School of Computer Science

1:SYLLABUS (NR21)

B.Tech. III Year I Semester								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maxumum Marks**		
CS3102PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**0**	**0**	**3**	**30**	**70**	**100**
Contact classes: 60	**Tutorial Classes :**	**Practical classes : NIL**				**Total Classes :60**		
Prerequisite: No Prerequisites								

Course Objectives:

1. To help the student differentiate between the programming approach and the software engineering approach and introduce the issues while building large programs.

2. To introduce basic concepts of software engineering through – project, product, process models, personal software process, team software process, umbrella activities.

3. To elaborate techniques and processes for software requirements, design methodologies, coding and testing methodologies, software metrics and quality.

4. To make the students understand how the applications of software engineering principles.

5. To make the students understand the quality of software and decrease the cost of software development and maintenance

Course Outcomes:

1. Understand underlying principles of software engineering, software myths and Software Process models.

2. Understand requirements engineering process and related system models.

3. Understand software design process, design quality, design models and create architectural designs, component designs and UI designs.

4. Develop a strategic approach to testing and use debugging Techniques.

5. Analyze the importance of software metrics and apply risk management strategies.

COURSE SYLLABUS

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 frame work, the capability maturity model integration (CMMI), process patterns, process assessment, personal and team process models.
Process models: The water fall model, incremental process models, evolutionary process models, the unified process.

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.

System models: Context models, behavioral models, data models, object models, structured methods.

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.
Product metrics: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

UNIT- V

Metrics for Process and Products: Software measurement, metrics for software quality.

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan.

Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the

ISO9000 quality standards.

TEXT BOOKS:

1. Software Engineering, A practitioner's Approach-Roger S.Pressman, 6th edition, McGraw Hill International Edition.
2.Software Engineering- Sommerville, 7th edition, Pearson Education.
3.The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.

REFERENCE BOOKS:

1.Software Engineering, an Engineering approach-James F.Peters, WitoldPedrycz, John Wiley.
2.Software Engineering principles and practice-Waman S Jawadekar, The McGraw-Hill Companies.
3.Fundamentals of object-oriented design using UML Meiler page-Jones: Pears