Question Bank

Subject: Software Engineering **Subject Code: CSC 502**
Semester: V **Rev-2019 CBCS**

CO 1: Able to select appropriate model of software process using basic fundamentals of software engineering

Syllabus Content: Process model

Software Engineering-process framework, the Capability Maturity Model (CMM), Prescriptive Process Models: The Waterfall, Incremental Process Models, Evolutionary Process Models: RAD & Spiral, Agile process model: Extreme Programming (XP), Scrum, Kanban

Sample Questions:

1. What is software?
2. Justify with a neat failure curve that "software doesn't wear out but it deteriorates".
3. List down the broad categories of computer software.
4. With a detailed diagram summarize the concept of software engineering as a layered approach in the development of software.
5. What is a software process and elaborate on each of the software engineering framework activities?
6. Differentiate between software process and software engineering.
7. Quote the different umbrella activities that work in conjunction with the software framework activities.
8. The CMMM model describes a five-level evolutionary path of increasingly organized and systematically more mature processes. Therefore, elaborate on all of these five levels.
9. Differentiate between incremental and evolutionary models.
10. Compare the Spiral process model with the waterfall model in terms of feasibility in growing requirements
11. What features a software project should possess to apply incremental and evolutionary models
12. Justify with appropriate reasons why you will use the spiral process model for rapid prototyping
13. Considering the case study for canteen ordering system of a college, identify which model can be best suited.
14. Differentiate between incremental and evolutionary models.
15. What is agility?
16. Briefly paraphrase the various agile models. (Scrum, Kanban, XP)
17. Compare and contrast Scrum model with the Kanban model.
18. Write an XP user story that describes the Google search feature on many Web browser home pages.

Co2: Design and analyze the requirement models.

Syllabus Content: Module: Requirement Modelling

Requirement Engineering, Requirement Modelling, Data flow diagram, Scenario-based model, Software Requirement Specification document format (IEEE)

Sample Question:

1. Summarize seven steps in requirement engineering.
2. You have been given the responsibility to elicit requirements from a customer who tells you he is too busy to meet with you. What should you do?
3. Identify some of the problems that occur when requirements must be elicited from three or four different customers.
4. Draw a use case diagram for Library Management System.
5. Construct a class diagram for an Automatic Attendance Monitoring System that shows the relationship and interaction between set of objects and classes.
6. Develop a state diagram for any dynamic class of Railway Reservation System.
7. Inspect Why SRS is known as the back-box specification of the system?
8. Identify characteristics of a good software requirement specification document?
9. Prepare an SRS document for an e-Administration of computer Labs that highlights all functional and non-functional requirements of the system.
10. It's said that the analysis model act as a bridge between 'system description' and the 'design model'. Inspect this statement with at least 5 reasons

CO3: Identify the size of the project, schedule, estimate the cost and track the progress of the software project.

Syllabus Content: Module: Software Estimation Metric:

Software Metrics, Software Project Estimation (LOC, FP, COCOMO II), Project Scheduling & Tracking

Sample Question:

1. Give examples of where Metrics are used Issues with Software Metrics
2. Outline major parts of the Software Project Plan
3. Provide size, schedule & cost estimate to support user requirements and project charter.
4. Differentiate between FP-based & LOC-based cost estimation techniques.
5. Elaborate COCOCMO method of cost estimation
6. Use multiple techniques to estimate software tasks, projects, and products.
7. Outline the task set required for planning the Project on an online ordering system.
8. Select an appropriate task set for the OLCRS project.
9. Estimate cost and effort for Library Management System
10. Develop an initial project schedule for Library Management System
11. Estimate cost and effort for University Management System
12. Develop an initial project schedule for University Management System
13. Develop a list of software characteristics that affect the complexity of a project. Prioritize the list.
14. Performance is an important consideration during planning. Discuss how performance can be interpreted differently depending upon the software application area.
15. Using a scheduling tool (ganttpro), develop a timeline chart for a software project 'online course registration systems.

16. Compare and contrast automated estimation tools over decomposition tool. Give reasons why automated estimation tools implement one or more empirical models or decomposition techniques.
17. Do a functional decomposition of the Library management software describe in problem
18. Estimate the size of each function in LOC of the Library management software
19. Assume that a university has contracted you to develop an online course registration system (OLCRS). First, act as the customer (if you're a student, that should be easy) and specify the characteristics of a good system. (Alternatively, your instructor will provide you with a set of preliminary requirements for the system.) Using the estimation methods discussed in this Module, develop an effort and duration estimate for OLCRS. Suggest how you would:
    a. Define parallel work activities during the OLCRS project.
    b. Distribute effort throughout the project.
    c. Establish milestones for the project.
20. Using a scheduling tool develop a time-line chart for the OLCRS project.
21. Determine the various ways how tracking on a task set can be accomplished for a given project.

