

**Subject Code: 09CT0406**  
**Subject Name: Software Engineering**  
**Diploma Year – II ( Semester IV)**

**Objective:** The rationale behind this course is to understand steps of building a software.

**Credits Earned:** 3 Credits

**Course Outcomes:** After learning this course, students should be able to,

1. Acquire knowledge about principles of software engineering, software project, process, SDLC, Risk Analysis.
2. Understand the Basic Knowledge about various Software Model, Design process, Software Requirement specification.
3. Understand Project management, object oriented Modeling techniques, Testing strategies.
4. Apply design and development principles in the construction of software systems of varying complexity.
5. Analyse software engineering model and testing tool to prepare software model for appropriate project.

**Pre-requisite of course:** Basic operating system concepts.

**Teaching and Examination Scheme**

Teaching Scheme (Hours)			Credits	Theory Marks			Tutorial/ Practical Marks		Total Marks
Theory	Tutorial	Practical		ESE (E)	IA (M)	CSE (I)	Viva (V)	Term work (TW)	
3	0	0	3	50	30	20	0	0	100

**Contents:**

Sr No	Course content	Total Hrs.
<b>1</b>	<b>Introduction to Software development</b> What is Software Engineering, Problem Identification, Program versus Project, Software engineering applications	<b>04</b>
<b>2</b>	<b>Software Lifecycle models</b> Introduction to life cycle model, Documentation of model, Types of models: Waterfall Model, Iterative Waterfall model, Prototyping Model, Spiral Model.	<b>07</b>



3	<b>Requirements Gathering and Analysis</b> Introduction of Requirements Gathering and Analysis, Software Requirement specification (SRS): Characteristics and categories, Functional Requirements, organization and techniques.	06
4	<b>Software Design and Project Management</b> Design process, Characteristics of Good Software Design, Cohesion and Coupling. Functional Design: Structured Analysis, Data Flow Diagrams, Sequence Diagrams Project Planning, Size Estimation metrics of Project, Project Estimation techniques, COCOMO model: Estimation technique, Scheduling	07
5	<b>Object oriented Modelling</b> Basic concepts and mechanisms, UML: Use case model, Class Diagrams, Activity Diagrams, Interaction Diagrams, State Diagrams	06
6	<b>Risk and Quality Management</b> What is Risk in Software development, Reactive vs Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, Quality Management: Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews	08
7	<b>Testing Strategies</b> Concept of Testing, Types of Testing: Black Box Testing, White Box testing, Unit Testing	04
	<b>Total</b>	<b>42 hrs.</b>

**References:**

1. Rajib Mall, Fundamentals of software Engineering, Prentice Hall of India
2. Roger S.Pressman, Software engineering-A practitioner's Approach, McGraw-Hill International Editions
3. Ian Sommerville, Software engineering, Pearson education Asia
4. Pankaj Jalote, Software Engineering –A Precise Approach Wiley
5. Software Engineering Fundamentals by Ali Behhforoz& Frederick Hudson OXFORD

**Suggested Theory distribution:**

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

R Level	U Level	A Level	N Level	E Level	C Level
40	40	10	10	0	0

**Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create and above Levels (Revised Bloom's Taxonomy)**



Distribution of Theory for course delivery and evaluation					
Remember	Understand	Apply	Analyse	Evaluate	Create
40%	40%	10%	10%	0%	0%

**Instructions:**

- a) Students can refer any reference material for tutorials/assignments.
- b) Assignments will be evaluated based on quiz conducted at the end of every Unit.

**Case Study evaluations:**

Students have to work on given case studies for following tasks to understand practical implementation of lectures outcome.

1. Problem Statement Identification
2. Problem Statement Description and Design
3. Requirements Gathering for problem Solution
4. Proposed Solution of Problem
5. Data Flow Modelling
6. UML diagrams introduction
7. Use Case Modelling
8. Activity Diagram
9. Class Diagram
10. Sequence diagram

**Example of Case Studies**

- Library Management System
- Railway Transportations management system
- E-Ticket Booring
- Shopping Cart

**Reference Materials:**

- <https://www.startertutorials.com/uml/introduction-rational-rose.html>
- <http://nptel.ac.in/courses/106101061/2>
- <https://www.joelonsoftware.com/3.http://www.codesimplicity.com/4.http://www.sparxsystems.com/products/ea/index.html5>
- <http://www.smartdraw.com6>
- [http://viu.eng.rpi.edu7.www.en.wikipedia.org/wiki/Software\\_engineering](http://viu.eng.rpi.edu7.www.en.wikipedia.org/wiki/Software_engineering)