SOFTWARE ENGINEERING

V Semester: **CSE |** **IV Semester:** **IT**

Course Code	Category	Hours / Week			Credits	Maximum Marks		
		L	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
ACS008	**Core**	3	1	-	4	30	70	100
Contact Classes: 45	**Tutorial Classes: 15**	**Practical Classes: Nil**				**Total Classes: 60**		

I. COURSE OVERVIEW:

This course concentrates on developing basic understanding about various activities that are involved in a software development. This course enables the student to develop necessary skills for developing a product or applications. The course focuses on all activities involved in software development (communication, planning, modeling, construction, deployment).In this course; students will gain a broad understanding of the discipline of software engineering and its application to the development and management of software systems. Student can implement and get knowledge about development of the software and gains knowledge of basic engineering methods and practices, and their appropriate application.

II. OBJECTIVES:

The course should enable the students to:

I. Learn how to elicitate requirements and develop software life cycles.
II. Understand the design considerations for enterprise integration and deployment.
III. Analyze testing methodologies.
IV. Prepare a project plan for a software project that includes estimates of size and effort, a schedule, resource allocation, configuration control, and project risk.

III. COURSE OUTCOMES:

After successful completion of the course, students should be able to:

CO 1	**Describe** process models, approaches and techniques for managing a software development process.	Understand
CO 2	**Recognize** the importance project planning activities that accurately help in selection and initiation of individual projects andof portfolios of projects in the enterprise.	Understand
CO 3	**Explain** software model and behavior of a software system.	Understand
CO 4	**Develop** the approaches to verification and validation including static analysis and reviews.	Apply
CO 5	**Demonstrate** the concept of risk management through risk identification, risk measurement and mitigation.	Understand
CO 6	**Make use of** earned value analysis and project metric for schedulingand improving the quality of software.	Analyze

IV. SYLLABUS:

UNIT-I	**SOFTWARE PROCESS AND PROJECT MANAGEMENT**	**Classes: 08**
Introduction to software engineering, software process, perspective and specialized process models; Software project management: Estimation: LOC and FP based estimation, COCOMO model; Project scheduling: Scheduling, earned value analysis, risk management.		
UNIT-II	**REQUIREMENTS ANALYSIS AND SPECIFICATION**	**Classes: 09**
Software requirements: Functional and nonfunctional, user requirements, system requirements, software requirements document; Requirement engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management; Classical analysis: Structured system analysis, petri nets, data dictionary.		

UNIT-III	SOFTWARE DESIGN	Classes: 09

Design process: Design concepts, design mode, design heuristic, architectural design architectural styles, architectural design, and architectural mapping using data flow.

User interface design: Interface analysis, interface design; Component level design: Designing class based components, traditional components.

UNIT-IV	IMPLEMENTATION AND TESTING	Classes: 10

Software testing fundamentals: Internal and external views of testing, white box testing, basis path testing, control structure testing, black box testing, regression testing, unit testing, integration testing, validation testing, system testing and debugging; Software implementation techniques: Coding practices, refactoring.

UNIT-V	PROJECT MANAGEMENT	Classes: 09

Estimation: FP based, LOC based, make/buy decision; COCOMO II: Planning, project plan, planning process, RFP risk management, identification, projection; RMMM: Scheduling and tracking, relationship between people and effort, task set and network, scheduling; EVA: Process and project metrics.

Text Books:

1. Roger S. Pressman, “Software Engineering – A Practitioner’s Approach”, Tata McGraw-Hill International Edition, 7th Edition, 2010.
2. Ian Somerville, “Software Engineering”, Pearson Education Asia, 9th Edition, 2011.

Reference Books:

1. Rajib Mall, “Fundamentals of Software Engineering”, PHI Learning Private Limited, 3rd Edition, 2009.
2. Pankaj Jalote, “Software Engineering, A Precise Approach”, Wiley India, 1st Edition, 2010.

Web References:

1. http://www.softwareengineerinsider.com/articles/what-is-software-engineering.html
2. https://www.udacity.com/courses/software-engineering
3. http://www.tutorialspoint.com/software_engineering
4. http://computingcareers.acm.org/?page_id=12
5. http://en.wikibooks.org/wiki/Introduction_to_Software_Engineering

E-Text Books:

1. http://www.acadmix.com/eBooks_Download
2. http://www.freetechbooks.com/software-engineering-f15.html

Course Home Page: