	Course Title: Software Engineering		
	Course Code: **18CS51**	**No. of Credits: 3 : 0 : 0** **(L-T-P)**	**No. of lecture hours/week : 3**
	Exam Duration : 3 hours	**CIE+ Assignment + SEE = 45+5+50=100**	**Total No. of Contact Hours : 42**

Course Objectives:	**Description**
	1. To provide the idea of decomposing the given problem into Analysis, Design, Implementation, Testing and Maintenance phases. 2. To provide an idea of using various process models in the software industry according to given circumstances. 3. To gain the knowledge of how Analysis, Design, Implementation, Testing and Maintenance processes are conducted in a software project.

Unit No	**Syllabus Content**	**No of Hours**
1	**SOFTWARE AND SOFTWARE ENGINEERING:** The Nature of Software, Software Engineering, The Software Process, Software Engineering Practice. **THE SOFTWARE PROCESS and PROCESS MODELS:** A Generic Process Model, Process Assessment and Improvement, Prescriptive Process Models: The Waterfall Model, Incremental Process Models, Evolutionary Process Models, Concurrent Models, Final Word on Evolutionary Processes, Specialized Process Models: Component-Based Development, The Formal Methods Model, The Unified Process, Phases of the Unified Process, Personal and Team Process Models. **AGILE DEVELOPMENT:** What Is Agility? Agility and the Cost of Change, What Is an Agile Process? , Extreme Programming, Other Agile Process Models: Scrum, Dynamic Systems Development Method, Agile Modeling, Agile Unified Process.	**10**
2	**UNDERSTANDING REQUIREMENTS:** Definition of Requirements Engineering, Establishing the Groundwork, Eliciting Requirements, Developing Use Cases, Building the Requirements Model, Negotiating Requirements and Validating Requirements. **REQUIREMENTS MODELING: SCENARIO-BASED METHODS:** Requirements Analysis, Scenario-Based Modeling, UML Models That Supplement the Use Case.	**8**
3	**DESIGN CONCEPTS:** Design within the Context of Software Engineering, The Design Process, Design Concepts, The Design Model. **ARCHITECTURAL DESIGN:** Software Architecture, Definition of software architecture, Architectural Genres, Architectural Styles, Architectural Design. **COMPONENT-LEVEL DESIGN:** What Is a Component? Designing Class-Based Components, Conducting Component-Level Design, Designing Traditional Components and Component-Based Development.	**8**

4	**SOFTWARE TESTING STRATEGIES:** A Strategic Approach to Software Testing, Strategic Issues, Test Strategies for Conventional Software, Validation Testing, System Testing, The Art of Debugging. **TESTING CONVENTIONAL APPLICATIONS:** Software Testing Fundamentals, Internal and External Views of Testing, White-Box Testing, Basis Path Testing, Control Structure Testing, Black-Box Testing.	**8**
5	**SELF-STUDY –** **PROJECT MANAGEMENT CONCEPTS:** The management spectrum, People, Product, Process, Project, W^5HH principle. **PROCESS AND PROJECT METRICS:** Metrics in the process and project domains, Software measurement, metrics for Software quality, Integrating metrics within the software process, Metrics for small organizations, Establishing a software metrics program. **ESTIMATION FOR SOFTWARE PROJECTS:** Observations on estimation, The project planning process, Software scope and feasibility, Resources, Software project estimation, Decomposition techniques, Empirical estimation models.	**8**

Course Outcomes	**Description**	**RBT Levels**
CO1	Decompose the given project in various phases of a lifecycle.	Knowledge, Understand (Level1, Level2)
CO2	Choose appropriate process model depending on the user requirements.	Apply, Create (Level 2)
CO3	Perform various life cycle activities like Analysis, Design, Implementation, Testing and Maintenance.	Evaluate(Level 3)
CO4	Analyze various processes used in all the phases of the product.	Analyze(Level 3)
CO5	Apply the knowledge, techniques, and skills in the development of a software product.	Apply (Level 3)

CO-PO Mapping	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	**3**	**2**	**2**									
CO2	**3**	**2**	**1**									
CO3	**2**	**2**	**1**		**3**			**1**				
CO4	**2**	**2**		**2**		**1**		**1**			**2**	**2**
CO5	**1**	**2**										**2**

Strong -3 Medium -2 Weak -1

TEXT BOOKS:

1. **Software Engineering - A Practitioner’s approach,** Roger S. Pressman and Bruce R. Maxim, 8th Edition, Tata McGraw-Hill, 2019.

REFERENCE BOOKS:

1. **Software Engineering,** 10th Edition, Ian Sommerville, Pearson Education Ltd., 2017.

2. **Software Engineering - A Precise Approach,** Pankaj Jalote, Wiley, 2010.

SELF STUDY REFERENCES/WEBLINKS:

1. http://www.site.uottawa.ca/school/research/lloseng/weblinks.html

2. https://www.ece.rutgers.edu/~marsic/books/SE/links/

COURSE COORDINATOR:	**Praveena M V**