

| CST 309 | MANAGEMENT OF SOFTWARE SYSTEMS | Category | L | T | P | Credit | Year of Introduction |
|------------|-----------------------------------|----------|---|---|---|--------|-------------------------|
| | | PCC | 3 | 0 | 0 | 3 | 2019 |

Preamble: This course provides fundamental knowledge in the Software Development Process. It covers Software Development, Quality Assurance, Project Management concepts and technology trends. This course enables the learners to apply state of the art industry practices in Software development.

Prerequisite: Basic understanding of Object Oriented Design and Development.

Course Outcomes: After the completion of the course the student will be able to

| | |
|-----|---|
| CO1 | Demonstrate Traditional and Agile Software Development approaches (Cognitive Knowledge Level: Apply) |
| CO2 | Prepare Software Requirement Specification and Software Design for a given problem. (Cognitive Knowledge Level: Apply) |
| CO3 | Justify the significance of design patterns and licensing terms in software development, prepare testing, maintenance and DevOps strategies for a project. (Cognitive Knowledge Level: Apply) |
| CO4 | Make use of software project management concepts while planning, estimation, scheduling, tracking and change management of a project, with a traditional/agile framework. (Cognitive Knowledge Level: Apply) |
| CO5 | Utilize SQA practices, Process Improvement techniques and Technology advancements in cloud based software models and containers & microservices. (Cognitive Knowledge Level: Apply) |

Mapping of course outcomes with program outcomes

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|
| CO1 | ✓ | ✓ | ✓ | ✓ | | ✓ | | | | | | ✓ |
| CO2 | ✓ | ✓ | ✓ | ✓ | | ✓ | | | | ✓ | ✓ | ✓ |
| CO3 | ✓ | ✓ | ✓ | ✓ | | | | ✓ | | ✓ | ✓ | ✓ |
| CO4 | ✓ | ✓ | ✓ | ✓ | | ✓ | | | ✓ | ✓ | ✓ | ✓ |
| CO5 | ✓ | ✓ | ✓ | ✓ | | ✓ | | | | | | ✓ |

| Abstract POs defined by National Board of Accreditation | | | |
|---|--|------|--------------------------------|
| PO# | Broad PO | PO# | Broad PO |
| PO1 | Engineering Knowledge | PO7 | Environment and Sustainability |
| PO2 | Problem Analysis | PO8 | Ethics |
| PO3 | Design/Development of solutions | PO9 | Individual and team work |
| PO4 | Conduct investigations of complex problems | PO10 | Communication |
| PO5 | Modern tool usage | PO11 | Project Management and Finance |
| PO6 | The Engineer and Society | PO12 | Lifelong learning |

Assessment Pattern

| Bloom's Category | Continuous Assessment Tests | | End Semester Examination Marks |
|-------------------------|------------------------------------|---------------------------|---------------------------------------|
| | Test1 (Percentage) | Test2 (Percentage) | |
| Remember | 30 | 30 | 30 |
| Understand | 40 | 40 | 50 |
| Apply | 30 | 30 | 20 |
| Analyse | | | |
| Evaluate | | | |
| Create | | | |

Mark Distribution

| Total Marks | CIE Marks | ESE Marks | ESE Duration |
|--------------------|------------------|------------------|---------------------|
| 150 | 50 | 100 | 3 hours |

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks (Each student shall identify a software development problem and prepare Requirements Specification, Design Document, Project Plan and Test case documents for the identified problem as the assignment.)

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks.

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing the remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have a maximum of 2 subdivisions and carries 14 marks.

Syllabus

Module 1 : Introduction to Software Engineering (7 hours)

Introduction to Software Engineering - Professional software development, Software engineering ethics. Software process models - The waterfall model, Incremental development. Process activities - Software specification, Software design and implementation, Software validation, Software evolution. Coping with change - Prototyping, Incremental delivery, Boehm's Spiral Model. Agile software development - Agile methods, agile manifesto - values and principles. Agile development techniques, Agile Project Management. Case studies : An insulin pump control system. Mentcare - a patient information system for mental health care.

