

# CONTENTS.....

## **1. INTRODUCTION TO SOFTWARE ENGINEERING**

- ☐ Introduction (what is software Engineering)
- ☐ Software Applications
- ☐ Evolution of Software
- ☐ Software crisis : Problem & Causes
- ☐ Software Components

## **2. SOFTWARE DEVELOPMENT LIFE-CYCLE**

- ☐ Introduction
- ☐ The system development life cycle
  - ▼▼ Recognition of Need/what is the Problem?
  - ▼▼ Feasibility Study
  - ▼▼ Analysis
  - ▼▼ Design
  - ▼▼ Implementation
  - ▼▼ Post implementation Review
  - ▼▼ Maintenance
- ☐ Software Development Process Models
  - ▼▼ Waterfall model
  - ▼▼ Prototyping Model
  - ▼▼ Iterative Enhancement model
  - ▼▼ Spiral model
  - ▼▼ RAD Model

- ☐ Software metrics and models
  - ▼▼ Measure, metrics and indicators
  - ▼▼ Process metrics and software process improvement.
  - ▼▼ Project metrics

### **3. PROJECT PLANNING AND SCHEDULING**

- ☐ Project Planning
- ☐ Project Monitoring & Control
- ☐ Cost Estimation
- ☐ Project scheduling
- ☐ Software Quality
- ☐ Quality Assurance
- ☐ Project Monitoring Plans
- ☐ Risk Management
- ☐ Project Staffing
- ☐ Software Configuration Management

### **4. SYSTEM DESIGN**

- ☐ What is design?
  - ▼▼ Design Objective
  - ▼▼ Design principles
- ☐ Different view of design
- ☐ Decomposition and modularity
  - ▼▼ Object oriented design
  - ▼▼ modular decomposition
  - ▼▼ Event-oriented decomposition
  - ▼▼ Data-oriented decomposition.
- ☐ Characteristics of good design
  - ▼▼ Component independence
    - ▼▼▼ Cohesion
    - ▼▼▼ Coupling
  - ▼▼ Exception identification & handling
  - ▼▼ Fault detection, correction & tolerance
- ☐ Techniques for improving design
- ☐ Structured analysis
  - ▼▼ Goals of structured analysis
  - ▼▼ Need for structured analysis
  - ▼▼ Data flow diagram
  - ▼▼ Data dictionary

- ▼▼ Structured English
- ▼▼ Decision table
- ▼▼ Decision tree
- ☐ Summary
- ☐ Question

## 5. OBJECT ORIENTED DESIGN

- ☐ What is object Orientation?
- ☐ Object oriented design
- ☐ Software design Concepts
- ☐ Objects and classes
- ☐ Object Oriented design process
  - ▼▼ OO System design
  - ▼▼ OO Program design
- ☐ Benefits of Object oriented design over structural programming.

## 6. OBJECT ORIENTED DESIGN

- ☐ Building the Structure
- ☐ Relationships
- ☐ Inheritance Relationship
- ☐ Why Generalize?
- ☐ Composition Relationship
- ☐ Utilization Relationship
- ☐ Instantiation Relationship
- ☐☐ **Object Relationship Diagrams**
- ☐ State the significance of an Object Relationship Diagram
- ☐ Explain Cardinality
- ☐ Draw an Object Relationship Diagram representing:
  - ▼▼ Classes
  - ▼▼ Relationships
  - ▼▼ Cardinality
- ☐☐ **Object Flow Diagrams Network Diagrams and State Transition Diagrams**
- ☐ Explain the terms
  - ▼▼ Activity
  - ▼▼ Product
- ☐ Draw an OFD representing activities and products

- ☐ Appreciate the need for a Network Diagram
- ☐ Draw a Network Diagram
- ☐ State the significance of State Transition Diagram
- ☐ Identify the states of an object.
- ☐ Identify the events that change the state of an object.
- ☐ Draw a State Transition Diagram.

