

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

10. What is cardinality in data modeling?

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

11. What is ERD?

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

12. 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.

13. What does Level0 DFD represent?

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.

14. 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.

15. 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.

16. 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.

17. What are functional requirements?

Functional requirements are” statements of services the system should provide how the system should react to particular input and how the system should behave in particular situation.

18. What are non functional requirements?

Non functional requirements are constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.

19. What is the outcome of feasibility study?

The outcome of feasibility study is the results obtained from the following questions:

- x Which system contributes to organizational objectives?
- x Whether the system can be engineered? Is it within the budget?
- x Whether the system can be integrated with other existing system?

20.What is meant by structural analysis?

The structural analysis is mapping of problem domain to flows and transformations. The system can be modeled by using Entity Relationship diagram, Data flow diagram and Control flow diagrams.

PART B

1.Narrate the importance of software specification of requirements, explain typical SRS structure and its parts.

2.Explain the feasibility studies. What are the outcomes? Does it have either explicit or implicit effects on software requirement collection?

3.Explain feasibility studies.

4.Explain Data Dictionary.

5.State its process and explain requirements elicitation and analysis.

6.Explain about structured system analysis (DFD).

7.Explain the basic concepts of software design.

8.Write about the following requirement engineering activities.

(i) Inception.

(ii) Elicitation.

(iii) Elaboration.

(iv) Negotiation

(v) Specification

(vi) Validation

(vii) Requirements management

UNIT III SOFTWARE DESIGN

1. What are the elements of design model?

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

2. Define design process.

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

3. 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.

4. What is the benefit of modular design?

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

5. **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.

6. 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.

ii. Logically cohesive – A module that performs the tasks that are logically related with each other.

iii. Temporal cohesion – The module in which the tasks need to be executed in some specific time span.

iv. Procedural cohesion – When processing elements of a module are related with one another and must be executed in some specific order.

v. Communicational cohesion – When the processing elements of a module share the data then such module is called communicational cohesive.

7. **What is coupling?**

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

8. 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 modules.

iv. Content coupling – Content coupling occurs when one module makes use of data

9. 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.

10. 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 side effects.
iv. Software that is easier to extend.

11. What is vertical partitioning?

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

12. What are the advantages of vertical partitioning?

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

13. 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.

14. List the guidelines for data design.

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.

15. 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.

16. 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.

17. 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.

18. What is an Architectural design?

The architectural design defines the relationship between major structural elements of the software, the "design patterns" that can be used to achieve the requirements that have been defined for the system.

19. What is data design?

The data design transforms the information domain model created during analysis into the data structures that will be required to implement the software.

21. What is interface design?

The interface design describes how the software communicates within itself, with systems that interoperate with it, and with humans who use it.

22. What is component level design?

The component level design transforms structural elements of the software architecture into a procedural description of software components.

23. What is software design?

Software design is an iterative process through which the requirements are translated into a "blueprint" for constructing the software.

24. What is user interface design?

User interface design creates an effective communication medium between a human and a computer.

25. What is system design?

System design process involves deciding which system capabilities are to be implemented in software and which in hardware.

PART B

1. What is transform mapping? Explain the process with an illustration. What is its strength and weakness? (16)
2. a) Explain about the various design concepts considered during design? (12)
   b) Write short notes on user interface design process? (4)
3. a) Explain data architectural and procedural design for a software? (8)
   b) Describe the design procedure for data acquisition system (8)
4. Explain the importance of user interface design in sale of software. (16)
5. Describe decomposition levels of abstraction and modularity concepts in software design? (16)
6. What are the characteristics of a good design? Describe different types of coupling and cohesion. How design evaluation is performed? (16)
7. Draw the basic structure of analysis model and explain each entity in detail.(16)
8. a) discuss in detail about the design process in software development process
   b) Justify "Design is not coding and coding is not design". (8)
9. a) Explain in detail about the characteristics and criteria for a good design.

10 a) Describe the golden rules for interface design. (6)
   b) What is the design document? c) How is it organized?

11. What are the various software architectures available for the developer according to you? which is the best and why?
12. What do you mean by modularity in software development? Why is it needed? What is its strength? (8)
13. a) What are the various model of abstraction? Discuss any two in detail? (8)

b) How does a real time system design differ comparing distributed system design? (8)

14. a) Explain the set of principles for software engineering design? (10)
    b) Describe the concept of information hiding. (6)

UNIT -IV TESTING AND IMPLEMENTATION

1. Define software testing?

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

2. What are the objectives of testing?

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.

3. 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.

4. What are the two levels of testing?

i. Component testing - Individual components are tested. Tests are derived from developers 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.

5. What are the various testing activities?

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

6. 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.

7. 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.

8. 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.

9. 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.

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.

10. **What is cyclomatic complexity?**

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

11. **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 (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.

12. Distinguish between verification and validation.

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.

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

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

14. Write about drivers and stubs.

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 subprogram that uses the module interfaces and performs the minimal data manipulation if required.

