SOFTWARE ENGINEERING

		L	P	C
Course code:	**13CS2103**	**4**	**0**	**3**

Pre requisites: **Proficiency in Programming.**

Course Educational Objectives:

To provide an understanding of the various processes that software engineers may employ in developing contemporary software systems. To examine all phases of the software development life cycle, from initial planning through implementation and maintenance.

Course Outcomes:

- Students can demonstrate the skills required to analyze, design, test and maintain software systems
- Students can demonstrate an appreciation of good practices in software engineering
- Students can demonstrate the application of software quality concepts.
- Students will demonstrate an understanding of the proper contents of a software requirements document.
- Students will demonstrate an understanding of distributed system architectures and application architectures.

UNIT- I

INTRODUCTION TO SOFTWARE ENGINEERING: Software, The Nature of Software, Software Engineering, The Software Process, Software Engineering practice, Software Myths, A Generic Process Model, Process Assessment and Improvement, Product and Process, CMMI.

PROCESS MODELS: Prescriptive Process Models- The Waterfall Model, Incremental Process Models, Evolutionary Process Models, Concurrent Models. Specialized Process Models. The Unified Process, Personal and Team Process Models.

UNIT–II

SOFTWARE REQUIREMENTS: Functional and Non-functional Requirements, User Requirements, Interface Specification, the Software requirements document.

REQUIREMENTS ENGINEERING PROCESS: Feasibility Studies, Requirements Elicitation and Analysis, Requirements Validation, Requirements Management.

UNIT- III

DESIGN ENGINEERING: The Design Process, Design Concepts, the Design Model.

ARCHITECTURAL DESIGN: Software Architecture, Architectural Genres, Architectural Styles, Architectural Design, Architectural Mapping using Data Flow.

SYSTEM MODELS: Context Models, Behavioral Models, Data Models, Object Models, Structured Methods.

UNIT-IV

OBJECT ORIENTED DESIGN: Objects and Object Classes, an Object Oriented Design Process, Design Evolution.

USER-INTERFACE DESIGN: The Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, Design Evaluation.

SOFTWARE TESTING STRATEGIES: A Strategic Approach to Software Testing, Test Strategies for Conventional Software and Object Oriented Software, Validation Testing, White- Box Testing, Basis Path Testing, Black-Box Testing, System Testing.

UNIT-V

PRODUCT METRICS: A Framework for Product Metrics, Metrics for Requirements Model, Metrics for Design Model, Metrics for Source Code, Metrics for Testing, Metrics for Maintenance.

PROCESS AND PROJECT METRICS: Software Measurement, Metrics for Software Quality. **(Text Book-1)**

RISK MANAGEMENT: Reactive versus Proactive Risk Strategies, Software Risks, Risk Identification, Risk Projection, Risk Refinement, RMMM, RMMM Plan.

QUALITY MANAGEMENT: Software Quality, Informal Reviews, Formal Technical Reviews, Statistical Software Quality Assurance, Software Reliability, the ISO 9000 Quality Standards.

Text Books:

1. Roger S. Pressman: Software Engineering- A Practitioner's Approach, 6th edition , TMH, 2010.

2. Somerville: Software Engineering, 9th edition, Pearson Education, 2011.

References:

1. 1K.K.Agarwal & Yogesh Singh: Software Engineering, 3rd Edition, New Age International Publishers, 2008.

2. Shely Cashman Rosenblatt: System Analysis and Design, 2nd Edition, Thomson Publications, 2011.

3. Pankaj Jalote: An Integrated Approach to Software Engineering, 3rd Edition, Narosa Publishing House, 2011.