Course Name: SOFTWARE ENGINEERING					
Course Code: INFO3104					
Contact Hours per week	L	T	P	Total	Credit Points
	3	0	0	3	3

Course Outcome:

After successfully completing this course the students will be able to:

1. Prepare software requirement specifications as per IEEE guidelines.
2. Model function-oriented software systems using DFD.
3. Develop object-oriented software systems using ERD, UML.
4. Analyze differentapproaches of testingmethodology in software system.
5. Estimate software size using Function PointAnalysis.
6. Work out software project schedule and staffing plan.

Detailed Syllabus:

MODULE-1[11L]
Introduction to Software Engineering [3L] Software Engineering – objectives and definitions, SDLC – different phases; Lifecycle Models - Waterfall, RAD, Prototyping, Incremental, Iterative, Spiral, Agile, Introduction to DevOps.
Requirements Phase [3L] Requirements Gathering and Analysis, Requirement Specifications – General Structure of Software Requirement Specifications (SRS).
Structured Analysis Modelling Techniques [4L] Process Model using Data Flow Diagram (DFD) with examples, Data Dictionary, Decision Tree, Decision Table with examples, Data Model using Entity Relationship Diagram (ERD) with examples.

MODULE-II [10L]
Design Phase [4L] Overview – Comparison between Requirement Analysis and Design, Attributes of Good Design , Design Approaches – Functional and Object Oriented , Design Aspects – Top-Down and Bottom-Up ,Structured Design – High Level Design and Low Level Design. Functional Decomposition – Abstraction, Cohesion, Coupling, Structure Chart, Structured English
Object Oriented Analysis and Design [6L] OOAD Basic Concepts, Unified Modelling Language (UML) – different types of diagrams for different views of system ,User View – Use Case Diagram with examples, Structural View – Class Diagram with examples, Behavioral View – Sequence, Collaboration, Activity and State Chart Diagrams with examples

MODULE-III [10L]
Coding or Programming [2L] Programming Principles and Guidelines – Structured Programming, Code Re-use, Coding Standards / Guidelines

Review and Testing [8L] Self-Review / Peer Review , Testing Overview -- Objective, Definition, Static and Dynamic Testing, Functional vs. Non-functional Testing, Testing Artifacts – Test Cases and Test Suites, Stub and Driver, Testing Methods --
White Box Testing with Test Coverage using Control Flow Graph (CFG) and Cyclomatic Complexity, Black Box Testing with Equivalence Class Partitioning and Boundary Value Analysis, Testing Level – Unit Testing, Integration Testing, System Testing, (User) Acceptance Testing, Regression Testing, Performance Testing, Usability Testing, Non-functional Testing

MODULE-IV [11L]

Software Maintenance [2L] Types of Maintenance – Corrective, Preventive, Adaptive Change Management and Maintenance Process, Introduction to Continuous Integration (CI) and Continuous Deployment (CD).

Software Estimation [3L] Overview of Software Estimation – Size, Effort, Duration and Cost ,Size Estimation Methods – Lines of Code (LOC) and Function Points (FP),Estimation of Effort and Duration based on Size and Productivity, Constructive Cost Model (COCOMO) – Basic COCOMO, Intermediate COCOMO , Detailed COCOMO . **Project Management [4L]** Project Management Overview -Planning, Staffing, Execution, Monitoring and Control, Project Scheduling – Work Breakdown Structure (WBS) ,Gantt Charts and PERT diagram.

Configuration Management [2L] Overview of Configuration Management - Identification, Control, Status Accounting, Audits, Concept of Baseline, Versioning of Configurable Items (CI).

Books

1. Software engineering - a practitioner's approach - Roger pressman, McGraw hill, New York.
2. Fundamentals of Software Engineering-Rajib Mal,PHI Learning Pvt. Ltd.
3. Software engineering-Ian sommerville, addison-wesley publishing company, England. 4. An integrated approach to software engineering-Pankaj Jalote, Narosa publishing House, New Delhi. 5. The unified modeling language User guide-Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson education, New York.