SARVAJANIK UNIVERSITY
Sarvajanik College of Engineering and Technology
Bachelor of Technology

Year: B.Tech III (Semester – V)

Subject Name: Software Engineering **Subject Code:** BTIT14501
Type of course: Professional Elective Course
Prerequisite: Programming for Problem solving**,** Object Oriented Programming

Rationale:
The aim of the course is to provide an understanding of the process development of quality software. This course covers software process models, requirement engineering, and software analysis modeling using unified modeling language, software design, software testing, quality concepts, and configuration management.

Teaching and Examination Scheme:

Teaching Scheme				Theory Marks			Practical Marks		Total
L	T	P	C	TEE	CA1	CA2	TEP	CA3	
3	0	0	3	60	25	15	0	0	100

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 the course

Contents:

Sr. No.	Contents	Total Hrs.
1	**Introduction to Software Engineering** The Evolving Role of Software, Changing nature of software, Software Myths, Software Engineering: Layered Technology, Software Process Models, The Waterfall Model, Incremental Process Models, Evolutionary Process Models, Component-Based Development, Unified Process, Agility and Agile Process model- Extreme Programming, Scrum	09
2	**Requirement Analysis and Modeling** Core Principles that guide process and practice, Requirement Engineering, Elicitation, Developing Use Cases, Requirement Specification(SRS), Requirement Modeling Approaches-Scenario-based modeling, Data Modeling, Class-based modeling, Flow-oriented Modeling, Behavioral model.	10
3	**Software Architecture and Design** Design Concepts and Design Principles, Architectural Design, Component Level Design (Class-based components, Traditional components), User Interface Design, Introduction to Design Patterns, basics of WebApp design	08

4	**Software Testing Strategies** Verification and Validation, Testing Strategies: Unit Testing, Integration Testing, Validation Testing, System Testing, Debugging process, Software Testing Fundamentals, White-Box Testing, Black-Box Testing, Techniques and Test Case, Test Suites Design, Introduction to object-oriented testing, Testing Web Applications	**09**
5	**Software Quality Management** Quality Concepts, Garvin's Quality Dimensions, McCall's Quality Factors, ISO 9126 Quality Factors, the cost of quality, Software Reviews, Defect amplification and removal, Formal Technical Review, Software Quality Assurance, Software Reliability, The Quality Standards: ISO 9000, CMM, Six Sigma for SE, SQA Plan	**05**
6	**Software Configuration Management:** The Software Configuration Management (SCM) Scenario, The SCM Process: Identification of Objects in the Software Configuration, Version Control, Change Control, Configuration Audit and Status Reporting	**04**

Suggested Specification table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	15	25	10	0	0

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

Reference Books:

Sr. no	**Title of book /article**	**Author(s)**	**Publisher and details like ISBN**
1.	Software Engineering- A practitioner's Approach	Roger S.Pressman	McGraw-Hill
2.	Software engineering	Ian Sommerville	Pearson Education
3.	Fundamentals of Software Engineering	Rajib Mall	PHI
4.	An Integrated Approach to Software Engineering,	Pankaj Jalote –	NAROSA

Note: Students should refer to the latest editions of books

Course Outcomes (CO):

Sr. No.	**CO statements**	**Marks % weightage**
CO-1	Ability to analyze software process models and choose a suitable approach for various real-time system development.	20%
CO-2	Identify the end-user requirement, prepare analysis models and structure the requirements in a Software Requirement Specification document.	22%

CO-3	Apply the concepts and principles for high-level architecture design, component-level design and user interface design	18%
CO-4	Apply various testing strategies and techniques and able to design test cases and plan to test Software.	20%
CO-5	Recognize how to ensure the quality of software products, and apply software review techniques and configuration management concepts.	20%

List of Open learning website and Tools:

1. https://www.modelio.org/
2. www.rspa.com/spi