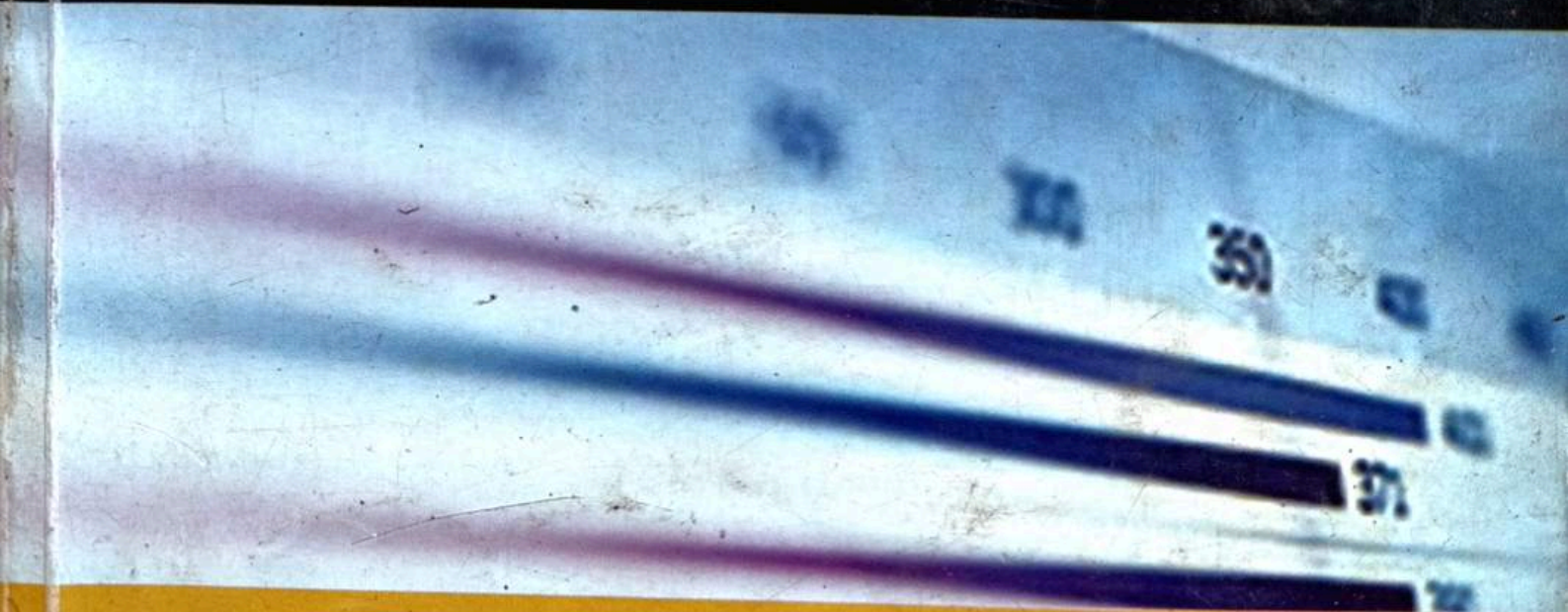
New Edition

Software Engineering

BCA/CSIT



Ramesh Singh Saud

Manoj Giri

BCA Syllabus

Course Title: Software Engineering (3 Cr.) **Full Marks: 60 + 20 + 20**
Course No: CACS253 **Pass Marks: 24 + 8 + 8**
Year/Semester: II/IV
Class Load: 4 Hrs. Week (Theory: 3 Hrs, Tutorial : 1)
Course Description:

This course includes the topics that provide fundamental concept and standard of software engineering so that students will be able to develop software and/or handle software project using the global standard of software.

Course Objectives:

This course is designed to provide the students with the basic competencies required to identify requirements, documents the system design and maintain a developed system. It presumes a general understanding of computer and programming which are covered in the first and second semester of the degree.

Course Content

Unit 1 Introduction **4 Hrs**

Definition of Software, Type of Software, Characteristics of Software, Attributes of Goods Software, Definition of Software Engineering, Software Engineering Costs, Key Challenges that Software Engineering Facing, System Engineering and Software Engineering, Professional Practices.

Unit 2 Software Development Process Model **8 Hrs**

Software Development Process Model: The Waterfall Model, Evolutionary Development, Component-Based Software Engineering (CBSE); Process Iteration: Incremental Delivery, Spiral Development; Rapid Software Development: Agile Methods, Extreme Programming, Rapid Application Development, Software Prototyping; Rational Unified Rapid (RUP), Computer Aided Software Engineering (CASE): Overview of CASE Approach Classification of CASE tools.

Unit 3 Software Development Process Model **10 Hrs**

System and Software Requirements Type of Software Requirements: Functional and Non-Functional Requirements, Domain Requirements, User Requirements; Elecitation and Analysis of Requirements: Overview of Techniques, View Points, Interviewing, Scenarios, Use-Case, Ethnography, Requirement Validation, Requirement Specification, Feasibility.

Unit 4 Software Design **10 Hrs**

Design Concept: Abstraction, Architecture Patterns, Modularity: Cohesion, Coupling; Information Hiding, Functional Independence, Refinement; Architecture Design: Repository Model, Client Server Model, Layered Model, Modular Decomposition; Procedural Design Using Structure Methods, User Interface Design: Human Computer, Information Presentation, Interface Evaluation; Design Notation.

