

## 2/3 MCA First Semester

CA3T3

SOFTWARE ENGINEERING

Credits : 4

Lecture Hours : 4 periods / week

Internal assessment : 30 Marks  
Semester and Examination: 70 Marks

---

### Course Description:

Software Engineering is a core subject that is part of 3<sup>rd</sup> semester in the curriculum. In order to take this course, a student should have a requisite knowledge of what is software, object-oriented concepts introduction and basic programming.

This course provides an overview of the software engineering discipline, introducing the student to the fundamental principles and methods in software engineering. The course highlights the need for an engineering approach to software. It provides an opportunity for the students to gain knowledge of industrial approach to real-world projects and importance of team environment.

The course teaches the student various methods and models to gather the user requirements, analyze them, and model the specified requirements both functional and non functional including system constraints. The course also aims at teaching various existing design models and patterns for different domains. Design methodologies including structured design and object-oriented design are also covered. Students will learn the implementation and testing concepts in developing a successful software solution that meets the specified requirements.

This course discusses the importance of software as layered technology and emphasis the application of tool based software engineering. The course addresses the necessity of various umbrella activities of software engineering such as software project Management, Formal Technical Reviews, software Quality assurance, Configuration management, Reusability Management, Measurement and metrics, Risk Management.

Tutorials, Case study assignments, and group discussions focused on projects and problem solving will provide practice in the application of theory and procedures. These allow students to explore concepts with teachers and other students, and receive feedback. The course will contribute to the development of Critical Analysis, Problem Solving, Communication and Teamwork.

### Course Objectives:

- Facilitate an understanding of the software development lifecycle, and the engineering practices that define it.
- To help students to develop skills that will enable them to construct software of high quality.
- Summarizes the software that is reliable, and that is reasonably easy to understand, modify and maintain.

### UNIT I :

**Introduction to Software Engineering :** The evolving role of software, Changing Nature of Software, Software myths. A Generic view of process : Software engineering- A layered technology, a process framework, The Capability Maturity Model Integration (CMMI), Process patterns, process assessment, personal and team process models.

### UNIT II :

**Process models :** The waterfall model, Incremental process models, Evolutionary process models, The Unified process. Software Requirements : Functional and non-functional requirements, User requirements,

System requirements, Interface specification, the software requirements document.

**UNIT III :**

**Requirements engineering process:** Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management. System models : Context Models, Behavioral models, Data models, Object models, structured methods.

**UNIT IV :**

**Design Engineering :** Design process and Design quality, Design concepts, the design model. Creating an architectural design : Software architecture, Data design, Architectural styles and patterns, Architectural Design.

**UNIT V :**

**Object-Oriented Design:** Objects and object classes, An Object-Oriented design process, Design evolution. Performing User interface design : Golden rules, User interface analysis and design, interface analysis, interface design steps, Design evaluation.

**UNIT VI :**

**Testing Strategies:** A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing, the art of Debugging. Product metrics : Software Quality, Metrics for Analysis Model, Metrics for Design Model, Metrics for source code, Metrics for testing, Metrics for maintenance.

**UNIT VII :**

**Metrics for Process and Products:** Software Measurement, Metrics for software quality. Risk Management : Reactive vs. Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM Plan.

**UNIT VIII :**

**Quality Management:** Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software quality Assurance, Software reliability, The ISO 9000 quality standards.

**Learning Resources**

**Text Books:**

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6/e, 2010.
2. Software Engineering, Ian Sommerville, Pearson Education 7/e, 2008.

**Reference Books:**

1. Software Engineering, A practical Approach, Pankaj Jalote, Wiley, 2010.
- Software Engineering, James PHI, 1/e, 2008