## **7. DEPENDABILITY AND SYSTEM SPECIFICATION**

- ☐ Dependability
  - ▼▼ Dimensions of dependability
    - ▼▼▼ Availability
    - ▼▼▼ Reliability
    - ▼▼▼ Safety
    - ▼▼▼ Security
  - ▼▼ Approaches to improve dependability
- ☐ Software reliability specifications
  - ▼▼ Reliability metrics
  - ▼▼ Non functional reliability requirements
  - ▼▼ Steps in reliability specifications
- ☐ Safety & its specification
- ☐ Security & its specification

## **8. FAULT MINIMIZATION AND TOLERANCE**

- ☐ Introduction
- ☐ Fault minimization
  - ▼▼ Error avoidance
  - ▼▼ Information hiding
  - ▼▼ Reliable software processes
- ☐ Fault tolerance

## **9. VERIFICATION & VALIDATION**

- ☐ Introduction
- ☐ Verification & Validation Planning
- ☐ Software Inspection
  - ▼▼ Program inspection
- ☐ Automated state analysis

## **10. CODING**

- ☐ Introduction
- ☐ Top-down and Bottom-up approaches
- ☐ Structured programming.

- ☐ Programming style
- ☐ Internal documentation
- ☐ Low of Demeter for OO programs
- ☐ Verification & Validation
- ☐ Code reading
- ☐ Static analysis
- ☐ Symbolic execution
- ☐ Path conditions
- ☐ Metrics for coding
  - ▼▼ Halstead method
  - ▼▼ Line Variable
- ☐ Monitoring and control

## 11. TESTING

- ☐ Testing Issues
- ☐ Testing Objectives
- ☐ Testing Principle
- ☐ Terms related to testing functions
- ☐ Levels of Testing
  - ▼▼ Unit testing
  - ▼▼ Integration testing
    - ▼▼▼ Top-down integration
    - ▼▼▼ Bottom-up integration
    - ▼▼▼ Big-bang integration
    - ▼▼▼ Sandwich integration
    - ▼▼▼ Regression testing
    - ▼▼▼ Smoke testing
    - ▼▼▼ Stress testing
- ☐ Test case and test criteria and test Oracle
- ☐ Black-box testing or functional testing
- ☐ White-box testing or structural testing
  - ▼▼ Control flow criteria
  - ▼▼ Data flow criteria
  - ▼▼ loop testing
  - ▼▼ Test case generation
  - ▼▼ Mutation testing
- ☐ Test Plan
- ☐ Alpha and Beta testing

## **12. DEVELOPING AND MAINTAINING THE SYSTEM**

- ☐ Training
  - ▼▼ Training Aids
- ☐ Documentation
- ☐ Nature of Maintenance
- ☐ Maintenance Problems
- ☐ Maintenance techniques & tools

## **13. SOFTWARE RE-ENGINEERING**

- ☐ Testing Issues
- ☐ Introduction
- ☐ Source code translation
- ☐ Reverse engineering
- ☐ Program structure improvement
- ☐ Program modularisation
- ☐ Data re-engineering

**APPENDIX: CASE**

**SOLVED QUESTIONS**

**GLOSSARY**

## About the Authors

Ritesh Rastogi PGDM (IS), M.Sc (CS), MCA has been engaged in field of teaching since last Six years. Through his careers of Six year he had taught various subject in computers and Management he has vast knowledge of this subject and has been teaching this subject for past three years. He has also written other books and is devoted his career towards teaching and believes in providing knowledge to the students.

Vandana Rastogi. MA (Maths) MCA has been engaged in teaching and has a vast experience and knowledge. She has been teaching various subject in computers. She has also written other books in computers and has devoted her career towards teaching.

## About the Book

This book of Software Engineering has been designed in an easy language which help students to understand. Basic concept of Software Engineering has been explained in detail. More Over live examples, case studies and solved exercises has also been given so that student can get hand on practise. Various books, journals have been consulted in order to make this book more effective. It contains to the point material for benefit of the students.

Furthermore we would welcome suggestion from our readers and respected teachers so that this book can be improved.

Authors.