Marwadi University
Faculty of Technology
Department of Information and Communication Technology

Subject Code: 01CT0615
Subject Name: Software Engineering
B. Tech. Year – III (Semester VI)

Objective:
To understand and apply various software project management techniques based on Software Engineering guidelines and Principles.

Credits Earned: 03 Credits

Course Outcomes: After completion of this course, student will be able to:

1. Understand various software engineering principles and their application
2. Demonstrate use of various Agile methodologies for software development
3. Apply various modelling techniques for designing system requirement
4. Identify different types of risk and evaluate its impact on software system
5. Distinguish different testing strategies and Create test cases
6. Able to understand and apply the basic project management practices in real life projects

Pre-requisite of course:
Object Oriented Programming fundamental

Teaching and Examination Scheme:

Teaching Scheme (Hours)			Credits	Theory Marks			Tutorial / Practical Marks		Total Marks
				E	I		V	T	
Theory	Tutorial	Practical		ESE	IA	CSE	Viva	Term Work	
03	00	00	03	50	30	20	00	00	100

Contents:

Unit	Topics	Hours
1	**Introduction** Importance of Software Engineering, Discipline of Software Engineering; Eclipse Introduction, Overview, and Demo; Lifecycle models: Requirements Engineering, Design and Implementation, Maintenance, Software Process Model Introduction, Waterfall Process, Spiral Process, Evolutionary Prototyping Process, Agile Process, Choosing a Model, Lifecycle Documents; Version Control System: Introduction to Git, Git Demo: Git + Eclipse, Git .	06

2	**Requirements Engineering:** General RE Definition, Functional and Non-functional Requirements, User and System Requirements, Modelling Requirements, Analysing Requirements, Requirements Prioritization, Requirements Engineering Process, and steps; Creating SRS and performing requirements inspections. Engineering standards in building, testing, operation and maintenance of the computer and software systems	06
3	**OO Software and UML:** Object Orientation Introduction, UML Structural Diagrams: Class Diagrams, Component Diagram, UML Structural Diagram: Deployment Diagram. UML creation tips. **UML Behavioral Diagram**: Use Case, Use Case Diagram: Creation Tips, **UML behavioral Diagrams:** Sequence, **UML behavioral Diagrams:** State Transition Diagram. UML creation tips; **Software Architecture**: What is Software Architecture? Advantages and use of architectural models. Architectural patterns. Designing architectural patterns. Design Patterns: Patterns Catalogue, Pattern Format, Factory Method Pattern, Strategy Pattern, Choosing a Pattern, Negative Design Patterns.	12
4	**Software Testing:** Black Box Testing Failure, Fault and Error, Verification Approaches, Pros and Cons of Approaches, Testing Introduction, Testing Granularity Levels, Alpha and Beta Testing, Black-Box Testing, **Systematic Functional Testing Approach**; Test Data Selection, Equivalence Partitioning and Boundary Value Analysis, Create and Evaluate Test Case Specifications, Generate Test Cases from Test Case Specifications. **White-Box Testing:** Coverage Criteria Intro, Statement Coverage, Control Flow Graphs, Test Criteria.	09
5	**Agile Development Methods:** Cost of Change, Agile Software Development, Extreme Programming (XP), XP's Values and Principles, Test First Development, Refactoring, Pair Programming, Continuous Integration, Testing Strategy, High Level Scrum Process. **Unified Software Process:** Use-Case Driven, Inception Phase, Elaboration Phase, Construction Phase, Transition Phase, Phases and Iterations. Software Evolution: Evolution processes, Legacy Systems, Software Maintenance. Situations during software evolution and maintenance. Software Reengineering and Refactoring: Reasons to Reengineer and Refactor, Advantages, Refactoring Demo, Refactoring Risks, Cost of Refactoring, When Not to Refactor	09
Total Hours		42

Suggested Hands on activities:

1. Examine each phase of the Software Development Life Cycle (SDLC) in detail and the numerous activities that are carried out during each one. Determine the goals and summary results for each SDLC phase.
2. As a software architect or project manager, think about each project that will be produced using any technology. Create the project's Software Requirement

Specification (SRS) document.

3 Design all UML diagrams: Activity Diagram, Use-case Diagram, Data Dictionary, e E-R Diagram, Data Flow Diagram of the system.

4 Apply Basic and Intermediate COCOMO to resolve the situation.

5 Examine the case study to find the problem and fix it. In the end, an FP-oriented estimate model must be applied to evaluate the calculation portion of the effort.

6 Draw a control flow diagram and add cyclomatic complexity to the programming code (any java code). Figure out how many of separate paths needed for testing.

7 Assign a group of students a mini project to create software docs for the systems specified below,

- Create a of Software Requirements Specification (SRS)
- Function oriented design using SA/SD
- OO design using UML diagrams
- Develop a test case for given system
- Implementation of any testing method (White box)

8 Address design-based problems (DP) and open-ended issues.

Suggested Text books / Reference books:

1. R. Pressman, Software Engineering A Practitioner's Approach (8 ed.), McGraw Hill International, 2019. ISBN 978-1259253157.
2. Sommerville, Software Engineering (10 ed.), Person Publications Publishing Company, 2017. ISBN 978-9332582699.

Suggested Theory distribution:

The suggested theory distribution as per Bloom's taxonomy is as per follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process.

Distribution of Theory for course delivery and evaluation					
Remember	Understand	Apply	Analyze	Evaluate	Create
10%	20%	10%	30%	20%	10%

Supplementary Resources:

1. http://nptel.ac.in/courses/106101061/
2. https://www.joelonsoftware.com/
3. http://www.codesimplicity.com/
4. http://www.sparxsystems.com/products/ea/index.html
5. http://www.smartdraw.com
6. http://viu.eng.rpi.edu
7. www.en.wikipedia.org/wiki/Software_engineering
8. www.win.tue.nl

9. www.rspa.com/spi
10. www.onesmartclick.com/engsineering/software-engineering.html
11. www.sei.cmu.edus
12. https://www.edx.org/school/uc-berkeleyx