

Software Engineering	L	P	C
	3		3

Discipline(s) / EAE / OAE	Semester	Group	Sub-group	Paper Code
CSE/IT/CST/ITE	5	PC	PC	CIC-309

Marking Scheme:												
1. Teachers Continuous Evaluation: 25 marks												
2. Term end Theory Examinations: 75 marks												
Instructions for paper setter:												
1. There should be 9 questions in the term end examinations question paper.												
2. The first (1st) question should be compulsory and cover the entire syllabus. This question should be objective, single line answers or short answer type question of total 15 marks.												
3. Apart from question 1 which is compulsory, rest of the paper shall consist of 4 units as per the syllabus. Every unit shall have two questions covering the corresponding unit of the syllabus. However, the student shall be asked to attempt only one of the two questions in the unit. Individual questions may contain upto 5 sub-parts / sub-questions. Each Unit shall have a marks weightage of 15.												
4. The questions are to be framed keeping in view the learning outcomes of the course / paper. The standard / level of the questions to be asked should be at the level of the prescribed textbook.												
5. The requirement of (scientific) calculators / log-tables / data – tables may be specified if required.												
Course Objectives :												
1.	To introduce the basic concepts of the software development processes, Software requirements and specifications											
2.	To impart knowledge of Software Project Planning and various Software design techniques for developing large software systems.											
3.	To understand Software Metrics, Software Reliability, and Quality assurance using ISO 9001 and SEI-CMM.											
4.	To impart the knowledge and use of software engineering processes and tools in analysis, design, implementation, software testing, documentation, and maintenance for software systems.											
Course Outcomes (CO)												
CO 1	Ability to have an understanding of SDLC Models, Techniques for Requirement Elicitation, and SRS Document.											
CO 2	To be able to explain Software Project Planning and various methods for software design											
CO 3	To Understand Software Metrics, Software Reliability, and Quality assurance											
CO 4	Ability to have an understanding of Software testing, documentation and maintenance.											
Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)												
	PO01	PO02	PO03	PO04	PO05	PO06	PO07	PO08	PO09	PO10	PO11	PO12
CO 1	3	2	2	2	3	-	-	-	3	2	2	3
CO 2	3	2	2	2	3	-	-	-	3	2	2	3
CO 3	3	2	2	2	3	-	-	-	3	2	2	3
CO 4	3	2	2	2	3	-	-	-	3	2	2	3
UNIT-I												
Introduction: Introduction to Software Engineering, Importance of software engineering as a discipline, Software applications, Software Crisis, Software Processes & Characteristics, Software life cycle models, Waterfall, Prototype, Evolutionary and Spiral Models.												
Software Requirements Analysis & Specifications: Requirement engineering, Functional and non-functional requirements, User requirements, System requirements, requirement elicitation techniques like FAST, QFD & Use case approach, requirements analysis using DFD, Data dictionaries & ER Diagrams, Requirements documentation, Nature of SRS, Characteristics & organization of SRS, Requirement Management, IEEE Std. for SRS.												

UNIT-II

Software Project Planning: Size Estimation like lines of Code & Function Count, Cost Estimation Models, COCOMO, Putnam resource allocation model, Validating Software Estimates, Risk Management.

Software Design: Cohesion & Coupling, Classification of Cohesiveness & Coupling, Function Oriented Design, Object Oriented Design, User Interface Design.

UNIT-III

Software Metrics: Software measurements: What & Why, Token Count, Halstead Software Science Measures, Data Structure Metrics, Information Flow Metrics.

Software Reliability: Importance, Hardware Reliability & Software Reliability, Failure and Faults, Reliability Models- Basic Model, Logarithmic Poisson Model, Software Quality Models, CMM & ISO 9001.

UNIT – IV

Software Testing: Testing process, Functional testing: Boundary value analysis, Equivalence class testing, Decision table testing, Cause effect graphing, Structural testing: Path testing, Data flow and mutation testing, unit testing, integration and system testing, Debugging, Testing Tools & Standards.

Software Maintenance: Management of Maintenance, Maintenance Process, Maintenance Models, Regression Testing, Reverse Engineering, Software Re-engineering, Configuration Management, Documentation.

Textbook(s):

1. K. K. Aggarwal and Yogesh Singh, "Software Engineering", New Age International, 3rd Ed., 2005.
2. R. S. Pressman, "Software Engineering – A Practitioner's Approach", McGraw Hill Int. , 5th Ed., 2001.
3. Pankaj Jalote, "An Integrated Approach to Software Engineering", Narosa, 3rd Ed., 2005.

References:

1. Stephen R. Schach, "Classical & Object Oriented Software Engineering", IRWIN, 1996.
2. James Peter, W. Pedrycz, "Software Engineering: An Engineering Approach", John Wiley & Sons.
3. I. Sommerville, "Software Engineering", Addison Wesley, 8th Ed., 2009.
4. Frank Tsui and Orlando Karan, "Essentials of Software Engineering", Joes and Bartlett, 2nd Ed., 2010.
5. Kassem A. Saleh, "Software Engineering", Cengage Learning, 2009.
6. Rajib Mall, "Fundamental of Software Engineering", PHI, 3rd Ed., 2009.
7. Carlo Ghizzi, Mehdi Jazayeri and Dino Mandrioli, "Fundamental of Software Engineering", PHI, 2nd Ed., 2003.
8. Carol L. Hoover, Mel Rosso-Llopert and Gil Taran, "Evaluating Project Decision Case Studies in Software Engineering", Pearson, 2010.