

COURSE TITLE : **SOFTWARE ENGINEERING**
COURSE CODE : **4072**
COURSE CATEGORY : **A**
PERIODS / WEEK : **4**
PERIODS / SEMESTER : **72**
CREDITS : **4**

TIME SCHEDULE

| MODULE | TOPICS | PERIODS |
|---------------|---|----------------|
| 1 | Software engineering discipline – evolution and impact Software Project Management | 18 |
| 2 | Requirements analysis and specification Software Design | 18 |
| 3 | Coding and Testing Software Reliability and Quality Management | 18 |
| 4 | Computer Aided Software Engineering (CASE) Case study and evaluation | 18 |
| | Total | 72 |

OBJECTIVES

MODULE I

- 1.1 Software Engineering discipline – Evolution and Impact
 - 1.1.1 Explain the relevance of Software Engineering
 - 1.1.2 Discuss the evolution of an art into an Engineering discipline
 - 1.1.3 Discuss Software as a program versus software as product
 - 1.1.4 Explain the emergence of software Engineering
 - 1.1.5 Discuss Early computer programming and high level programming concepts
 - 1.1.6 Discuss Control Flow – Based Design
 - 1.1.7 Discuss Data Structure – Oriented Design
 - 1.1.8 Discuss Data Flow – oriented Design
 - 1.1.9 Discuss Object oriented Design
 - 1.1.10 State changes in Software development practices
 - 1.1.11 State Software Life Cycle Models
 - 1.1.12 Explain Classical waterfall model & Iterative waterfall model
 - 1.1.13 Explain Prototyping Model
 - 1.1.14 Explain Revolutionary Model
 - 1.1.15 Explain Spiral Model
 - 1.1.16 Compare Life cycle models
- 1.2 Software Project Management
 - 1.2.1 Discuss main goal of software project management
 - 1.2.2 Know the job responsibilities of a Software Project Manager
 - 1.2.3 Awareness of necessary skills to be possessed for Project Management

- 1.2.4 Know the concept of Project planning
- 1.2.5 Explain metrics for project size estimation
- 1.2.6 Discuss Lines of code(LOC) and Function Point Metric
- 1.2.7 Discuss Project estimation techniques –empirical , heuristics and analytical
- 1.2.8 Discuss COCOMO – basic , intermediate and complete .
- 1.2.9 Explain Project Scheduling
- 1.2.10 Discuss Work breakdown structure
- 1.2.11 State Activity network & CPM
- 1.2.12 Explain Organization and team structure
- 1.2.13 Discuss the attributes of a good software Engineer
- 1.2.14 Explain Risk Management
- 1.2.15 Discuss Risk identification
- 1.2.16 Discuss Risk assessment
- 1.2.17 Discuss Risk containment
- 1.2.18 Explain Configuration management
- 1.2.19 Discuss necessity of software configuration management
- 1.2.20 Discuss configuration management activities
- 1.2.21 Discuss configuration management tools – SCCS and RCS

MODULE II

- 2.1 Requirements Analysis and specifications
 - 2.1.1 Discuss Requirements Gathering
 - 2.1.2 Discuss Requirements Analysis
 - 2.1.3 Briefly explain software prototyping
 - 2.1.4 State prototyping tools and methods
 - 2.1.5 Explain SRS document
 - 2.1.6 Discuss characteristics of a good SRS document
 - 2.1.7 Explain functional requirements
 - 2.1.8 Understand how to identify functional requirements
 - 2.1.9 Understand how to document Functional Requirements
- 2.2 Software Design
 - 2.2.1 Understand the outcome of a design process
 - 2.2.2 State modules, their relationships and data structure of individual modules.
 - 2.2.3 State design representation techniques structure chart and UML
 - 2.2.4 Compare analysis versus design with respect to goal and scope.
 - 2.2.5 Specify the characteristic of a good software design
 - 2.2.6 Explain Cohesion and their classification
 - 2.2.7 Explain coupling and their classification
 - 2.2.8 Explain layered arrangement of modules
 - 2.2.9 Discuss approaches to software design
 - 2.2.10 Discuss Structured analysis
 - 2.2.11 Detailed study of DFD
 - 2.2.12 Data dictionary and data definition
 - 2.2.13 Discuss Structured Design
 - 2.2.14 Identify the characteristic of a good user interface
 - 2.2.15 Explain types of user interfaces

