

GUJARAT TECHNOLOGICAL UNIVERSITY

MASTER OF COMPUTER APPLICATIONS (MCA)

Semester: V

Subject Name: **Software Engineering (SE)**

Subject Code: **650001**

Learning Objectives: To Develop Ability to:

- Understand, Analyze and Model User's Requirements
- Select Appropriate Process Model Apply it to All Stages of Software Development Life Cycle (SDLC)
- Select and Apply Appropriate Design Methodology
- Decide the Feasibility of Using and Applying Agile Development Process
- Assure Software Quality, Select and Apply Appropriate Testing Strategies
- Select and Apply Appropriate Metrics to Estimate Software Size, Effort, and Cost
- Prepare Project Schedule, and Monitor the Project Progress
- Project, Mitigate, Monitor, and Manage Risk
- Understand the Characteristics and Applicability of Various Software Tools

Prerequisites: Systems & Object Oriented Design Methodologies

Contents:

- 1. Process Models; and Agile Development (4 Lectures)**
Introduction to Software Engineering; Prescriptive Process Models; Agile Process; Extreme Programming (XP); Brief Overview of Other Agile Process Models: Adaptive Software Development, Scrum
- 2. Principles of Software Engineering; and Requirements Modeling (7 Lectures)**
Introduction; Core Principles of Process and Practice; Principles Guiding Each Framework Activity;
Requirements Engineering; Groundwork for Understanding of Software Requirements; Overview of Eliciting Requirements, Developing Use Cases, Building the Requirements Model; Negotiating Requirements; Validating Requirements;
Requirement Modeling Strategies; Overview of Flow-Oriented Modeling, Behavioral Modeling; Requirements Modeling for WebApps
- 3. Design Concepts; and Architectural Design (5 Lectures)**
Design Concepts; Design Model;
Architectural Styles, Architectural Design; Assessing Alternative Architectural Designs; Architectural mapping Using Data Flow
- 4. Component-Level Design (4 Lectures)**
Three Views of Component; Designing Class-Based Components; Conducting Component-Level Design; Component-Level Design for WebApps; Designing Traditional Components, Component-Based Development;
- 5. User Interface Design; Design Patterns; and WebApp Design (8 Lectures)**
Golden Rules of User Interface Design; User Interface Analysis and Design; Interface Analysis; Interface Design; WebApp Interface Design; Design Evaluation;
Design Patterns; Pattern-based Software Design; Architectural Patterns; Component-Level Design Patterns; User Interface Design Patterns; WebApps Design Patterns;

WebApp Design Quality; WebApp Design Goals; Design Pyramid for WebApp; WebApp Interface Design; Aesthetic Design of WebApp; Content Design for WebApp; Architecture Design; Navigation Design; Component-Level Design; Object-Oriented Hypermedia Design Method

6. Software Review; Software Testing; and Software Metrics (10+1 Lectures)

Overview of Review Techniques **(ONLY 1 Lecture)**

A Strategic Approach to Software Testing; Strategic Issues; Test Strategies for Conventional Software; Test Strategies for Object Oriented Software; Test Strategies for WebApps; System Testing; Debugging;

Software Testing Fundamentals; White-Box Testing; Basis Path Testing; Control Structure Testing; Black-Box Testing; Testing for Specialized Environments; Patterns and Software Testing;

Overview of Testing O-O Applications; Testing OOA and OOD Models; O-O Testing Strategies; O-O Testing Methods; Testing Methods Applicable at Class Levels; Inter-Class Test Case Design;

Testing Concepts for WebApps; An Overview of Testing Process for WebApps; Content Testing; User Interface Testing; Component-Level Testing; Navigation Testing; Configuration Testing; Security Testing; Performance Testing

7. Product Metrics; and Software Project Estimation (5+1 Lectures)

Framework for Product Metrics; Metrics for Requirements Model; Metrics for Design Model; Design Metrics for WebApps; Metrics for Source Code; Metrics for Testing; Metrics for Maintenance;

Software Project Estimation; Decomposition Techniques; Empirical Estimation Models; Estimation for O_O Projects

Overview of Project Scheduling **(ONLY 1 Lecture)**

Text Book:

1. Roger S. Pressman, "Software Engineering – A Practitioner’s Approach", 7th Edition, McGraw Hill Publications

Reference Books:

1. Sommerville, "Software Engineering", 8th Edition, Pearson Education
2. Waman S. Jawadkar, "Software Engineering – Principles and Practices", TMGH Publication
3. Pankaj Jalote, "Software Engineering – A Precise Approach", Wiley India
4. Waman S. Jawadkar, "Software Engineering – A Primer", TMGH Publication
5. Shari Lawrence Pfleeger and Joanne M. Atlee, "Software Engineering – Theory and Practice", 3rd Edition, Pearson Education
6. M. G. Limaye, "Software Testing – Principles, Techniques and Tools", TMGH Publication

Chapter wise Coverage from Text book:

<i>Unit No.</i>	<i>Ch.-Sec. Nos.</i>	<i>No. of Pages</i>
1	Ch-2.3; 3.3, 3.4. 3.5.1, 3.5.2	28
2	Ch-4, 5, 7	68
3	Ch-8.3, 8.4, 9.1.1, 9.3 to 9.6	42
4	Ch-10	32
5	Ch-11, 12, 13	76
6	Ch-15 (Only Overview; 1 Lecture), 17, 18, 19, 20	96+10
7	Ch-23, 26.5 to 26.8, 27 (Only Overview; 1 Lecture)	44+10
Total	No. of Lectures = 45 (Avg. 9 Pages / Lecture)	406

Accomplishments of the student after completing the course:

- ⌚ Gain an insight into the concepts of software engineering
- ⌚ Be able to cultivate the art of building good software
- ⌚ Would be better equipped in implementing software testing using testing tools, test case designing and testing processes.