

CCS356 – OBJECT ORIENTED SOFTWARE ENGINEERING

UNIT I

Introduction to Software Engineering, Software Process, Perspective and Specialized Process Models –Introduction to Agility-Agile process-Extreme programming-XP Process-Case Study.

Part A

1. Define a system and computer based system.
2. Write the IEEE definition of software engineering.
3. What is Software Engineering? /Define software engineering.
4. What is Software? List the characteristics.
5. What are two types of software products?
6. Mention the characteristics of software contrasting It with characteristics of hardware.
7. Write any two software engineering challenges.
8. Software doesn't wear out. Justify.
9. What is software processes. / Distinguish between process and methods.
10. Define the terms product and process in software engineering.
11. Why software architecture is important in software process?
12. Draw Common Process Framework.
13. If you have to develop a word processing software product, what process model will you choose? Justify your answer.
14. Depict the relationship between work product, task, activity and system.
15. State the benefits and drawbacks of waterfall life cycle model for software development.
16. List two deficiencies in waterfall model. Which process model do you suggest to overcome each efficiency?
17. What are the pros and cons of Iterative software development models?
18. Which process model leads to software reuse? Why?
19. What are the phases encompassed in the RAD model?
20. What are the Drawbacks of rapid application development?
21. Define Evolutionary Process Model.
22. Compare prototyping approaches in a software process.
23. What are the advantages of prototyping model?
24. What is meant by blocking states in linear sequential model?
25. How does "Project Risk" factor affect the spiral model of software development?

Part – B

1. Define software life cycle. List all life cycle models and explain all the models in detail with neat diagram. (13)
2. Describe about agile modeling in details.
3. Discuss the Extreme Programming Process.
4. What is agility in the context of software engineering work?
5. (i) Compare the life cycle models based on their distinguishing factors, strengths and weaknesses. (7)
(ii) Discuss the prototyping model, what is the effect of designing prototype on the overall cost of the software project? (6)
6. What is process model? Describe the process model that you would choose to manufacture a car. Explain by giving suitable reasons.
7. Explain the component based software development model with a neat sketch.
8. What are some of the issues that leads to an XP debate?
9. Describe the various Evolutionary process models with neat diagram.
10. List the principles of agile software development.

Part C

1. For the scenario describe below, which life cycle model would choose? Give the reason why you choose this model. You are interacting with the MIS department of a very large oil company with multiple departments. They have a complex legacy system. Migrating the data from this legacy system is not easy task and would take a considerable time. The oil company is very particular about processes, acceptance criteria and legal contracts.
2. Describe at least one scenario where.
 1. RAD model would be applicable and not the waterfall model.
 2. Waterfall model is preferable to all other models
3. Assume that you are the technical manager of a software development organization. A client approached you for a software solution. The problems stated by the client have uncertainties which lead to loss if it not planned and solved. Which software development model you will suggest for this project – justify. Explain that model with its pros and cons with neat sketch.
4. Explain how waterfall model is applicable for the development of the following systems:
 - a) University accounting system
 - b) Interactive system that allows railway passengers to find time and other information for the terminals installed in the station.
5. Provide three examples of software projects that would be amenable to incremental model. Be specific.

UNIT II

Requirement analysis and specification – Requirements gathering and analysis – Software Requirement Specification – Formal system specification – Finite State Machines – Petrinets – Object modelling using UML – Use case Model – Class diagrams – Interaction diagrams – Activity diagrams – State chart diagrams – Functional modelling – Data Flow Diagram- CASE TOOLS.

Part A

1. Write distinct steps in requirement engineering process.
2. Why SRS Must be traceable? What is traceability?
3. What are non-functional requirements for a software?
4. What is the outcome of feasibility study?
5. Differentiate data flow diagram and state transition diagram.
6. Why it is so difficult to gain a clear understanding of what customer wants?
7. Identify ambiguities and omissions in the functional requirements.
What questions would you ask to clarify these functional requirements?
8. List out requirements engineering.
9. Define functional and non- functional requirements.
10. Give two examples of non functional requirements.
11. What is the need for feasibility analysis?
12. Define feasibility study and list the types.
13. List the characteristics of good SRS.
14. What is the purpose of petrinet?
15. Differentiate between normal and exciting requirements.
16. Draw a use case diagram for an online shopping which provide provisions for registering, authenticating the customers and also for online payment through any payment gateway like paypal.
17. Define Quality function development(QDF)
18. Draw the context flow graph of a ATM automation system.
19. State two advantages of using petri nets.
20. How does data flow diagram help in design of software system?
21. Define UML.
22. What are all the various types of diagram that can be drawn in UML?
23. When to use usecase diagram?
24. Define activity diagram?
25. What is the use of system sequence diagram?