CO4: Demonstrate various design principles to develop software projects.
Syllabus Content: Module: Software Design: Design Principles & Concepts Effective Modular Design, Cohesion and Coupling, Architectural design

Sample Questions:

1. State the various design patterns that can be used for the software development.
2. Using the architecture of a house or building as a metaphor, draw comparisons with software architecture.
3. How are the disciplines of classical architecture and the software architecture similar? How do they differ?
4. Once software requirements have been analysed, software design is the last engineering activity within the modelling activity. Therefore, with neat diagram demonstrate the translating of the requirements model into the design model.
5. Consider the following interactive application and develop a detailed class diagram for the same. -An automated course registration system for a university.
6. Consider the following interactive application and develop a detailed collaboration diagram for the same. -A Fast food billing system.
7. Apply a stepwise refinement approach to develop three different levels of procedural abstractions for a check writer that will print the amount in words.
8. Exemplify the relationship between the concepts of information hiding as an attribute of effective modularity.
9. Compare the concepts of cohesion and coupling with respect to modular designing.
10. Justify why cohesion is more preferred than coupling in modular designing of the software.
11. Suggest a design pattern that you may encounter in using an automobile.
12. Suggest a design pattern that you may encounter in using an electronic appliance like a Television.

CO5: Develop software architecture using various design principles.

Syllabus Content: Module: Software Testing

Unit testing, Integration testing, Validation testing, System testing, Testing Techniques, white-box testing: Basis path, Control structure testing black-box testing: Graph based, Equivalence, Boundary Value 5.3 Types of Software Maintenance, Re-Engineering, Reverse Engineering.

Sample Question:

1. What is Software testing?
2. What are the testing strategies?
3. Identify the role of testing in software development?
4. Differentiate between white box and black box testing
5. Specify the principles of software testing?
6. What is regression testing in software testing?
7. Mention the characteristics of a good unit test
8. Compare black-box testing, white-box testing, and grey-box testing.
9. State the difference between bugs and errors
10. List some of the popular software testing tools/frameworks, providing a brief description of each.
11. What is A/B testing?
12. What are defects in software testing?
13. Specify features of sanity testing in software testing?
14. How much testing is sufficient? Or, is it possible to do exhaustive testing of the software?
15. Why developers shouldn't test the software they wrote?
16. What is the software testing life cycle?
17. State the difference between verification and validation in software testing.
18. What is a Test Plan? What does it include?
19. Develop a test plan for testing the architectural prototype of the Course Registration System. This Test Plan document supports the following objectives:
    a) Identify existing project information and the software that should be tested.
    b) List the recommended test requirements (high level).
    c) Recommend and describe the testing strategies to be employed.
    d) Identify the required resources and provide an estimate of the test efforts.
    e) List the deliverable elements of the test activities.
21. Develop a complete test strategy for the Hall Ticket generation system. Document it in a Test Specification.
22. Is it possible to test every configuration that a WebApp is likely to encounter on the server side? If it is not, how do you select a meaningful set of configuration tests?
23. Using the BPR model elaborate various elaborate the six activities related with software maintenance.
20. Specify the steps involved in Re-engineering.
21. Mention Reverse Engineering Goals
22. Study advantages and disadvantages of Re-engineering.

23. There is a subtle difference between reengineering and reverse engineering. What is it?
24. Research the literature and/or Internet sources that debates on the case studies of mainframe to client-server reengineering.
25. Explore the inventory analysis checklist of an organization's website. Develop a quantitative software rating system that could be applied to existing programs software. This process achieves reengineering. Also analyse the cost benefit model to perform reengineering.

CO6: Analyze the risks for mitigation for quality management of software.

Syllabus Content: Module: Software Configuration Management, Quality Assurance and Maintenance

Risk Analysis & Management: Risk Mitigation, Monitoring and Management Plan (RMMM). Quality Concepts and Software Quality Assurance Metrics, Formal Technical Reviews, Software Reliability, The Software Configuration Management (SCM), Version Control and Change Control

Sample Question:

1. Note activities in Risk Analysis & Management: Risk Mitigation, Monitoring, and Management Plan
2. Specify Different types of risks in Software Project Development
3. Identify Software Risk Management Activities
4. Study risk prioritization
5. What are the key objectives of the RMMM plan?
6. What are the drawbacks of the RMMM plan?
7. Create a risk table for the software that is developing "next generation" word –processing software. Prioritize all the risks depending on their impact value.
8. Develop a risk mitigation strategy and specific risk mitigation activities for the following given risks.
    a) Customer will change requirement.
    b) Lack of training on tools.
9. Illustrate the list of characteristics to assess the quality of the requirements model.
10. Classify the software metrics that provides insights into the quality of the analysis model.
11. Software for System X has 24 individual functional requirements and 14 non-functional requirements. Determine the specificity and completeness of these requirements.
12. What are the four elements that exists when an effective SCM system is implemented? Elaborate each briefly.
13. Briefly describe the difference between SCM for conventional software and SCM for Web or Mobile Apps.
14. Assume that you are the manager of a small project. What baselines would you define for the project and how would you control them?
15. Develop a checklist for use during configuration audit trails.
16. What is the difference between an SCM audit and a technical review? Can their function be folded into one review? What are the pros and cons?
17. Design a project database(repository) system that would enable a software engineer to store, cross-reference, trace, update, and change all important software configuration items.
18. Research an existing SCM tool and describe how it implements control of versions, variants and configuration objects in general.
19. How version controlling is done if 10 % additional requirement changes occur after 3 sprints