MODULE III

- 3.1 Coding and Testing
 - 3.1.1 State the objectives of coding
 - 3.1.2 Discuss coding standards and guidelines
 - 3.1.3 Explain code review process
 - 3.1.4 Identify code walkthrough and code inspection
 - 3.1.5 Explain Software documentation
 - 3.1.6 Compare internal documentation and external documentation
 - 3.1.7 State the objective of testing
 - 3.1.8 Identify test activities
 - 3.1.9 Explain unit testing
 - 3.1.10 Discuss Black Box testing
 - 3.1.11 Discuss white box testing
 - 3.1.12 Explain debugging concept
 - 3.1.13 Understand program analysis tools
 - 3.1.14 Explain Integration Testing
 - 3.1.15 Study testing of object oriented programs
 - 3.1.16 Explain system Testing
 - 3.1.17 Discuss general issues associated with testing
- 3.2 Software Reliability and Quality Management
 - 3.2.1 Explain Software reliability
 - 3.2.2 Understand reliability Metrics
 - 3.2.3 Discuss Statistical Testing
 - 3.2.4 Understand the usage software quality
 - 3.2.5 List the factors measuring software quality
 - 3.2.6 Explain Software Quality Management System
 - 3.2.7 Explain Characteristic of software maintenance
 - 3.2.8 Explain Software Reverse Engineering
 - 3.2.9 Explain software Reuse
 - 3.2.10 Identify the prominent items that can be reused
 - 3.2.11 Identify Basic issues in any reuse program

MODULE IV

- 4.1 Computer Aided Software Engineering [CASE]
 - 4.1.1 Define CASE and importance of CASE tools in software industry
 - 4.1.2 Identify primary objective of any CASE tool
 - 4.1.3 Discuss a CASE environment
 - 4.1.4 List benefits of CASE
 - 4.1.5 Discuss CASE support in software life cycle
 - 4.1.6 Understand general characteristics of CASE tools
 - 4.1.7 Discuss the architecture of a typical CASE environment
- 4.2 Case study
 - 4.2.1 Students are to undergo case study to build a simple software product with a powerful database applying all the systematic approaches learned in software life cycle highlighting design and testing
 - 4.2.2 Students are to undergo case study to build simple web based software product applying all the systematic approaches learned in software life cycle

highlighting user interface design

CONTENT OUTLINE

MODULE I

Introduction to Software Engineering - Evolution of an art into an Engineering discipline - Software as a product - Emergence of software engineering – Notable changes in software development practices – software life cycle models.

Software Project Management – Responsibilities of a software Project manager – Project Planning – Metrics for project size estimation – project scheduling – organization and Team structure – Risk Management – Software Configuration Management

MODULE II

Requirement analysis and specification – Requirement gathering – Requirement analysis – Software Requirement Specification (SRS) document – Functional Requirements

Software Design – Design concept – outcome of a design process – characteristics of a good software design – cohesion and coupling – layered arrangement of modules – Approaches to software design – Structured analysis and design methodology – user interface design

MODULE III

Coding – objectives of coding – coding standards and guidelines – code review – software documentation – concept of testing – test activities – unit testing – black box testing – white box testing – debugging – program analysis tools – Integration testing – testing object oriented programs – system testing – General issues associated with testing

Software reliability – reliability metrics – statistical testing – software quality management system – characteristics of software maintenance – Software reverse engineering – software reuse

MODULE IV

CASE – understand CASE – CASE environment – CASE support in software life cycle – characteristics of CASE tools

Case study -1 :- Demonstrate how a software product with secured and powerful database is built using the entire software development life cycle highlighting design and testing.

Case study -2 :- Demonstrate the application of software engineering principles to web based software systems .

TEXT BOOK: Fundamentals of software Engineering by Rajib Mall
PHI publication

REFERENCE BOOKS

1. Software Engineering a practitioners approach – Roger S Pressman
2. Software Engineering – Concepts and applications
Subhajith Datta (Oxford Higher Education)
3. Software Engineering – Shajan Mathew