Course: BTech **Semester:** 4

Prerequisite: Basic knowledge of software applications

Course Objective: This course provides a broad introduction to software engineering. The various process models required to develop software is also being described. Moreover the functional and non-functional requirements are also described

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	0	0	0	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction:** Study of Different Models, Software Characteristics, Components, Applications, Layered Technologies, Processes, Methods and Tools, Generic View Of Software Engineering, Process Models- Waterfall model, Incremental, Evolutionary process models- Prototype, Spiral And Concurrent Development Model Agile Development : Agility and Agile Process model, Extreme Programming, Other process models of Agile Development and Tools.	10	6
2	**Software Project Management:** Management Spectrum, People 'Product 'Process- Project, W5HH Principle, Importance of Team Management Planning a Software Project : Scope and Feasibility, Effort Estimation, Schedule and staffing, Quality Planning, Risk management- identification, assessment, control, project monitoring plan, Detailed Scheduling	10	5
3	**Requirements Engineering:** Problem Recognition, Requirement Engineering tasks, Processes, Requirements Specification, Use cases and Functional specification, Requirements validation, Requirements Analysis	10	5
4	**Structured System Design:** Design Concepts, Design Model, Software Architecture, Data Design, Architectural Styles and Patterns, Architectural Design, Alternative architectural designs, Modeling Component level design and its modeling, Procedural Design, Object Oriented Design. Data Oriented Analysis & Design : Difference between Data and Information, E-R Diagram, Dataflow Model, Control Flow Model, Control and Process Specification, Data Dictionary	15	5
5	**Coding and Unit Testing:** Programming principles and guidelines, Programming practices, Coding standards, Incremental development of code, Management of code evaluation, Unit testing- procedural units, classes, Code Inspection, Metrics- size measure, complexity metrics, Cyclomatic Complexity, Halstead measure,Knot Count, Comparison Of Different Metrics	10	4
6	**Software Testing and Quality Assurance:** Concepts, Psychology of testing, Levels of testing, Testing Process- test plan, test case design, Execution, Black-Box testing 'Boundary value analysis 'Pair wise testing- state based testing, White-Box testing criteria and test case generation and tool support Quality Assurance : Quality Control, Assurance, Cost, Reviews, Software Quality Assurance, Approaches to SQA, Reliability, Quality Standards- ISO9000 And 9001	15	7
7	**CASE Tools and Advance Practices of System Dependability and Security:** Computer Aided Software Engineering Tools, SCRUM Developments, Dependable System, Reliability Engineering, Safety Engineering, Security Engineering, Resilience Engineeirng	15	5

8	**Advance Software Engineering:** Software Reuse, Component Based Software Engineering, Distributed Software Engineering, Service-Oriented Software Engineering, Real-Time Software Engineering, Systems Engineering, Systems of System.	**15**	**5**

Reference Books

1.	**Software Engineering (TextBook) R.Pressmen; 6th (TextBook)**
2.	**Software Engineering** By Sommerville
3.	**Fundamentals of Software Engineering** By Rajib Mall | PHI
4.	**Software Engineering** By Pankaj Jalote | Wiley India

Course Outcome

After Learning the Course the students shall be able to:
After learning this course students will be able to : 1. Prepare and perform Software Requirement Specification and Software Project Management Plan. 2. Ensure the quality of software product, different quality standards and software review techniques 3. Apply the concept of Functional Oriented and Object Oriented Approach for Software Design. 4. Understand modern Agile Development and Service Oriented Architecture Concept of Industry 5. Analyze, design, verify, validate, implement and maintain software systems. 6. Execute a Project Management Plan, tabulate Testing Plans and Reproduce effective procedures.