Module 2 : Requirement Analysis and Design (8 hours)

Functional and non-functional requirements, Requirements engineering processes. Requirements elicitation, Requirements validation, Requirements change, Traceability Matrix. Developing use cases, Software Requirements Specification Template, Personas, Scenarios, User stories, Feature identification. Design concepts - Design within the context of software engineering, Design Process, Design concepts, Design Model. Architectural Design - Software Architecture, Architectural Styles, Architectural considerations, Architectural Design Component level design - What is a component?, Designing Class-Based Components, Conducting Component level design, Component level design for web-apps. Template of a Design Document as per "IEEE Std 1016-2009 IEEE Standard for Information Technology Systems Design Software Design Descriptions". Case study: The Ariane 5 launcher failure.

Module 3 : Implementation and Testing (9 hours)

Object-oriented design using the UML, Design patterns, Implementation issues, Open-source development - Open-source licensing - GPL, LGPL, BSD. Review Techniques - Cost impact of Software Defects, Code review and statistical analysis. Informal Review, Formal Technical Reviews, Post-mortem evaluations. Software testing strategies - Unit Testing, Integration Testing, Validation testing, System testing, Debugging, White box testing, Path testing, Control Structure testing, Black box testing, Testing Documentation and Help facilities. Test automation, Test-driven development, Security testing. Overview of DevOps and Code Management - Code management, DevOps automation, Continuous Integration, Delivery, and Deployment (CI/CD/CD). Software Evolution - Evolution processes, Software maintenance.

Module 4 : Software Project Management (6 hours)

Software Project Management - Risk management, Managing people, Teamwork. Project Planning, Software pricing, Plan-driven development, Project scheduling, Agile planning. Estimation techniques, COCOMO cost modeling. Configuration management, Version management, System building, Change management, Release management, Agile software management - SCRUM framework. Kanban methodology and lean approaches.

Module 5 : Software Quality, Process Improvement and Technology trends (6 hours)

Software Quality, Software Quality Dilemma, Achieving Software Quality Elements of Software Quality Assurance, SQA Tasks, Software measurement and metrics. Software Process Improvement(SPI), SPI Process CMMI process improvement framework, ISO 9001:2000 for Software. Cloud-based Software - Virtualisation and containers, Everything as a service(IaaS, PaaS), Software as a service. Microservices Architecture - Microservices, Microservices architecture, Microservice deployment.

Text Books

1. Book 1 - Ian Sommerville, Software Engineering, Pearson Education, Tenth edition, 2015.
2. Book 2 - Roger S. Pressman, Software Engineering : A practitioner's approach, McGraw Hill publication, Eighth edition, 2014
3. Book 3 - Ian Sommerville, Engineering Software Products: An Introduction to Modern Software Engineering, Pearson Education, First Edition, 2020.

References

1. IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications
2. IEEE Std 1016-2009 IEEE Standard for Information Technology—Systems Design—Software Design Descriptions

3. David J. Anderson, Kanban, Blue Hole Press 2010
4. David J. Anderson, Agile Management for Software Engineering, Pearson, 2003
5. Walker Royce, Software Project Management : A unified framework, Pearson Education, 1998
6. Steve. Denning, The age of agile, how smart companies are transforming the way work gets done. New York, Amacom, 2018.
7. Satya Nadella, Hit Refresh: The Quest to Rediscover Microsoft's Soul and Imagine a Better Future for Everyone, Harper Business, 2017
8. Henrico Dolfing, Project Failure Case Studies: Lessons learned from other people's mistakes, Kindle edition
9. Mary Poppendieck, Implementing Lean Software Development: From Concept to Cash, Addison-Wesley Signature Series, 2006
10. StarUML documentation - <https://docs.staruml.io/>
11. OpenProject documentation - <https://docs.openproject.org/>
12. BugZilla documentation - <https://www.bugzilla.org/docs/>
13. GitHub documentation - <https://guides.github.com/>
14. Jira documentation - <https://www.atlassian.com/software/jira>

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What are the advantages of an incremental development model over a waterfall model?
2. Illustrate how the process differs in agile software development and traditional software development with a socially relevant case study. (Assignment question)

Course Outcome 2 (CO2):

1. How to prepare a software requirement specification?
2. Differentiate between Architectural design and Component level design.
3. How does agile approaches help software developers to capture and define the user requirements effectively?
4. What is the relevance of the SRS specification in software development?
5. Prepare a use case diagram for a library management system.

