

Software Engineering (IT – 5001)

Course Code	IT-5001	Credits-4	L -3, T-1, P-0
Name of the Course	Software Engineering		
Lectures to be Delivered	52 (1 Hr Each) (L= 39, T = 13 for each semester)		
Semester End Examination	<i>Max. Marks: 100</i>	<i>Min. Pass Marks: 40</i>	Maximum Time:3hrs
Continuous Assessment (based on sessional tests (2) 50%, Tutorials/Assignments 30%, Quiz/Seminar 10%, Attendance 10%)			<i>Max. Marks: 50</i>

Instructions

- For Paper Setters: The question paper will consist of five sections A, B, C, D, and E. Section E will be Compulsory, it will consist of a single question with 10-20 subparts of short answer type, which will cover the entire syllabus and will carry 40% of the total marks of the semester end examination for the course. Section A, B, C and D will have two questions from the respective sections of the syllabus and each question will carry 15% of the total marks of the semester end examination for the course.
- For Candidates:** Candidates are required to attempt five questions in all selecting one question from each of the sections A, B, C and D of the question paper and all the subparts of the questions in section E. Use of non-programmable calculators is allowed.

Section – A

Introduction- Need for Software Engineering, issue in the design of large software, software life cycle models, overview of Software development process.

Software Requirements Analysis and Specification: Requirements Engineering, Crucial process step, state of the practice, problem analysis; data dictionaries, entity relationship diagram, coad object diagram, approaches to problem analysis, Structured requirements definition, Structured analysis & design technique, Software Prototyping, Software requirements specification; nature of the SRS, characteristics of a good SRS, Organization of the SRS, Specifying Behavioral requirements; Finite state machine, decision tables & tree, PDL.

Section – B

Software Metrics: What & Why: Definition, areas of applications, problems during implementation, size Metrics; The basic information Flow Model, the more sophisticated Information Flow Model, Metrics analysis: Using Statistics for Assessment, Problems with metric Data, the common of pool of Data, A pattern for successful applications.

Software Project Planning: Cost Estimation: Models, Static, single Variable Model, Static, Multivariable Models, The constructive Cost Model: Basic Model, International Model, Detailed COCOMO Model, The Putnam Resource Allocation Model: The trade-off-of-time versus cost, Development sub-cycle, software Risk Management: What is Risk, Typical Software Risks, Risk Management Activities; Risk Identification, Risk Projection, Risk Management Activity.

Section – C

Software Testing Techniques: Software Testing Fundamental testing objectives, testing principles, testability, test case design, white box testing, flow graph notation, cyclomatic complexity, driving test cases, graph matrices, black block testing, graph base testing methods, equalization partitioning, comparison testing, orthogonal Array testing, testing for real time system.

Software Testing Strategies: Strategic approach to software testing, Verification and validation, unit testing, unit test procedures, Integration Testing, Top down Integration, Bottom Up Integration, regression testing, Smoke testing, Validation Testing, Alpha testing and Beta Testing, system testing; recovery testing, Security testing, Stress testing, performance testing.

Section – D

Software Maintenance: What is software maintenance; Categories of Maintenance, Problem during maintenance, Potential solution to maintenance problems, the maintenance process: program understanding, generating particular

maintenance proposal, Ripple Effect, Modified program testing, maintenance Models; Quick fix model, Iterative enhancement model, Reuse oriented model, Boehm's model Estimation of maintenance cost; Belady & Lehman model, Boehm model, Configuration Management; Configuration management activities, software version, change control process.

Software Quality Assurance: Quality Concepts, quality, quality control, quality assurance, cost of quality, SQA Activities, cost impact of defects, defect application and removal, Review meeting, reporting and record keeping, statistical software quality assurance, software reliability, measure of reliability and availability.

Books:

1. Software Engineering – A practitioner' Approach, Roger. S. Pressman
2. Software Engineering – K.K. Aggarwal & Yogesh