Part – B

1. Explain in detail about the functional and non functional requirements. Explain the software requirement engineering process with neat diagram.
2. Explain the feasibility studies. What are the outcomes? Does it have implicit or explicit effects on the software requirement collection?
3. What is requirements elicitation? Briefly describe the various activities performed in requirement elicitation phase with an example of a watch system that facilitates to set time and alarm.
4. What are the components of the standard structure for the software requirement document? Explain in detail. (Or) Show the template of IEEE standard software requirement document.
5. Explain Petri Net in details. Draw a Petri Net that depicts the operation of an “Automated Teller Machine”. State the functional requirements you are considering.
6. Explain with an example, how use case modeling is used to describe functional requirements. Identify the actors, scenarios and use cases for example.
7. Describe the strategies used to identify the conceptual classes.
8. Describe the steps to create a domain model used for representing the conceptual classes.
9. What are system sequence diagram? What is the relationship between SSDs and Use cases? Explain with an example.
10. Explain the state chart diagram with a suitable example. Also define its components and use.

Part – C

1. Develop an online railway reservation system, which allows the user to select route, book/cancel tickets using net banking/ credit /debit cards. The site also maintains the history of the passengers. For the above system, list and draw the use case scenario and model the above specification.
2. Develop the software requirement document for the following requirement. A coffee vending machine serves coffee to customers. A customer can choose a type of coffee among a list of options, supply the amount required and get served. Each coffee is prepared by adding units of hot water, coffee powder, milk and sugar. The recipe for each coffee is stored. Develop a usecase diagram, Activity diagram and Sequence diagram.
3. Model a class diagram for “Banking System” State the functional requirements you are considering.
4. Draw Sequence and collaboration diagram for online course reservation system.

5. Develop the process of ordering a pizza over the phone. Draw the use case diagram and also sketch the activity diagram representing each step of the process, from the moment you pick up the phone to the point where you start eating the pizza. Include activities that others need to perform. Add exception handling to the activity diagram you developed. Consider at least two exceptions (e.g. delivery person wrote down wrong address, deliver person brings wrong pizza).

UNIT III

Software design – Design process – Design concepts – Coupling – Cohesion – Functional independence – Design patterns – Model-view-controller – Publish-subscribe – Adapter – Command – Strategy – Observer – Proxy – Facade – Architectural styles – Layered - Client Server - Tiered - Pipe and filter- User interface design-Case Study.

Part – A

1. Define software design.
2. Describe software design process.
3. What are the various models produced by the software design process?
4. What are the Characteristics of Good Design? What are the steps involved in design stage of a software?
5. What is the quality attributes of software design? Mention the design quality model proposed by Hewlett Packard.(or) Write a note on FURPS model of design quality.(or) What are the design qualities attributes 'FURPS' meant?
6. Define software design concept.
7. What are certain issues that are considered while designing the software?
8. Define data abstraction.
9. Name the levels of abstraction, which are in practice for the design.
10. Define Modularity.
11. Modularity is the single attribute of the software that allows a program to be intellectually manageable" - How this is true?
12. How can we evaluate a design method to determine if it will lead to effective modularity?
13. Why modularity is important in software projects?
14. State the guidelines for modular design.
15. Define Architecture.
16. What are the architectural design various system models can be used?
17. Define Refinement.
18. In what way abstraction differs from refinement?
19. Define Pattern.
20. Define Cohesion.
21. If a module has logical cohesion what kind of coupling is this module likely to have with others?
22. What are the Different types of cohesion?
23. Define Coupling.

24. What are the Various types of coupling?

25. Define Refactoring.

Part – B

1. Explain the following list of design concept
 - a. Abstraction
 - b. Patterns
 - c. Functional independence
2. Describe the concept of coupling and cohesion.
3. Write short note on structural pattern.
4. Write short note on adapter pattern.
5. Explain architecture styles of
 - a. Client server
 - b. Tiered architecture
 - c. Layered architecture
6. Explain in detail about golden rules for user interface design.
7. Discuss about pipe and filter architectural pattern.
8. Explain façade design pattern with implementation.
9. Explain strategy design pattern for any scenario with neat class diagram.
10. With suitable example discuss command pattern.
11. Write a note on observer and proxy.

Part – C

1. Explain user interface analysis and design.
2. Summarize characteristics of good design, quality guideline and quality attributes.
3. Explain modularity concept in software design.
4. Explain core activities involved in user interface design process with necessary block diagram.

