

COURSE TITLE	SOFTWARE ENGINEERING
COURSE CODE	09CE3402
COURSE CREDITS	3

Objective:

- 1 To understand and apply various project management techniques based on Software Engineering guidelines and Principles.

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

- 1 Apply appropriate software process models and requirement engineering techniques to develop structured solutions for real-world software problems
- 2 Apply project planning, scheduling, and tracking techniques to manage software development activities effectively
- 3 Apply risk identification, analysis, and mitigation strategies to manage uncertainties in software projects
- 4 Apply UML-based design and modeling techniques to represent system structure and behavior
- 5 Apply coding standards, testing techniques, and quality assurance practices to ensure reliable and maintainable software systems

Pre-requisite of course: Object Oriented Programming Fundamentals

Teaching and Examination Scheme

Theory Hours	Tutorial Hours	Practical Hours	ESE	IA	CSE	Viva	Term Work
2	0	2	50	30	20	25	25

Contents : Unit	Topics	Contact Hours
1	Introduction to Software Engineering Software engineering, Dual role of software, Various Myths Associated with Software, Different Software Process Models, The Linear Sequential Model, The Prototyping Model, The RAD Model, Evolutionary Process Models, Component-Based Development	7
2	Project Management Concepts, Requirement Engineering & Metrics: The Management Spectrum 4P's (People, Project, Product and Process), The W5HH Principle, Basic concept of Requirement (Functional & Non Functional), Metrics in the Process and Project Domains, Software Measurement	4
3	Project Planning Scheduling & Tracking Software Scope, Feasibility Analysis, Empirical Estimation Models, Gantt Charts and Tools used	4

Contents : Unit	Topics	Contact Hours
4	Risk Analysis And Management Reactive versus Proactive Risk Strategies, Risk Management Process, Risk Identification, Risk Projection, Risk Refinement, RMMM Plans	4
5	Design and Modelling using UML Outcome of Design Process, Effective Modular Design(Functional & Non-Functional), Cohesion, Coupling	4
6	Software Coding & Testing Coding standards & Coding Guidelines, Code Review, Software Documentation: Internal and External Documentation, Software Testing Fundamentals, Software Testing Techniques, White Box Testing Techniques, Black Box Testing Techniques	5
Total Hours		28

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
1	Practical-1 Identify Project (System), Identify basic requirements and users. Based on the information Write Introduction, Abstract, Scope and prepare Use Case Description Document	2
2	Practical-2 Use Case Diagram: Design a UML Use Case Diagram to represent the functional requirements of the selected system. Identify actors, use cases, and relationships such as include, extend, and generalization	2
3	Practical-3 Activity Diagram: Develop an Activity Diagram to model the workflow of the overall system. Represent decision points, parallel activities, and process flow logically.	2
4	Practical-4 Flowchart: Create a system-level and module-level flowchart showing procedural logic. Use standard flowchart symbols to represent process, input/output, and control flow	2
5	Practical-5 Sequence Diagram: Design a Sequence Diagram to illustrate interaction between objects over time. Show lifelines, activation bars, and message passing for a core transaction.	2
6	Practical-6 Context-Level DFD (Level 0): Prepare a Context Diagram representing the system as a single process. Identify external entities and major data flows entering and leaving the system.	2
7	Practical-7 DFD Level 1: Develop a Level 1 Data Flow Diagram by decomposing the main system process. Ensure proper balancing of data flows with the Context Diagram.	2

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
8	Practical-8 DFD Level 2: Create a Level 2 DFD for one major subsystem of the project. Provide detailed decomposition while maintaining logical data consistency.	2
9	Practical-9 Entity Relationship (ER) Diagram: Design an ER Diagram to model database entities, attributes, and relationships. Specify primary keys, foreign keys, and cardinality constraints.	4
10	Practical-10 Class Diagram: Develop a UML Class Diagram representing system structure in object-oriented form. Define classes, attributes, methods, and relationships such as inheritance and aggregation.	2
11	Practical-11 Mini Software Requirement Specification (Mini SRS): Prepare a concise SRS document following IEEE standards for the selected project. Integrate all diagrams and clearly define functional and non-functional requirements.	6
Total Hours		28

Textbook :

- 1 Fundamentals of software Engineering, 5th Edition, Rajib Mall, Prentice Hall of India, 2018
- 2 Software engineering- A practitioner's Approach, 9th Edition, Roger S.Pressman, McGraw-Hill International Editions, 2023
- 3 Software Engineering, 10th Edition, Ian Sommerville, Pearson education Asia, 2017

References:

- 1 Engineering Software as a Service and Agile Software Approach, Engineering Software as a Service and Agile Software Approach, Armando Fox and David Patterson, O'Reilly Media, 2017
- 2 Project Management for Business, Engineering and Technology, Project Management for Business, Engineering and Technology, John M. Nicholas, Elsevier, 2012

Suggested Theory Distribution:

The suggested theory distribution as per Bloom's taxonomy is as follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process

Distribution of Theory for course delivery and evaluation					
Remember / Knowledge	Understand	Apply	Analyze	Evaluate	Higher order Thinking / Creative
0.00	0.00	35.00	35.00	30.00	

Instructional Method:

- 1 The course delivery method will depend upon the requirement of content and need of students. The teacher in may be using following teaching approaches: black board, or use of any of tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.
- 2 The internal evaluation will be done on the basis of continuous evaluation of students in the laboratory and class-room.
- 3 Practical examinations/Viva will be conducted at the end of semester for evaluation of performance of students in laboratory.
- 4 Students will use supplementary resources such as online videos, NPTEL videos, e courses, Virtual Laboratory.

Supplementary Resources:

- 1 <http://nptel.ac.in/courses/106101061/>
- 2 <https://www.joelonsoftware.com/>
- 3 <http://www.codesimplicity.com/>
- 4 <http://www.sparxsystems.com/products/ea/index.html>
- 5 <http://www.smartdraw.com/>
- 6 <http://www.win.tue.nl/>