

Year: B. Tech IV (Semester VII)

Subject Name: Software Engineering
Type of course: Professional Core Course
Prerequisite (if any): None

Subject Code: BTCO13702

List of Courses where this course will be prerequisite: Project, Internship

Rationale: Student will be able to learn the core principles consistent in software construction and maintenance: fundamental software processes and life-cycles, mathematical foundations of software engineering, requirements analysis, software engineering methodologies and standard notations, principles of software architecture and re-use, software quality frameworks and validation, software development, maintenance environments and tools. An introduction to function oriented and object-oriented software development process and design is also covered.

Teaching and Examination Scheme:

Teaching Scheme				Theory Marks			Practical Marks		Total
L	T	P	C	TEE	CA1	CA2	TEP	CA3	
3	0	2	4	60	25	15	30	20	150

CA1: Continuous Assessment (assignments/projects/open book tests/closed book tests CA2: Sincerity in attending classes/class tests/ timely submissions of assignments/self-learning attitude/solving advanced problems TEE: Term End Examination TEP: Term End Practical Exam (Performance and viva on practical skills learned in course) CA3: Regular submission of Lab work/Quality of work submitted/Active participation in lab sessions/viva on practical skills learned in course

Content:

Sr. No.	Content	Total Hrs
1	Overview of Software and Software Engineering : The Evolving Role of Software, Software: A Crisis on the Horizon and Software Myths, Software Engineering: A Layered Technology, Software Process Models, The Linear Sequential Model, The Prototyping Model, The RAD Model, Evolutionary Process Models	5

2	Agile Development : Agility and Agile Process model, Extreme Programming, SCRUM, Other process models of Agile Development and Tools.	4
3	Software Project Management : Software Metrics (Process, Product and Project Metrics), Software Project Estimations, Software Project Planning, Project Scheduling & Tracking, Risk Analysis & Management (Risk Identification, Risk Projection, Risk Refinement , Risk Mitigation).	5
4	Requirement Analysis and Specification : Understanding the Requirement, Requirement Modeling, Requirement Specification (SRS), Requirement Analysis and Requirement Elicitation, Requirement Engineering. Object oriented Modeling, analysis and design with various UML Diagrams(Class Diagrams, State Diagrams, Use Case Diagrams, Activity Diagrams, Sequence Diagrams)	8
5	Software Design : Design concepts and principles - Abstraction - Refinement - Modularity, Cohesion coupling, , Architectural design, Detailed Design, Function-oriented Design, User-Interface Design	6
6	Software Coding & Testing : Coding Standard and coding Guidelines, Code Review, Software Documentation, Testing Strategies, Testing Techniques and Test Case, Test Suites Design, Testing Conventional and Object Oriented Systems .	7
7	Software Maintenance and Configuration Management : Software Maintenance, Types of Maintenance, Software Configuration Management, Overview of RE-engineering, Reverse Engineering, Forward Engineering, Version Control and change control	4
8	Software Quality Assurance : Quality Concepts and Software Quality Assurance, Software Reviews (Formal Technical Reviews), Software Reliability, The Quality Standards: ISO 9000, CMM, Six Sigma for SE, SQA Plan.	3
9	Advanced Topics in Software Engineering : Component-Based Software Engineering, Client/Server Software Engineering, Computer-Aided Software Engineering, Software Process Improvement, Emerging Trends in software Engineering.	3

Suggested Specification table with Marks (Theory): (For B.Tech only)

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
15	15	10	10	5	5

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create and above Levels (Revised Bloom's Taxonomy)

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Reference Books:

Sr No	Title of book /article	Author(s)	Publisher and details like ISBN	Year of publication/ Publication Edition
1	Software Engineering: A Practitioner's Approach	Roger Pressman	McGraw- Hill	Latest Edition
2	Fundamentals of software Engineering	Rajib Mall	Prentice Hall of India	
3	Software Project Management	Bob Hughes, Mike Cotterell and Rajib Mall	Tata McGraw Hill ISBN 978-0071072748	
4	Software Engineering	Ian Sommerville	Addision-Wesley	
5	A Concise Introduction to Software Engineering	Pankaj Jalote	Springer	
6	Software Testing and Continuous Quality Improvement	William E. Lewis	Auerbach Publications	

Course Outcomes:

Sr. No.	CO statement	Marks % weightage
CO-1	Describe the knowledge of basic Software Engineering and Process Models	20
CO-2	Illustrate various software requirements using the SRS documents and UML Diagrams.	20
CO-3	Discuss the role of project management including planning, scheduling, risk management, quality control.	15
CO-4	Demonstrate software design principles using Function oriented design and object-oriented design.	15
CO-5	Examine coding standards as well as software testing techniques and strategies.	20
CO-6	Demonstrate the concepts of software maintenance, configuration, recent trends in software engineering.	10

List of Open learning website:

1. NPTEL Software Engineering Course <https://nptel.ac.in/courses/106/105/106105182/>

List of Open Source Software:

- **Software Testing Tools**
 - Selenium
 - Appium
 - Apache JMeter
 - Sikuli
- **Project Management Tools for agile teams**
 - OpenProject
 - MyCollab
 - Gitlab
 - Odo
- **DevOp Tools**
 - Docker
 - Git
 - Puppet
 - SaltStack

Major Equipment Needed: Nil

List of Experiments:

The Experiments are to be performed as a mini project in group of 3 students.

Sr.No	Practical
1	Explore Jira Tool for Software Engineering.
2	Study the complete Software Development Life Cycle (SDLC) and analyze various activities conducted as a part of various phases. For each SDLC phase, identify the objectives and summaries outcomes.
3	Consider yourself as a system analyst for the system you want to develop. Gather requirements from all the stakeholders of your system. Document the requirements in the form of question answers.
4	Create the SRS (Software Requirements Specification) document for your system.
5	Analyze your system at a high level and prepare the following UML diagrams for your system. a. Class Diagram b. Use Case Diagram c. Sequence Diagram d. Activity Diagram e. State Diagram
6	Perform structured analysis (SA) to convert analysis model into Data Flow Diagram (DFD) up to level 2.
7	Perform structured design (SD) to convert DFD into the structure chart.
8	Prepare the Timeline Chart (Gantt Chart) for your system.
9	Estimate the approximate size, effort, development time and cost for your decided system.
10	Perform implementation of any 3 modules from the modules you have designed.
11	Perform testing for the 3 modules you have implemented using black box and white box testing methods.