UNIT IV

Testing – Unit testing – Black box testing– White box testing – Integration and System testing– Regression testing – Debugging - Program analysis – Symbolic execution – Model Checking- Case Study

Part – A

1. What is software testing?
2. List the objectives of testing.
3. When do you stop testing? Justify your answer.
4. Why testing is important?
5. What are the principles of testing?
6. What is test case?
7. Define regression testing?
8. Difference between testing and debugging.
9. Write short note on debugging techniques.
10. Why debugging is so difficult?
11. Difference between black and white box testing.
12. What are the attributes of good test?
13. Define cyclomatic complexity.
14. List the errors uncovered by black box testing.
15. Write the type of system test.
16. List the errors identified during unit testing.
17. Calculate cyclomatic complexity for the following program,

```
int temp (a>b)
temp a
else temp b
if (c>temp) t
emp=c
return temp
```
18. What are the classes of loops that can be used?
19. Outline the need for system testing.
20. Difference between alpha testing and beta testing.
21. Mention the purpose of stubs and driver in testing?
22. Distinguish between verification and validation.
23. How will you test simple loop?

24. What is static program analysis?

25. What is model checking?

Part – B

1. Explain unit testing.
2. Explain equivalence partitioning techniques with suitable example.
3. Write short note on
 - a. Regression testing
 - b. Smoke testing
4. Explain various systems testing strategies.
5. Discuss about validation testing methods.
6. Write about brute force and backtrack method in debugging techniques.
7. With suitable example explain boundary value analysis.
8. Write short note on symbolic execution.
9. Explain how program analysis can be carried out?
10. With an example explain loop testing.

Part C

1. Describe black box testing
2. Consider the pseudocode for simple subtraction given below Program 'simple subtraction'
Input (x,y)Output(y)
If $x > y$ then
DO $x - y = z$
else $y - x = z$
endif
output(z)
output 'End Program'
3. perform the basic path testing
4. Discuss about integration testing technique with example.
5. Write about testing objective, principles? Justify why testing is important.
6. Write a program for sorting of n numbers. Draw the flowchart, flowgraph and find cyclomatic complexity.

UNIT V

Software Project Management- Software Configuration Management - Project Scheduling- DevOps: Motivation-Cloud as a platform-Operations- Deployment Pipeline:Overall Architecture Building and Testing-Deployment- Tools- Case Study

Part A

1. Write a short note on empirical estimation models.
2. What is the standardization for the software metrics?
3. What are project indicators and how do they help a project manager?
4. Define basic equation for the effort estimation models.
5. List out few process and product metrics.
6. Define software measure.
7. Distinguish between direct and indirect measures of metrics.
8. Differentiate between size oriented and function oriented metrics.
9. How to measure the function point (FP)?
10. List out the different approaches to size of the software.
11. Mention difference between organic mode and embedded mode in cocomo model.
12. Name the metrics for specifying non-functional requirements.
13. An organic software occupies 15,000 LOC. How many programmers are needed to complete?
14. Differentiate between size oriented and function oriented metrics.
15. State the advantages and disadvantages in LOC based cost estimation.
16. What is scheduling?
17. State the importance of scheduling activity in project management.
18. What is error tracking?
19. What are the issues in measuring the software size using LOC as metric?
20. List a few process and project metrics.
21. How productivity and cost is are related to function points?
22. What are the different types of productivity estimation measures?
23. List out the principles of project scheduling.
24. List two advantages of COCOMO model.
25. If team A found 342 errors prior to the release of software and team B found 182 errors. What additional measures and metrics are needed to find out if the teams have removed the errors

effectively?

26. What is EVA?
27. Bring the importance between COCOMO I and II
28. State any two project scheduling techniques.

Part – B

1. Explain the role of people, product and process in project management.
 - (i) What are the categories of stakeholders? What are the characteristics of effective project manager?
 - (ii) Explain W5HHH principle.
2. Explain the project estimation in detail with the example.
3. Discuss the process of function point analysis. Explain function point analysis with sample cases for components of different complexity.
4. Describe in detail COCOMO model for software cost estimation. Illustrate considering a suitable example.
5. Explain in detail about project scheduling with the example.
6. Explain the overall architecture of DevOps?
7. List the features of LOC and FP based estimation models. Compare the two models and list the advantages of one over other.
8. Discuss how Make/Buy decision helps to track a project quantitatively.
9. What is configuration management repository? Discuss role and features of SCM repository.

Part C

1. Suppose you have budget cost of a project \$9,00,000. The project is to be completed in 9 months. After a month you have completed 10% of the project at the total expenses of \$100000. The planned completion should have been 15%. You need to determine whether the project is on-time and on-budget? Use earned value analysis approach and interrupt.
2. Suppose you are managing a software development project. The project is expected to be completed 8 month at the cost of \$10000 per month. After 2 months you realize the project is 30% completed at the cost of \$40000. You need to determine whether the project is on time and on budget after 2 months?
3. Compute the function point FP for a payroll program that reads a file of employees and file of information for the current month and prints cheques for all the employees. The program is capable of handling an interactive command to print an individually requested cheque immediately