COURSE TITLE	**SOFTWARE ENGINEERING**
COURSE CODE	**01CT0615**
COURSE CREDITS	**3**

Objective:

1 To understand and apply various software project management techniques based on Software Engineering guidelines and Principles

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

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 modeling 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

Theory Hours	Tutorial Hours	Practical Hours	ESE	IA	CSE	Viva	Term Work
3	0	0	50	30	20	0	0

Contents : Unit	Topics	Contact 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	6
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	6

MR. CHANDRASINH DHIRUBHA PARMAR
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)

Contents : Unit	Topics	Contact Hours
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, Design Patterns : Strategy Pattern, Choosing a Pattern, Negative Design Patterns	12
4	**Software Testing** 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	9
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, 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	9
Total Hours		**42**

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
1	**Hands on activity 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
2	**Hands on activity 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.	2

MR. CHANDRASINH DHIRUBHA PARMAR
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
3	**Hands on activity 3** Design all UML diagrams: Activity Diagram, Use-case Diagram, Data Dictionary, e E- R Diagram, Data Flow Diagram of the system.	2
4	**Hands on activity 4** Apply Basic and Intermediate COCOMO to resolve the situation.	2
5	**Hands on activity 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.	2
6	**Hands on activity 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.	2
7	**Hands on activity 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)	2
8	**Hands on activity 8** Address design-based problems (DP) and open-ended issues.	2
Total Hours		**16**

Textbook :

1 Software Engineering A Practitioner's Approach , R. Pressman, McGraw Hill International, 2019

2 Software Engineering (10 ed.), Sommerville, Person Publications , 2017

References:

1 Software Engineering A Practitioner's Approach, Software Engineering A Practitioner's Approach, R. Pressman, McGraw Hill International, 2019

2 Software Engineering (10 ed.),, Software Engineering (10 ed.),, Sommerville, Person Publications Publishing Company, 2017

Suggested Theory Distribution:

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

Distribution of Theory for course delivery

MR. CHANDRASINH DHIRUBHA PARMAR
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)

Remember / Knowledge	Understand	Apply	Analyze	Evaluate	Higher order Thinking / Creative
10.00	20.00	10.00	30.00	20.00	10.00

Instructional Method:

1. The internal evaluation will be done on the basis of continuous evaluation of students in the laboratory and class-room.
2. Practical examination will be conducted at the end of the semester for evaluation of performance of students in laboratory.
3. Students may use supplementary resources such as online videos, NPTEL videos, e-courses, Virtual Laboratory, etc.
4. The course delivery method will depend upon the requirement of content and need of the students. The teacher in addition to conventional teaching method (Chalk and Talk) may use any of the tools such as demonstration, role play, Quiz, brainstorming, Flipped class, Project based learning, Collaborative learning, MOOCs etc. for effective teaching.

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

MR. CHANDRASINH DHIRUBHA PARMAR
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)