

## JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

**B. Tech II - II sem (C.S.E)**

<b>T</b>	<b>Tu</b>	<b>C</b>
<b>3</b>	<b>1</b>	<b>3</b>

### **(15A05401) SOFTWARE ENGINEERING**

#### **Course Objectives**

- To understand the software life cycle models.
- To understand the software requirements and SRS document.
- To understand the importance of modeling and modeling languages.
- To design and develop correct and robust software products.
- To understand the quality control and how to ensure good quality software.
- To understand the planning and estimation of software projects.
- To understand the implementation issues, validation and verification procedures.
- To understand the maintenance of software

#### **Course Outcomes**

- Define and develop a software project from requirement gathering to implementation.
- Ability to code and test the software
- Ability to plan, Estimate and Maintain software systems

#### **Unit I:**

**Software and Software Engineering:** The Nature of Software, The Unique Nature of WebApps, Software Engineering, The Software Process, Software Engineering Practice, Software Myths

**Process Models:** A Generic Process Model, Process Assessment and Improvement, Prescriptive Process Models, Specialized Process Models, The Unified Process, Personal and Team Process Models, Process Technology, Product and Process.

**Agile Development:** Agility, Agility and the Cost of Change, Agile Process, Extreme Programming, Other Agile Process Models

#### **Unit II:**

**Understanding Requirements:** Requirements Engineering, Establishing the groundwork, Eliciting Requirements, Developing Use Cases, Building the requirements model, Negotiating Requirements, Validating Requirements.

**Requirements Modeling (Scenarios, Information and Analysis Classes):** Requirements Analysis, Scenario-Based Modeling, UML Models that Supplement the Use Case, Data Modeling Concepts, Class-Based Modeling.

**Requirements Modeling (Flow, Behavior, Patterns and WEBAPPS):** Requirements Modeling Strategies, Flow-Oriented Modeling, Creating a Behavioral Model, Patterns for Requirements Modeling, Requirements Modeling for WebApps.

#### **Unit III:**

**Design Concepts:** Design with Context of Software Engineering, The Design Process, Design Concepts, The Design Model.

**Architectural Design:** Software Architecture, Architecture Genres, Architecture Styles, Architectural Design, Assessing Alternative Architectural Designs, Architectural Mapping Using Data Flow.

**Component-Level Design:** Component, Designing Class-Based Components, Conducting Component-level Design, Component Level Design for WebApps, Designing Traditional Components, Component-Based Development.

#### **Unit IV:**

**User Interface Design:** The Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, WebApp Interface Design, Design Evaluation.

**WebApp Design:** WebApp Design Quality, Design Goal, A Design Pyramid for WebApps, WebApp Interface Design, Aesthetic Design, Content Design, Architecture Design, Navigation Design, Component-Level Design, Object-Oriented Hypermedia Design Method(OOHMD).

#### **Unit V:**

**Software Testing Strategies:** A strategic Approach to Software Testing, Strategic Issues, Test Strategies for Conventional Software, Test Strategies for Object-Oriented Software, Test Strategies for WebApps, Validation Testing, System Testing, The Art of Debugging.

**Testing Conventional Applications:** Software Testing Fundamentals, Internal and External Views of Testing, White-Box Testing, basic Path testing, Control Structure Testing, Black-Box Testing, Model-based Testing, Testing for Specialized Environments, Architectures and Applications, Patterns for Software Testing. Testing Object-Oriented Applications: Broadening the View of Testing, Testing with OOA and OOD Models, Object-Oriented Testing Strategies, Object-Oriented Testing Methods, Testing Methods Applicable at the Class level, Interclass Test-Case Design.

#### **Textbook:**

1. “Software engineering A practitioner’s Approach”, Roger S. Pressman, McGraw Hill International Education, Seventh Edition, 2016.

#### **Reference Textbooks:**

1. Fundamentals of Software Engineering, Fourth Edition, Rajib Mall, PHI,
2. Software Engineering, Ninth Edition, IAN Sommerville, Pearson, Ninth edition.
3. Software Engineering, A Precise Approach, Pankaj Jalote, Wiley India, 2010.
4. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.
5. Software Engineering1: Abstraction and modeling, Diner Bjorner, Springer International edition, 2006.
6. Software Engineering2: Specification of systems and languages, Diner Bjorner, Springer International edition, 2006.
7. Software Engineering Foundations, Yingxu Wang, Auerbach Publications, 2008.
8. Software Engineering Principles and Practice, Hans Van Vliet, 3<sup>rd</sup> edition, John Wiley & Sons Ltd.
9. Software Engineering 3: Domains, Requirements, and Software Design, D. Bjorner, Springer International Edition.
10. Introduction to Software Engineering R.J. Leach, CRC Press