3. SYLLABUS

SOFTWAREENGINEERING

B.Tech IIYear IISem

Course Code	Category	Hours/ Week			Credits	Maximum Marks		
		L	T	P		CIE	SEE	TOTAL
23CS405	Professional Core	3	0	0	3	40	60	100
Contact Classes:48	Tutorial Classes: Nil	Practical Classes:-				TotalClasses:48		

Course Objectives

1. The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects.
2. Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams
3. To elaborate techniques and processes for software requirements, design methodologies, coding and testing methodologies, softwaremetrics and quality.
4. To make the students understand how the applications of softwareengineering principles.
5. To make the students understand the quality of software and decrease the cost of software development and maintenance

Course Outcomes

1. Ability to translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document (SRD).
2. Identify and apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.
3. Will have experience and /or awareness of testing problems and will be able to develop a simple testing report
4. Develop a strategic approach to testing and use debugging Techniques.
5. Analyze the importance of software metrics and apply risk managementstrategies.

UNIT-I

Introduction to Software Engineering : The evolving role of software, changing nature of software, Software myths.

A Generic view of process: Software engineering-a layered technology, a process

your roots to success.

Frame work, the capability maturity model integration (CMMI).
Process models: The waterfall model, Spiral model and agile methodology

UNIT–II
Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document.
Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

UNIT–III
Design Engineering: Design process and design quality, design concepts, the design model. Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

UNIT-IV
Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging. Metrics for Process and Products: Software measurement, metrics for software quality.

UNIT–V
Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM.
Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.

TEXTBOOKS:

1. Software Engineering, Apractitioner'sApproach-RogerS.Pressman,6th edition, McGraw Hill International Edition.
2. Software Engineering-Somerville, 7th edition, Pearson Education.

REFERENCEBOOKS:

1. The unified modeling language user guide Grady Botch, James Rambaugh, Ivar Jacobson, and Pearson Education.
2. Software Engineering, an Engineering approach-James F.Peters, Wit old Pedrycz, John Wiley.
3. Software Engineering principles and practice-WamanSJawadekar, The McGraw-Hill Companies.
4. Fundamentals of object-oriented design using UMLMeilerpage-Jones: Pearson Education.

your roots to success.