

SOFTWARE ENGINEERING (ECS-202)

Teacher Name:

Dr. Mousumi Dhara

Course Structure

Sr. No.	Course Type	Course Code	Course Name	Credits	Details of Sessional Marks				ESM	Total Marks
					CT	TA	Lab	Total		
1	PCC	ECS-202	Software Engineering	5 (3-1-2)	15	20	15	50	50	100

Prerequisite:

Course Content:

Unit-1:

Introduction to Software Engineering, Software Components, Software Characteristics, Software Crisis, Software Engineering Processes, Similarity and Differences from Conventional Engineering Processes, Software Development Life Cycle (SDLC) Models: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models, Selection of Software Development Models,

Unit-2:

Software Requirement Specifications (SRS) Requirement Engineering Process: Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study, Information Modeling, Data Flow Diagrams, Entity Relationship Diagrams, Decision Tables, SRS Document, IEEE Standards for SRS, Estimation of various Parameters such as Cost, Efforts, Schedule/Duration, Constructive Cost Models (COCOMO), Resource Allocation Models, Software Risk Analysis and Management.

Unit-3:

Software Design Basic Concept of Software Design, Architectural Design, Low Level Design: Modularization, Design Structure Charts, Pseudo Codes, Flow Charts, Coupling and Cohesion Measures, Design Strategies: Function Oriented Design, Object Oriented Design, Top-Down and Bottom-Up Design. Software Measurement and Metrics: Various Size Oriented Measures:

Halestead's Software Science, Function Point (FP) Based Measures, Cyclomatic Complexity Measures: Control Flow Graphs.

Unit-4:

Software Testing Objectives, Unit Testing, Integration Testing, Acceptance Testing, Regression Testing, Testing for Functionality and Testing for Performance, Top-Down and Bottom-Up Testing Strategies: Test Drivers and Test Stubs, Structural Testing (White Box Testing), Functional Testing (Black Box Testing), Test Data Suit Preparation, Alpha and Beta Testing of Products. Static Testing Strategies: Formal Technical Reviews (Peer Reviews), Walk Through, Code Inspection, Compliance with Design and Coding Standards.

Unit-5:

Software Maintenance: Software as an Evolutionary Entity, Need for Maintenance, Categories of Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Software Re-Engineering, Reverse Engineering, Software Configuration Management Activities, Change Control Process, Software Version Control, Defect Detection and Removal: Defect Amplification Model, An Overview of CASE Tools.

[UNIT 1](#)

[UNIT 2](#)

[UNIT 3](#)

[UNIT 4](#)

[UNIT 5](#)

Lab Work:

Lab exercises or a Mini Project (as per list given below) to be carried out using languages like C++, Java, C# and tools like Visio, ARGOUML, Rational Rose etc. Design and Implementation of an Object based application using any one of the above languages/tools is desirable.

- Hotel Automation System
- Book Shop Automation Software
- Word processing Software
- Software Component Cataloguing Software
- Payroll System
- Banking System
- Purchase Order System
- Library Management System

- Railway Reservation System
- Bill Tracking System
- University Admission System
- Estate Management System.

Text and References Books:

1. R. S. Pressman, Software Engineering: A Practitioners Approach, McGraw Hill.
2. K. K. Aggarwal and Yogesh Singh, Software Engineering, New Age International Publishers.
3. Ian Sommerville, Software Engineering, Addison Wesley.
4. Pankaj Jalote, Software Engineering, Narosa Publication
5. Pfleeger, Software Engineering, Macmillan Publication.

Course Outcomes:

1. Understand and explain various concepts of software engineering and software life cycle development models. (Understand)
2. Prepare SRS and Compute cost and effort required to complete a given project, using various estimation techniques and models. (Apply)
3. Understand various concepts of Software design and Construct Data Flow Diagrams, Data Dictionaries and UML diagrams for a given software requirement specification. (Understand, Apply)
4. Understand various testing techniques and use these concepts to design optimal test cases. (Understand, Apply, Analyze)
5. Understand software configuration management, version control, reverse engineering, defect tracking etc. (Understand)
6. Build a project report as a team which contains the requirement specification, plan, schedule and design documents based on the knowledge of software development lifecycle. (Apply)

Assignments

Assignment#1

1. What do you mean by prototyping in a SDLC model?
2. Draw a decision table and decision tree for educational management system.
3. Discuss significant points regarding SRS document.
4. How SRS is helpful in structural design and analysis.
5. What is the role of validation and verification for SDLC?
6. Describe disadvantages of iterative waterfall modelling approach

Assignment#2

1. Define the meaning of software quality and detail the factors which affect the quality not productivity of a software product?
2. Define the meaning of quality assurance. Explain the role of testing in Quality assurance.
3. What is the major difference between structured English and Pseudo Code?
4. What is mean by level-0 Data flow diagram?
5. Describe the difference between Interface-oriented, Object-oriented and Aspect-oriented programming.

Assignment#3

1. What are the difference between white box testing and black box testing techniques?
2. What are the difference between alpha testing and Beta testing?
3. Explain software reliability and define how software and hardware reliability related to each other.
4. Write short note on Black box testing, White box testing and Stress Testing.
5. What are CASE tools?