

# Th-3 SOFTWARE ENGINEERING

(Common to CSE/IT)

|                      |                           |                            |                 |
|----------------------|---------------------------|----------------------------|-----------------|
| <b>Theory</b>        | <b>4 Periods per week</b> | <b>Internal Assessment</b> | <b>20 Marks</b> |
| <b>Total Periods</b> | <b>60 Periods</b>         | <b>End Sem Exam</b>        | <b>80 Marks</b> |
| <b>Examination</b>   | <b>3hours</b>             | <b>Total Marks</b>         | <b>100Marks</b> |

## A. Topic wise distribution of periods

| Sl. No. | Topics                                 | Periods   |
|---------|--|-----------|
| 1       | INTRODUCTION TO SOFTWARE ENGINEERING   | 06        |
| 2       | SOFTWARE PROJECT MANAGEMENT            | 10        |
| 3       | REQUIREMENT ANALYSIS AND SPECIFICATION | 06        |
| 4       | SOFTWARE DESIGN                        | 10        |
| 5       | USER INTERFACE DESIGN                  | 08        |
| 6       | SOFTWARE CODING & TESTING              | 12        |
| 7       | SOFTWARE RELIABILITY                   | 08        |
|         | <b>TOTAL</b>                           | <b>60</b> |

**B. RATIONALE:** Software Engineering technology is now a days largely adopted by most computer based applications to bridge the gap between a human user & the computer. By this multiple media are implemented and used in computer based application to enhance their understanding ability before a common man. This will expose the students to various project building and testing techniques which they will encounter during there professional life as a software engineer or manager.

**C. OBJECTIVE:** After completion of this course the student will be able to:

- Understand the concept of Software Engineering.
- Understand how costs, schedule and quality drive a software project.
- Understand the role of software process and a process model in a project.
- Understand planning and estimation of a software project.
- Understand the role of SRS in a project and how requirements are validated
- Know the key design concepts of software engineering.
- Learn the structured code inspection process.
- Learn how testing is planned and testing done

## D. CORSE CONTENTS:

### 1.0 Introduction to Software Engineering

- 1.1 Program vs. Software product
- 1.2 Emergence of Software Engineering.
- 1.3 Computer Systems Engineering
- 1.4 Software Life Cycle Models
  - 1.4.1 Classical Water fall model
  - 1.4.2 Iterative Water fall model
  - 1.4.3 Prototyping model
  - 1.4.4 Evolutionary model
  - 1.4.5 Spiral model

### 2.0 Software Project Management

- 2.1 Responsibility of Project Manager
- 2.2 Project Planning
- 2.3 Metrics for Project size estimation(LOC and FP)

- 2.4 Project Estimation Techniques
- 2.5 COCOMO Models, Basic, Intermediate and complete
- 2.6 Scheduling
- 2.7 Organization and Team structure
- 2.8 Staffing
- 2.9 Risk Management
- 2.10 Configuration Management

### **3.0 Requirement Analysis and specification**

- 3.1 Requirements gathering and analysis
- 3.2 Software Requirements Specification
  - 3.2.1 Contents of SRS
  - 3.2.2 Characteristics of Good SRS
  - 3.2.3 Organization of SRS
  - 3.2.4 Techniques for representing complexing logic

### **4.0 Software Design**

- 4.1 What is a Good S/W design
- 4.2 Cohesion and coupling
- 4.3 Neat arrangement
- 4.4 S/W Design approaches
- 4.5 Structured analysis
- 4.6 Data Flow Diagrams
- 4.7 Symbols used in DFD
- 4.8 Designing DFD
- 4.9 Developing DFD model of a system
- 4.10 Shortcomings of DFD
- 4.11 Structured design
- 4.12 Principles of transformation of DFD to Structure Chart
- 4.13 Transform analysis and Transaction Analysis
- 4.14 Design Review

### **5.0 User Interface Design**

- 5.1 Characteristics of Good Interface
- 5.2 Basic concepts of UID
- 5.3 Types of User interfaces
- 5.4 Components based GUI development

### **6.0 Software Coding & Testing**

- 6.1 Coding
- 6.2 Code Review
  - 6.2.1 Code walk through
  - 6.2.2 Code inspections and software Documentation
- 6.3 Testing
- 6.4 Unit testing
- 6.5 Black Box Testing
- 6.6 Equivalence class partitioning and boundary value analysis
- 6.7 White Box Testing
- 6.8 Different White Box methodologies statement coverage branch coverage, condition coverage, path coverage, cyclomatic complexity data flow based testing and mutation testing
- 6.9 Debugging approaches
- 6.10 Debugging guidelines
- 6.11 Integration Testing

- 6.12 Phased and incremental integration testing
- 6.13 System testing alphas beta and acceptance testing
- 6.14 Performance Testing, Error seeding
- 6.15 General issues associated with testing

## 7.0 Software Reliability

- 7.1 Software Reliability
- 7.2 Different reliability metrics
- 7.3 Reliability growth modeling
- 7.4 Software quality
- 7.5 Software Quality Management System

### Coverage of Syllabus upto Internal Exams (I.A.) Chapter 1,2,3,4

#### BOOKS Recommended:-

| Sl.No | Name of Authors | Title of the Book                             | Name of the publisher   |
|-------|-----------------|---|-------------------------|
| 01    | Rajib Mall      | Fundamentals of Software Engineering          | PHI                     |
| 02    | Deepak Jain     | Software Engineering: Principles and Practice | Oxford university press |
| 03    | Jawadekar       | Software Engineering: A Primer                | TMH                     |