**CSE 3104 SOFTWARE ENGINEERING [3 1 0 4]**

**Course Objectives:**

* To illustrate how a given problem can be broken down into different modules.
* To construct UML diagrams.
* To demonstrate the SDLC model.

**Course Outcomes:**

* **To understand basic concepts and life cycle models**
* **Capable to estimate the project cost and manage the project**
* **To analyze the requirements of the project**
* **To model and design the project**
* **Understand the analysis and design of the project using UML**
* **Ability to use standards in coding and testing**

1. INTRODUCTION:

Evolution from an art form to an engineering discipline, Software development Projects, Exploratory style of software development, Emergence of software Engineering, Notable changes in software development practices. Computer Systems Engineering.

(Chapter 1 of Text Book 1) (3 hrs)

1. SOFTWARE LIFE CYCLE MODELS:

A few basic concepts, Waterfall model and its extensions, Rapid Application Development, Agile development models, Spiral Model , A Comparison of different Life Cycle models.

(Chapter 2 of Text Book 1) (5 hrs)

1. SOFTWARE PROJECT MANAGEMENT:

Software Project Management Complexities, Responsibilities of a software project manager, Project Planning, Metrics for Project Size Estimation, Project Estimation Techniques, Empirical estimation techniques, COCOMO- A heuristic estimation technique, Halstead’s software science – An Analytical Technique, Staffing level estimation, Scheduling, Organization and team structures, Staffing, Risk Management, Software Configuration Management, Miscellaneous plans.

(Chapter 3 of Text Book1)  (8 hrs)

1. REQUIREMENT ANALYSIS AND SPECIFICATION:

Requirement Gathering and Analysis, Software Requirement Specifications, Formal System Specification, Axiomatic Specification, Algebraic Specification, Executable Specification and 4GL.

(Chapter 4 of Text Book1) (6 hrs)

1. SOFTWARE DESIGN

Overview of the design Process, How to characterize a good software design? Cohesion and coupling, Layered arrangement of modules, Approaches to software design.

(Chapter 5 of Text Book1) (3 hrs)

1. FUNCTION-ORIENTED SOFTWARE DESIGN

Overview of SA/SD methodology, Structured analysis, Developing the DFD Model of a system, Structured design, Detailed design, Design review (Chapter 6 of Text Book1) (5 hrs)

1. OBJECT MODELLING USING UML

Basic object-orientation concepts, UML, UML diagrams, Use case model, Class diagrams, Interaction diagrams, Activity Diagram, State chart diagram, Postscript. (Chapter 7 of Text Book1) (8 hrs)

1. USER INTERFACE DESIGN:

Characteristic of good user interface, Basic Concepts, Types of UI, Fundamentals of Component based GUI development, A user interface design methodology.

(Chapter 9 of Text Book1) (3 hrs)

1. CODING AND TESTING:

Coding, Code review, Software Documentation, Testing, Unit Testing, Black-Box testing, White-Box Testing, Debugging, Program Analysis tools, Integration testing, Testing OOP, System testing, Some general issues associated with testing.

(Chapter 10 of Text Book1) (7 hrs)

**Text Books:**

1. Rajib Mall, “*Fundamentals of Software Engineering*” PHI Learning PVT. LTD, 4th Edition, 2014

**References:**

* 1. Hans Van Vliet, “*Software Engineering: Principles and Practice*”, Wiley India, 3rd Edition 2012.
  2. Roger S. Pressman, “*Software Engineering - A Practitioner’s Approach*”, McGrawHill International Edition, 7th Edition, 2010.
  3. Bernd Bruegge, Allen H. Dutoit, “*Object-Oriented Software Engineering using UML Patterns and Java*” ,Pearson Publication, Second Edition, 2011.
  4. Ian Sommerville “*Software Engineering*” Addison-Wesley, 9th Edition, 2011.