SOFTWARE ENGINEERING ESSENTIALS

Course Code: 22CS1104

L	T	P	C
3	0	0	3

Course Outcomes: At the end of the Course the Student will be able to:

CO1: Explain about appropriate software process models for software project/product.(L2)

CO2: Interpret the functional, non–functional requirements and requirement Engineering Process.(L2)

CO3: Choose the Architecture for a given software application.(L1)

CO4: Identify appropriate test strategies that can be applied to a given software application.(L3)

CO5: Demonstrate various Risk Management and Quality Management Techniques.(L2)

UNIT-I (10 Lectures)

INTRODUCTION TO SOFTWARE ENGINEERING: Software, The Nature of Software, The Software Process, A Generic Process Model, CMMI.

PROCESS MODELS: Prescriptive Process Models- The Waterfall Model, Incremental Process Models, Evolutionary Process Models, Concurrent Models. Specialized Process Models. The Unified Process, Agile Development**.** (Text book 1)

Learning Outcomes: At the end of the unit the student will be able to

1. Illustrate about Software Development Process & Myths. (L2)
2. Compare and contrast various life-cycle models. (L2)
3. Outline the process of software development in practice.(L2)

UNIT-II (10 Lectures)

SOFTWARE REQUIREMENTS: Functional and Nonfunctional Requirements, the software requirements document, Requirements Specification, The Requirements Engineering Process, Requirements Elicitation and Analysis, Requirements Validation, Requirements Management**.** (Text book2)

Learning Outcomes: At the end of the unit the student will be able to

1. Illustrate Functional & Non-Functional Requirements for the given project. (L2)
2. summarize the requirement engineering process.(L2)
3. explain the Software Requirements Specification Document.(L2)

UNIT-III (10 Lectures)

DESIGN ENGINEERING: The Design Process, Design Concepts, the Design Model. **ARCHITECTURAL DESIGN:**Software Architecture, Architectural Styles, Architectural Design, Architectural Mapping using Data Flow. (Text book 1)

Learning Outcomes: At the end of the unit the student will be able to

1. Make use of various architectural Styles and patterns.(L3)
2. Apply the Software Architecture for a given problem.(L3)
3. Classify a Design Model for a given problem.(L2)

UNIT-IV **(9 Lectures)**

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. (Text book 1)

Learning Outcomes: At the end of the unit the student will be able to

1. Make use of various Test Strategies.(L3)
2. Apply Equivalence partitioning for any given application(L3)
3. Choose Appropriate testing techniques to evaluate the software.(L3)

UNIT-V **(10 Lectures)**

RISK MANAGEMENT:

Reactive versus Proactive Risk Strategies, Risk Identification, Risk Projection, Risk Refinement, RMMM, RMMM Plan.

QUALITY MANAGEMENT:

Software Quality, Informal Reviews, Formal Technical Reviews, Statistical Software Quality Assurance, Software Reliability. (Text book 1)

Learning Outcomes: At the end of the unit the student will be able to

1. Explain various Software risks for a given project.(L2)
2. Summarize the software quality for a given product.(L2)
3. Demonstrate Risk Mitigation,Monitoring and Management.(L2)

TEXT BOOKS:

1. Roger S. Pressman, *Software Engineering a Practitioner's Approach*, 7th Edition, TMH, 2010.
2. Sommerville, *Software Engineering*, 9th Edition, Pearson Education, 2011.

REFERENCES:

1. K.K.Aggarwal & Yogesh Singh, *Software Engineering*, 3rd Edition, New Age International Publishers, 2008.
2. PankajJalote, *An Integrated Approach to Software Engineering*, 3rd Edition, Narosa Publishing House, 2011.

WEB REFERENCES:

1. https://nptel.ac.in/courses/Software Engineering
2. https://www.coursera.org/courses?query=software engineering
3. https://www.udemy.com/courses/development/software-engineering