Subject Code CS102602	Software Engineering and Agile	L = 3	T = 0	P = 0	Credits = 3
Examination Scheme	ESE	CT	TA	Total	ESE Duration
	100	20	30	150	3 Hours
	Minimum number of class tests to be conducted=02			Minimum Assignments=02	

Course Objectives	Course Outcomes
1. To introduce software project and to understand about the different software processes & To introduce ethical and professional issues and to explain why they are concern to software engineers 2. Understanding good coding practices, including documentation, contracts, regression tests and daily builds. 3. Their uses. To understand how Software engineering & Project Management is concerned with theories, methods and tools for professional software development.	After completion of this course, the students would be able to : 1. Developing some basic level of software architecture/design 2. Extracting and analyzing software requirements specifications for different projects 3. Select and implement different software development process models 4. Defining the concepts of software quality and reliability on the basis of International quality standards. 5. Analyzing software risks and risk management strategies 6. Applying different testing and debugging techniques and analyzing their effectiveness. **7.** Defining the basic concepts and importance of Software project management concepts like cost estimation, scheduling and reviewing the progress.

UNIT I: Introduction, Software- problem and prospects Software development process: Software life cycle models, Open source software development, the unified process, documentation, configuration management, Safety, risk assessment.

UNIT II: Measures, Metrics and Indicators, Metrics in the Process and Project Domains, Software Measurement, Metrics of Software Quality, S/W reliability, Software estimation techniques, loc and FP estimation. Empirical models like COCOMO, project tracking and scheduling, reverse engineering.

UNIT III: Software requirements and specification: feasibility study, Informal/formal specifications, pre/post conditions, algebraic specification and requirement analysis models, Specification design tools. Software design and implementation: Software design objectives, design Like, Top-Down, bottom-up, team etc. techniques, User interface design, modularity Functional decomposition Data flow design, Data structure design, Object-oriented design, Design patterns implementation strategies.

UNIT IV: Coding standard and guidelines, programming style, code sharing, code review,

software components, rapid prototyping, specialization, construction, class extensions, intelligent software agents, reuse performance improvement, debugging. Software Testing Strategies: Verification and Validation, Strategic Issues, test plan, white box, black-box testing, unit and integration testing, system testing test case design and acceptance testing, maintenance activities.

UNIT V: Agile Vs Traditional SDLC Models, Phases of Agile Model, Principles of Agile model, Agile Model - Pros and Cons, , When to use the Agile Model? , Agile Testing Methods, Scrum, Product Backlog, Scrum Practices, Process flow of Scrum Methodologies, extreme Programming (XP), Phases of eXtreme programming, Crystal Methodologies, Dynamic Software Development Method (DSDM), Feature Driven Development (FDD), Lean Software Development, KANBAN, Agile metrics.

Text Books:

S. No.	Title	Author(s)	Publisher
1	Software Engineering: A Precise Approach	PANKAJ JALOTE	Wiley India
2	Software Engineering Concepts	Fairley Richard	TMH

Reference Books:

S. No.	Title	Author(s)	Publisher
1	Software Engineering	Ian Sommer ville	Pearson Education Inc
2	Software Engineering: A Practitioners Approach	Roger S. Pressman	McGraw-Hill