

<b>COURSE CODE</b>	<b>COURSE TITLE</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>1154CS105</b>	<b>SOFTWARE ENGINEERING</b>	<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**Course Category: University Elective**

**A. Preamble :**

Software engineers are those who contribute by direct participation or by teaching, to the analysis, specification, design, development, certification, maintenance, and testing of software systems

**B. Prerequisite Courses:**

<b>SI No</b>	<b>Course Code</b>	<b>Course Name</b>
1	1150CS201	Problem solving using C

**C. Related Courses:**

<b>SI No</b>	<b>Course Code</b>	<b>Course Name</b>
NIL		

**D. Course Educational Objectives :**

Students undergoing this course are expected to

- Understand conventional software management, software economics evolution
- Get the knowledge about life cycle phases, iterative process planning, organization and responsibilities and process automation can be understood
- Learn modern project profiles, next generation of software economics and modern project transition can be understood

**E. Course Outcomes :**

Upon the successful completion of the course, students will be able to:

<b>CO Nos.</b>	<b>Course Outcomes</b>	<b>Knowledge Level (Based on revised Bloom's Taxonomy)</b>
CO1	Identify an appropriate process model to develop a software	K3
CO2	Summarize the principles involved in gathering and validating software requirements	K2
CO3	Illustrate the knowledge on testing methods and comparison of various testing techniques.	K2
CO4	Infer the quality assurance procedures during software development.	K2
CO5	Summarize the current technologies, by managing resources economically and keeping ethical values in software engineering and maintenance Explain software project management and software maintenance practices	K2

## F. Correlation of COs with POs :

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	L	M							L						
CO2	L	L	H												L
CO3	L			H	L			L						L	
CO4	L			M											
CO5	L	L		M							M				

H- High; M-Medium; L-Low

### Course Content:

#### UNIT I INTRODUCTION 9

Introduction to Software Engineering, Software Process, Perspective and Specialized Process Models – Unified Process-Personal and Team Process Model- Agile development-Agile Process- Extreme Programming-Other Agile Process Models- A Tool set for the agile process.

#### UNIT II REQUIREMENTS ANALYSIS AND DESIGN 9

Software Requirements Document- Understanding Requirements- Design process – Design Concepts-Design Model– Design Heuristic – Architectural Design – Architectural styles, Architectural Mapping using Data Flow- User Interface Design: Interface analysis, Interface Design –Component level Design-Pattern Based Design

#### UNIT III TESTING AND IMPLEMENTATION 9

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-Testing Tools

#### UNIT IV QUALITY MANAGEMENT 9

Quality Concepts- Review Techniques- Software Quality Assurance- Software Configuration management- Product and Project Metrics- Software Estimation- Project Scheduling Risk management

#### UNIT V MAINTENANCE AND REENGINEERING 9

Software Maintenance and reengineering-Software Process Improvement- Emerging Trends in Software Engineering

**Total: 45 Hour**

## **G. Learning Resources**

### **i. Text Books :**

1. Roger. S. Pressman and Bruce R. Maxim, “Software Engineering – A Practitioner’s Approach”, eighth Edition, McGraw Hill ,International Edition, Singapore, 2015.
2. Ian Sommerville, “Software **Engineering**”, sixth Edition, Pearson Education, New Delhi, 2001.
3. Microsoft Project 2007 for Dummies.

### **ii. Reference:**

1. Ali Behforooz, Frederick J Hudson, “Software Engineering Fundamentals”, second edition, Oxford University Press, Noida, 2003.
2. Fairley R, “Software Engineering Concepts”, second edition, Tata McGraw Hill, New Delhi, 2003.
3. Jalote P, “An Integrated Approach to Software Engineering”, third edition, Narosa Publishers, New Delhi, 2005.

### **iii. Online resources**

1. <http://www.slideshare.net/rhspcte/software-engineering-ebook-roger-s-pressman>
2. <https://docs.google.com/folderview?id=0B2Q8Nd2L-6PjZDVmZjg3ZDUtNTYwMy00ODBITk4NzQtYWZmM2I3OWI0MDg4&tid=0B2Q8Nd2L-6PjMmNlYzk3YmMtZWQxZC00ODdmLWE5N2MtZmFmNjY5ZjY0NGJj>
3. [www.londoninternational.ac.uk/sites/default/files/.../co3353\\_ch1-3.pdf](http://www.londoninternational.ac.uk/sites/default/files/.../co3353_ch1-3.pdf)
4. [users.encs.concordia.ca/~grogono/CourseNotes/COMP-354-Notes.pdf](http://users.encs.concordia.ca/~grogono/CourseNotes/COMP-354-Notes.pdf)
5. [ufesmestradoanabringuente.googlecode.com/.../Thayer-SE-Proj-Mgt.pdf](http://ufesmestradoanabringuente.googlecode.com/.../Thayer-SE-Proj-Mgt.pdf)
6. [devedzic.fon.rs/publications/SEKE-Handbook-2.pdf](http://devedzic.fon.rs/publications/SEKE-Handbook-2.pdf)