Unit 5 Coding **2 Hrs**

Programming Language and Development Tools, Selecting Language and Tools Good Programming Practices

Unit 6 Software Maintenance **6 Hrs**

Verification and Validation, Techniques of Testing: Black-box and White-box Testing, inspections: level of testing: Unit Testing, Integration Testing, Interface Testing, System Testing, Alpha and Beta Testing, Regression Testing: Design of Test Cases, Quality Management Activities, Product and Process Quality, Standard: ISO 9000, capability Maturity Model (CMM)

Unit 7 Software Maintenance 3 Hrs

Evolving Nature of Software, Different Types of Maintenance: Fault Repair, Software Adaptation, Functionality Addition or Modification; maintenance prediction, Re-Engineering, Configuration Management (CM): Importance of CM, Conficguration items, Versioning;

Unit 8 Managing Software Projects 2 Hrs

Needs for the Proper management of Software Projects, Management Activities: Project Planning, Estimating Costs, Project scheduling, Risk Management, managing People.

Teaching Methods

The general teaching pedagogy includes class lectures, group discussion, case studies, guest lectures, research work, project work, assignments (theoretical and practical), and examination (write and verbal), depending upon the nature of the topics. The teaching faculty will determine the choice of teaching pedagogy as per the need of the topics.

Evaluation

Examination Scheme				
Internal Assessment		External Assessment		Total
Theory	Practical	Theory	Practical	
40	-	60 (3 Hrs.)	-	100

Text Book

1. Roger S. Pressman, "Software Engineering: A Practiioner's Appraoch, 6th Edition, MCgraw Hill International edition, 2005

Reference Books

1. Ali Behforooz and Frederick J. Hudson, "Fundamentals of Software Engineering", OUP, 1996
2. Ian Sommerville, "Software Engineering", 9th Edition, Addison-Wesley, 2010, ISBN: 978-0137035151
3. Pankaj Jalote, "An Integrated Approach to Software Engineering", 2nd Edition, Springer, 1997

CSIT Syllabus

Course Title: Software Engineering **Full Marks: 60 + 20 + 20**
Course No: CSC364 **Pass Marks: 24 + 8 + 8**
Nature of the Course: Theory + Lab Credit Hrs: 3
Semester: VI

Course Description:

This course familiarizes students with different concepts of software engineering mainly focusing on software process models, agile development, requirements engineering, models, design, implementation, testing, evolution, and software project management.

Course Objectives:

The main objective of this course is to provide knowledge of different concepts of software engineering so that students will be able to develop high quality software using different software management skills.

Course Contents:

Unit 1: Introduction (2 Hrs.)

Software and its Types; Attributes of Good Software; Software Engineering and its Importance; Fundamental Software Engineering Activities; Difference between Software Engineering and Computer Science; Difference between Software Engineering and System Engineering; Challenges and Cost of Software Engineering; Professional Software Development; Software Engineering Diversity; Internet Software Engineering; Software Engineering Ethics

Unit 2: Software Processes (5 Hrs.)

Software Process; Software Process Models (Waterfall Model; Incremental Development; Integration and Configuration); Software Process Activities (Software Specification, Software Design and Implementation; Software Validation; Software Evolution); Coping with Change (Prototyping, Incremental Delivery); Process Improvement

Unit 3: Agile Software Development (3 Hrs.)

Agile Development; Plan-Driven vs. Agile Development; Agile Methods; Agile Development Techniques; Introduction to Agile Project Management

Unit 4: Requirements Engineering (3 Hrs.)

Concept of User and System Requirements; Functional and Non-Functional Requirements; Requirements Engineering Process; Requirements Elicitation; Requirements Specification; Requirements Validation; Requirements Change

Unit 5: System Modeling **(6 Hrs.)**

Introduction to System Modeling; Context Models; Interaction Models; Structural Models; Behavioral Models; Model-Driven Architecture

Unit 6: Architectural Design **(6 Hrs.)**

Introduction; Architectural Design Decisions; Architectural Views; Architectural Patterns; Application Architectures

Unit 7: Design and Implementation **(5 Hrs.)**

Introduction; Object-Oriented Design using UML; Design Patterns; Implementation Issues; Open-Source Development

Unit 8: Software Testing **(5 Hrs.)**

Introduction; Validation and Verification Testing; Software Inspection; Software Testing Process; Development Testing; Test-Driven Development; Release Testing; User Testing

Unit 9: Software Evolution **(3 Hrs.)**

Evolution Process; Legacy Systems; Software Maintenance

Unit 10: Software Management (7 Hrs.)

Software Project Management; Project Management Activities (Project Planning, Risk Management, People Management, Reporting and Proposal Writing); Project Planning (Software Pricing, Plan-Driven Development, Project Scheduling, Estimation Techniques, COCOMO Cost Modeling); Introduction to Quality Management and Configuration Management

Laboratory / Project Work:

Students should prepare a project report along with software product using different concepts of software engineering. The project can be done in groups with at most four members in each group using any suitable database, programming, interfacing technologies, and project management concepts.

Text Book:

1. Software Engineering, 10th Edition, Ian Sommerville, Pearson Education 2016

References Books:

1. Software Engineering: A Practitioner's Approach, 8th Edition, Roger S. Pressman and Bruce R. Maxim, McGraw-Hill Education 2015
2. Beginning Software Engineering, Rod Stephens, John Wiley & Sons Inc 2015

Table of Contents

1 Chapter INTRODUCTION

2 Chapter SOFTWARE PROCESS AND PROCESS MODEL

3 Chapter AGILE SOFTWARE DEVELOPMENT

4 Chapter SOFTWARE REQUIREMENT & REQUIREMENT ENGINEERING

5 Chapter SYSTEM MODELLING

6 Chapter SOFTWARE ARCHITECTURAL DESIGN

7 Chapter DESIGN AND IMPLEMENTATION

8 Chapter CODING

9 Chapter SOFTWARE TESTING

10 Chapter SOFTWARE EVOLUTION & MAINTENANCE

11 Chapter MANAGING SOFTWARE PROJECT