Course Outcome 3 (CO3):

1. Differentiate between the different types of software testing strategies.
2. Justify the need for DevOps practices?
3. How do design patterns help software architects communicate the design of a complex system effectively?

4. What are the proactive approaches one can take to optimise efforts in the testing phase?

Course Outcome 4 (CO4):

1. Illustrate the activities involved in software project management for a socially relevant problem?
2. How do SCRUM, Kanban and Lean methodologies help software project management?
3. Is rolling level planning in software project management beneficial? Justify your answer.
4. How would you assess the risks in your software development project? Explain how you can manage identified risks?

Course Outcome 5 (CO5):

1. Justify the importance of Software Process improvement?
2. Explain the benefits of cloud based software development, containers and microservices.
3. Give the role of retrospectives in improving the software development process.
4. Illustrate the use of project history data as a prediction tool to plan future socially relevant projects.

Model Question Paper

QP CODE:

Reg No: _____

Name : _____

PAGES : 3

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
FIFTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR**

Course Code: CST 309

Course Name: Management of Software Systems

Duration: 3 Hrs

Max. Marks :100

PART A

Answer all Questions. Each question carries 3 marks

1. Why professional software that is developed for a customer is not simply the programs that have been developed and delivered.
2. Incremental software development could be very effectively used for customers who do not have a clear idea about the systems needed for their operations. Justify.
3. Identify any four types of requirements that may be defined for a software system
4. Describe software architecture
5. Differentiate between GPL and LGPL?
6. Compare white box testing and black box testing.
7. Specify the importance of risk management in software project management?
8. Describe COCOMO cost estimation model.
9. Discuss the software quality dilemma
10. List the levels of the CMMI model? (10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Compare waterfall model and spiral model

(8)

(b) Explain Agile ceremonies and Agile manifesto (6)

12. (a) Illustrate software process activities with an example. (8)

(b) Explain Agile Development techniques and Agile Project Management (6)

13. (a) What are functional and nonfunctional requirements? Imagine that you are developing a library management software for your college, list eight functional requirements and four nonfunctional requirements. (10)

(b) List the components of a software requirement specification? (4)

OR

14. (a) Explain Personas, Scenarios, User stories and Feature identification? (8)

(b) Compare Software Architecture design and Component level design (6)

15. (a) Explain software testing strategies. (8)

(b) Describe the formal and informal review techniques. (6)

OR

16. (a) Explain Continuous Integration, Delivery, and Deployment CI/CD/CD) (8)

(b) Explain test driven development (6)

17. (a) What is a critical path and demonstrate its significance in a project schedule with the help of a sample project schedule. (8)

(b) Explain plan driven development and project scheduling. (6)

OR

18. (a) Explain elements of Software Quality Assurance and SQA Tasks. (6)

(b) What is algorithmic cost modeling? What problems does it suffer from when (8)

compared with other approaches to cost estimation?

19. (a) Explain elements of Software Quality Assurance and SQA Tasks. (8)

(b) Illustrate SPI process with an example. (6)

OR

20. (a) Compare CMMI and ISO 9001:2000. (8)

(b) How can Software projects benefit from Container deployment and Micro service deployment? (6)

Teaching Plan

| No | Contents | No of Lecture Hrs |
|--|---|-------------------|
| Module 1 : Introduction to Software Engineering (7 hours) | | |
| 1.1 | Introduction to Software Engineering.[Book 1, Chapter 1] | 1 hour |
| 1.2 | Software process models [Book 1 - Chapter 2] | 1 hour |
| 1.3 | Process activities [Book 1 - Chapter 2] | 1 hour |
| 1.4 | Coping with change [Book 1 - Chapter 2, Book 2 - Chapter 4] | 1 hour |
| 1.5 | Case studies : An insulin pump control system. Mentcare - a patient information system for mental health care. [Book 1 - Chapter 1] | 1 hour |
| 1.6 | Agile software development [Book 1 - Chapter 3] | 1 hour |
| 1.7 | Agile development techniques, Agile Project Management.[Book 1 - Chapter 3] | 1 hour |
| Module 2 : Requirement Analysis and Design (8 hours) | | |
| 2.1 | Functional and non-functional requirements, Requirements engineering processes [Book 1 - Chapter 4] | 1 hour |
| 2.2 | Requirements elicitation, Requirements validation, Requirements change, Traceability Matrix [Book 1 - Chapter 4] | 1 hour |
| 2.3 | Developing use cases, Software Requirements Specification Template [Book 2 - Chapter 8] | 1 hour |