15. 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.

16. What are the advantages and disadvantages of big-bang? Advantage:

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.

17. What are the benefits of smoke testing?

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

18. Distinguish between alpha and beta testing

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 developers 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 customers site.

19.What are the various types of system testing?

1. Recovery testing – is intended to check the systems 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.

20. Define debugging.

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

21. What are the common approaches in debugging?

<u>Brute force method</u>:

The memory dumps and run-time tracks are examined and program with write statements is loaded to obtain clues to error causes.

<u>Back tracking method</u>:

The source code is examined by looking backwards from symptom to potential causes of errors.

<u>Cause elimination method</u>:

This method uses binary partitioning to reduce the number of locations where errors can exists

22. What is meant by structural testing?

In structural testing derivation of test cases is according to program structure. Hence knowledge of the program is used to identify additional test cases.

23. What is meant by regression testing?

Regression testing is used to check for defects propagated to other modules by changes made to existing program. Thus, regression testing is used to reduce the side effects of the changes.

24. What is meant by unit testing?

The unit testing focuses verification effort on the smallest unit of software design, the software component or module.

<u>PART B</u>

1. Discuss the differences between black box and white box testing models. Discuss how these testing models may be used together to test a program schedule.
2. What do you mean by system testing? Explain in detail.
3. Justify the importance of testing process. b) Discuss in detail about alpha and beta testing.
4. What do you mean by integration testing? Explain their outcomes:
5. What is black box testing? Is it necessary to perform this? Explain various test activities:
6. Explain the integration testing process and system testing process and discuss their

outcomes:

7. What do you mean by system testing? Give a case study of a system testing for operating system?

8 What do you mean by boundary value analysis? Give two examples of boundary value testing.

9. Explain black box testing methods and its advantages and disadvantages.

10. a) Explain the testing procedures for boundary conditions.
 b) Describe verification and validation criteria for a software.

11. a) Describe unit testing and integration testing. How test plans are generated?
 b) Suggest software testing sequence for a 100% bug free software. Explain.

12. Discuss software failures and faults? What are test coverage criteria? Discuss testing issues.

13. Explain automated testing tools. How test cases are generated? Discuss when to stop testing? What is performance testing? Describe.

14. What are the various testing strategies to software testing? Discuss them briefly.

UNIT – V PROJECT MANAGEMENT

1. Define measure and metrics.

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

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

2. What are the types of metrics?

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.

3.. 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.

4. Give the procedure of the Delphi method.

1. The co-coordinator presents a specification and estimation form to each expert.
2. Co-coordinator 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-coordinator prepares and distributes a summary of the estimates.
5. The Co-coordinator then calls a group meeting.

5. 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.

6. What is EVA?

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.

7. Define maintenance.

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

8. What are the types of software maintenance?

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.

9. 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).

10. What are the types of static testing tools?

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 test cases as per user requirements.

11. What is meant by Software project management?

Software project management is an activity of organizing, planning and scheduling software projects.

12. What is meant by software measurement?

Software measurement means deriving a numeric value for an attribute of a software product or process.

13. What is meant by software cost estimation?

The software cost estimation is the process of predicting the resources required for software development process.

14. What is meant by CASE tools?

The computer aided software engineering tools automatic the project management activities, manage all the work products. The CASE tools assist to perform various activities such as analysis, design, coding and testing.

15. What is meant by Delphi method?

The Delphi technique is an estimation technique intended to active a common agreement for estimation efforts.

16. Derive ZIP‟s law.

ZIP‟s first law of the form, fr r = C (or)

$n_{r = Cn / r}$

C - constant

r - rank for tokens

fr- frequency of occurrence

17. What is software configuration management (SCM)?

Software configuration management is the art of identifying, organizing, and controlling modifications to the software being built by a programming team.

18. What is meant by risk management?

Risk management is an activity in which risks in the software projects are identified.

19. What is meant by software project scheduling?

Software project scheduling is an activity that distributes estimated effort across the planned project duration by allocating the effort to specified software engineering tasks.

PART B

1. Explain the need for software measures and describe various metrics.
2. Discuss briefly on software maintenance activities and how do you estimate the cost involved.
3. a) Explain in detail about the maintenance process. b) Discuss in detail about software evolution.
4. Describe two metrics which are used to measure the software in detail. Discuss clearly the advantages and disadvantages of these metrics.
5. a) What is Halsted's software science metric. Define. b) Explain about function point metric in detail.
6. Write short notes on a) Software maintenance b) Task scheduling with an example.
7. Explain various cost estimation models and compare.
8. Write briefly on a) CASE (8) b) Software complexity measure.
9. Explain the maintenance activities and maintenance problems. How the cost of maintenance is estimated?
10. Write short notes on COCOMO estimation criteria.
11. a) Justify the statement "Software maintenance is costlier".
    b) Discuss the concept of software maintenance process.