

# SOFTWARE ENGINEERING CT

**Lecture : 3**                      **Year : III**  
**Tutorial : 1**                    **Part : I**  
**Practical : 1.5**

## Course Objectives:

This course provides a systematic approach towards planning, development, implementation and maintenance of system, also help developing software projects.

### 1. Software Process and requirements (12 hours)

- 1.1. Software crisis
- 1.2. Software characteristics
- 1.3. Software quality attributes
- 1.4. Software process model
- 1.5. Process iteration
- 1.6. process activities
- 1.7. Computer-aided software engineering
- 1.8. Functional and non –functional requirements
- 1.9. User requirements
- 1.10. System requirement
- 1.11. Interface specification
- 1.12. The software requirements documents
- 1.13. Feasibility study
- 1.14. Requirements elicitation and analysis
- 1.15. Requirements validation and management

### 2. System models (3 hours)

- 2.1. Context models
- 2.2. Behavioural models
- 2.3. Data and object models
- 2.4. Structured methods

### 3. Architectural design (6 hours)

- 3.1. Architectural design decisions
- 3.2. System organization
- 3.3. Modular decomposition styles
- 3.4. Control styles
- 3.5. Reference architectures
- 3.6. Multiprocessor architecture
- 3.7. Client –server architectures
- 3.8. Distributed object architectures
- 3.9. Inter-organizational distributed computing

### 4. Real –time software design (3 hours)

- 4.1. System design
- 4.2. Real-time operating systems
- 4.3. Monitoring and control systems
- 4.4. Data acquisition systems

### 5. Software Reuse (3 hours)

- 5.1. The reuse landscape
- 5.2. Design patterns
- 5.3. Generator –based reuse
- 5.4. Application frameworks
- 5.5. Application system reuse

### 6. Component-based software engineering (2 hours)

- 6.1. Components and components models
- 6.2. The CBSE process
- 6.3. Component composition

### 7. Verification and validation (3 hours)

- 7.1. Planning verification and validation
- 7.2. Software inspections
- 7.3. Verification and formal methods
- 7.4. Critical System verification and validation

### 8. Software Testing and cost Estimation (4 hours)

- 8.1. System testing
- 8.2. Component testing
- 8.3. Test case design
- 8.4. Test automation
- 8.5. Metrics for testing
- 8.6. Software productivity
- 8.7. Estimation techniques
- 8.8. Algorithmic cost modeling
- 8.9. Project duration and staffing

### 9. Quality management (5 hours)

- 9.1. Quality concepts
- 9.2. Software quality assurance
- 9.3. Software reviews
- 9.4. Formal technical reviews
- 9.5. Formal approaches to SQA
- 9.6. Statistical software quality assurance
- 9.7. Software reliability
- 9.8. A framework for software metrics
- 9.9. Matrices for analysis and design model
- 9.10. ISO standards
- 9.11. CMMI
- 9.12. SQA plan
- 9.13. Software certification

### 10. Configuration Management (2 hours)

- 10.1. Configuration management planning
- 10.2. Change management
- 10.3. Version and release management
- 10.4. System building
- 10.5. CASE tools for configuration management

**Practical**

The laboratory exercises shall include projects on requirements, analysis and designing of software system. Choice of project depend upon teacher and student, case studies shall be included too.

Guest lecture from software industry in the practical session.

**References:**

1. Ian Sommerville, Software Engineering , Latest edition
2. Roger S. Pressman, Software Engineering –A Practitioner’s Approach, Latest edition
3. Pankaj Jalote, Software Engineering-A precise approach, Latest edition
4. Rajib Mall, Fundamental of Software Engineering, Latest edition

**Evaluation Scheme:**

The questions will cover all the chapters in syllabus. The evaluation scheme will be as indicated in the table below:

Chapters	Hours	Marks distribution*
1	12	20
2	3	5
3	6	10
4	3	5
5	3	5
6	2	3
7	5	10
8	4	8
9	5	10
10	2	4
Total	45	80

\*There may be minor deviation in marks distribution