

**Code No.: ETCS 202**  
**Paper: Software Engineering**

**L T C**  
**3 1 4**

**INSTRUCTIONS TO PAPER SETTERS:**

**MAXIMUM MARKS: 75**

1. Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 25 marks.
2. Apart from question no. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be of 12.5 marks.

**UNIT – I**

**Introduction:** Software Crisis, Software Processes & Characteristics, Software life cycle models, Waterfall, Prototype, Evolutionary and Spiral Models

**Software Requirements analysis & specifications:** Requirement engineering, requirement elicitation techniques like FAST, QFD & Use case approach, requirements analysis using DFD, Data dictionaries & ER Diagrams, Requirements documentation, Nature of SRS, Characteristics & organization of SRS.  
**[No. of Hrs.: 11]**

**UNIT – II**

**Software Project Planning:** Size Estimation like lines of Code & Function Count, Cost Estimation Models, COCOMO, COCOMO-II, Putnam resource allocation model, Risk Management.

**Software Design:** Cohesion & Coupling, Classification of Cohesiveness & Coupling, Function Oriented Design, Object Oriented Design  
**[No. of Hrs.: 11]**

**UNIT - III**

**Software Metrics:** Software measurements: What & Why, Token Count, Halstead Software Science Measures, Design Metrics, Data Structure Metrics, Information Flow Metrics

**Software Reliability:** Importance, Hardware Reliability & Software Reliability, Failure and Faults, Reliability Models, Basic Model, Logarithmic Poisson Model, Software Quality Models, CMM & ISO 9001.  
**[No. of Hrs.: 11]**

**UNIT - IV**

**Software Testing:** Testing process, Design of test cases, functional testing: Boundary value analysis, Equivalence class testing, Decision table testing, Cause effect graphing, Structural testing, Path Testing, Data flow and mutation testing, Unit Testing, Integration and System Testing, Debugging, Alpha & Beta Testing, Testing Tools & Standards.

**Software Maintenance:** Management of Maintenance, Maintenance Process, Maintenance Models, Regression Testing, Reverse Engineering, Software Re-engineering, Configuration Management, Documentation.  
**[No. of Hrs.: 11]**

**TEXT BOOKS:**

1. K. K. Aggarwal & Yogesh Singh, “Software Engineering”, 2<sup>nd</sup> Ed., New Age International, 2005.
2. R. S. Pressman, “Software Engineering – A practitioner’s approach”, 5<sup>th</sup> Ed., McGraw Hill Int. Ed., 2001.

**REFERENCE BOOKS:**

1. Stephen R. Schach, “Classical & Object Oriented Software Engineering”, IRWIN, TMH, 1996.
2. James Peter, W. Pedrycz, “Software Engineering: An Engineering Approach”, John Wiley & Sons, 2004.

3. I. Sommerville, "Software Engineering", Addison Wesley, 2004
4. K. Chandrasekhkar, "Software Engineering & Quality Assurance", BPB, 2005.