| | | |
|---|--|--------|
| 2.4 | Personas, Scenarios, User stories, Feature identification [Book 3 - Chapter 3] | 1 hour |
| 2.5 | Design concepts [Book 2 - Chapter 12] | 1 hour |
| 2.6 | Architectural Design [Book 2 - Chapter 13] | 1 hour |
| 2.7 | Component level design [Book 2 - Chapter 14] | 1 hour |
| 2.8 | Design Document Template. Case study: The Ariane 5 launcher failure. [Ref - 2, Book 2 - Chapter 16] | 1 hour |
| Module 3 : Implementation and Testing (9 hours) | | |
| 3.1 | Object-oriented design using the UML, Design patterns [Book 1 - Chapter 7] | 1 hour |
| 3.2 | Implementation issues, Open-source development - Open-source licensing - GPL, LGPL, BSD [Book 1 - Chapter 7] | 1 hour |
| 3.3 | Review Techniques - Cost impact of Software Defects, Code review and statistical analysis. [Book 2 - Chapter 20] | 1 hour |
| 3.4 | Informal Review, Formal Technical Reviews, Post-mortem evaluations. [Book 2 - Chapter 20] | 1 hour |
| 3.5 | Software testing strategies - Unit Testing, Integration Testing, Validation testing, System testing and Debugging (basic concepts only). [Book 2 - Chapter 22] | 1 hour |
| 3.6 | White box testing, Path testing, Control Structure testing, Black box testing. Test documentation [Book 2 - Chapter 23] | 1 hour |
| 3.7 | Test automation, Test-driven development, Security testing. [Book 3 - Chapter 9] | 1 hour |
| 3.8 | DevOps and Code Management - Code management, DevOps automation, CI/CD/CD. [Book 3 - Chapter 10] | 1 hour |
| 3.9 | Software Evolution - Evolution processes, Software maintenance. [Book 1 - Chapter 9] | 1 hour |
| Module 4 : Software Project Management (6 hours) | | |
| 4.1 | Software Project Management - Risk management, Managing people, Teamwork [Book 1 - Chapter 22] | 1 hour |
| 4.2 | Project Planning - Software pricing, Plan-driven development, Project scheduling, Agile planning [Book 1 - Chapter 23] | 1 hour |
| 4.3 | Estimation techniques [Book 1 - Chapter 23] | 1 hour |
| 4.4 | Configuration management [Book 1 - Chapter 25] | 1 hour |

| | | |
|---|--|--------|
| 4.5 | Agile software management - SCRUM framework [Book 2 - Chapter 5] | 1 hour |
| 4.6 | Kanban methodology and lean approaches.[Ref 9 - Chapter 2] | 1 hour |
| Module 5 : Software Quality, Process Improvement and Technology trends (6 hours) | | |
| 5.1 | Software Quality, Software Quality Dilemma, Achieving Software Quality. [Book 2 - Chapter 19] | 1 hour |
| 5.2 | Elements of Software Quality Assurance, SQA Tasks , Software measurement and metrics. [Book 3 - Chapter 21] | 1 hour |
| 5.3 | Software Process Improvement (SPI), SPI Process [Book 2 - Chapter 37] | 1 hour |
| 5.4 | CMMI process improvement framework, ISO 9001:2000 for Software. [Book 2 - Chapter 37] | 1 hour |
| 5.5 | Cloud-based Software - Virtualisation and containers, IaaS, PaaS, SaaS.[Book 3 - Chapter 5] | 1 hour |
| 5.6 | Microservices Architecture - Microservices, Microservices architecture, Microservice deployment [Book 3 - Chapter 6] | 1 hour |