Course Name: Software Engineering Course Code: ITEC322

General Information

Course Code	**ITEC322**	**Level/Year**	**5/3rd**	**Required (R) /Selected Elective (SE)**		**R**
Credit Hours	**Theory**	**3**	**Lab**	**1**	**Total**	**3**
Prerequisites	**Nil**	**Course Coordinator**		**Dr. R. SIVA MALAR**		
Corequisites	**Nil**	**Track Leader**		**Dr. R. SIVA MALAR**		

Course Description

Software engineering is a major branch of computing science that deals with the development of software systems as practical and cost-effective solutions for individuals and society. This course covers the fundamentals of software engineering like software life cycle, requirements engineering, system development paradigm, and system modeling using UML. It also covers software verification & validation, important implementation issues, open-source development, and concepts of software re-engineering. The course has a strong technical relation with graduation project providing the opportunity to practice software engineering knowledge, skills, and practices in a realistic development setting with a real client.

Course Objectives : On completion of the course, the student will be able to:

This course will develop the students' ability to learn:

- What is software development life cycle (SDLC)?
- How to elicit requirements from a client and their classification?
- How to use graphical models (UML diagrams) to represent software architecture?
- What are the stages of software testing, and its role in V&V?
- What rules have to follow for Re-Engineering?

Course Contents

List of Topics	**Weeks**
CH 1: Introduction to Software Engineering :Basic Definitions, Software costs, Role of Management in Software Development, Software Products, Essential attributes of good software, Challenges for Software Engineering Practices, Software Engineering diversity, Application types, Deliverables and Milestones, Software engineering fundamentals, Software engineering ethics.	1,2
CH 2: Software Processes: Introduction, The Software process, Software Process Models, Waterfall Model, Incremental Development Process Model, Reuse-oriented Model, Boehm's Spiral Model, Agile methods, Comparison of Different Software Process Models.	3, 4, 5
CH 3: Requirement Engineering: Requirement Engineering, Crucial process steps of Requirement Engineering, Types of Requirements, User and System Requirement, Categories of Metric, SRS Document, Requirements Gathering Techniques	6,7
CH 4: System Modeling and Architectural Design: System Modeling, UML Diagram Types, Graphical Models, Context Model, Interaction Models , Structural Models, Architectural Design, Use of Architectural Models, Architectural Views, Application Architecture, Transaction Processing system.	8, 9
CH 5: Design and Implementation: Design and Implementation, An Object-Oriented Design Process, Object-Oriented Design Process Stages, Context And Interaction Model, Implementation Issues, Reuse, Configuration Management,	10, 11

Host-Target Development, Open Source Development, Open Source Systems, Open Source Business, Open Source Licensing.	
CH6: Software Testing : Software Verification and Validation, Software Testing, Why should we test?, Who should do the testing?, What should we test?, Stages Of Testing, Development Testing, Release Testing, User Testing, Black Box Testing, Equivalence Partitioning, Boundary Value Analysis, White Box Testing, Methods used in WBT.	12, 13
CH7: Software Evolution and Maintenance: Software Evolution, Software Maintenance, Problems during Maintenance, Types of Maintenance, The Software Maintenance Process, Maintenance Cost, Maintenance Cost Factor, Maintenance prediction, Software Re-Engineering, Reengineering process activities, Advantages of Re-Engineering, Reengineering cost factors.	14, 15

Textbook

- Software Engineering, 10th Edition, 2021, Ian Sommerville, Pearson Education. ISBN: 9780137503148

Reference Materials

- R. S. Pressman, Software Engineering: A Practitioners Approach, 9th edition, 2020, McGraw Hill International publication.

Course Learning Outcomes

CLO	Description	Mapped PI
CLO#01	**Recognize** basic software development and computing fundamentals that make up the Software Development Lifecycle.	PI 1.1
CLO#02	**Identify** effective software engineering process based on requirements classification.	PI 1.3
CLO#03	**Construct** computer-based systems to meet desired needs by producing software artifacts in UML.	PI 2.1
CLO#04	**Analyze** design procedures and implementation issues of the software product.	PI 2.2
CLO#05	**Demonstrate** and communicate software engineering principles effectively through verbal and/or written presentations	PI 3.2 PI 3.1
CLO#06	**Critique** about the software quality using V&V technique, cost for maintenance and re-engineering of software product.	PI 2.4

CLO-PI-SOMapping

	SO-1	SO-2	SO-3	SO-4	SO-5	SO-6
CLO#01	PI 1.1	-	-	-	-	-
CLO#02	PI 1.3	-	-	-	-	-
CLO#03	-	PI 2.1	-	-	-	-
CLO#04	-	PI 2.2	-	-	-	-
CLO#05	-	-	PI 3.1 PI 3.2	-	-	-
CLO#06	-	PI 2.4	-	-	-	-