

# CAP598:SOFTWARE ENGINEERING AND PROJECT MANAGEMENT

L:3 T:0 P:0 Credits:3

**Course Outcomes:** Through this course students should be able to

- CO1 :: understand the various phases of the software development life cycle
- CO2 :: identify appropriate process model depending on the user requirements
- CO3 :: apply software engineering practices to create complex software designs
- CO4 :: analyze the need of software maintenance activities
- CO5 :: assess the software with different testing strategies

## Unit I

**Introduction to software engineering** : software engineering terminologies, software engineering practices

**Software process models** : software process, classical software development lifecycle model, prototyping model, V-model, software development life cycle(SDLC), incremental model, introduction to agile method of software development, scrum, sprint, sprint planning, daily scrum, sprint backlog

## Unit II

**Requirement engineering** : requirement gathering, requirement analysis, software requirement specification document, characteristics of a good SRS, organization of functional requirements, fit-gap analysis, requirement engineering, requirement eliciting/gathering, negotiating requirement, validating requirement, functional and non-functional requirement

## Unit III

**Design** : design process, design concepts, coupling, cohesion, data flow diagram(DFD), flow chart, architectural design, component based design, object oriented design, class based components, use case diagram, class diagram, activity diagram, sequence diagram, collaboration diagram

## Unit IV

**User interface design** : golden rules, interface design models, interface design process, interface design activities

**Standards** : good coding practices, coding standards, code reusability, documentation, documentation standards

## Unit V

**Software testing** : software testing introduction, software testing terminologies, software testing lifecycle

**Testing strategies** : black box testing and its methods, test planning, test case design

**Automated testing with selenium** : introduction to Selenium IDE, creating test cases and suites using Selenium IDE commands, using Javascript with Selenium(variables manipulation)

## Unit VI

**Product metrics** : metrics and indicators, introduction to measures, introduction to COCOMO model, function based metrics(function point analysis)

**Software maintenance and metrics** : need for software maintenance, business process reengineering, reverse engineering, types of software maintenance

## Text Books:

1. SOFTWARE ENGINEERING: A PRACTITIONER'S APPROACH by ROGER S. PRESSMAN, BRUCE R. MAXIM, MC GRAW HILL
2. FUNDAMENTALS OF SOFTWARE ENGINEERING by RAJIB MALL, PHI Learning

## References:

1. SOFTWARE ENGINEERING PRINCIPLES & PRACTICES by RAJESH NARANG, MCGRAW HILL EDUCATION
2. INTEGRATED APPROACH TO SOFTWARE ENGINEERING by PANKAJ JALOTE, NAROSA PUBLISHING HOUSE

