



College of Natural and Computational Sciences
Department of Computer Science

Course title: Software Engineering

Credit hours: 4

Course code: CoSc3061

Course Description

This course provides an introduction to the problems of software development and maintenance and the processes and methods used to address them. All phases in the software development life cycle will be introduced. The course also deals with software project planning, cost estimation, tracking and control, staffing, risk management, and software configuration plan.

Covers object oriented (O-O) concepts, tools, development life cycle, problem solving, modeling, analysis, and design, while utilizing UML (Unified Modeling Language) for O-O modeling. UML has become the standard notation for modeling O-O systems and is being embraced by major software developers like Microsoft and Oracle.

Course objectives

On completion of the course successfully, students will be able to:

- Understand the basic principles of Software Engineering
- Write requirements specification documents
- Design a system, component, or process to meet desired needs.
- Verify and validate a Software system practice with effective communication skill.
- Describe in detail the theory, concepts and methods pertaining to the Unified Modeling Language (UML).
- Create requirements using use case modeling concepts.
- Demonstrate conceptual and technical skills in the analysis, design and implementation of a software system using Object Oriented Concepts.
- Employ tools and techniques for Object Oriented Software Engineering.
- Demonstrate an ability to adapt and solve problems in software development activities from specification to testing individually and as part of a team.

Course outline

Chapter 1: Introduction

- 1.1. Two Orthogonal view of software.
- 1.2. Software development process models
 - 1.2.1. Software Process
 - 1.2.2. Software life cycle and process models
 - 1.2.3. Process assessment models
 - 1.2.4. Software process metrics

- 1.3. Object oriented system development methodology.
 - 1.3.1. Why an object oriented
 - 1.3.2. Overview of the unified approach.
 - 1.3.3. An object oriented philosophy
 - 1.3.4. Basic concepts of an object
 - 1.3.5. Attributes of an object, its state and properties.

Chapter 2: Software Project management

- 2.1. Responsibility of Software Project Managers
- 2.2. Project Planning
- 2.3. The organization of SPMP document
- 2.4. Project Size Estimation Metrics
- 2.5. Project Estimation Techniques
- 2.6. Scheduling, Organization and Team Structures
- 2.7. Staffing
- 2.8. Risk Management
- 2.9. Quality Assurance
- 2.10. Project Monitoring Plans

Chapter 3: Unified Modeling Language (UML)

- 3.1. Where Can the UML Be Used
- 3.2. Building Blocks of the UML.
- 3.3. Relationships in the UML
- 3.4. Diagrams in the UML.
 - 3.4.1. Use Case Diagrams
 - 3.4.2. Class Diagrams
 - 3.4.3. Sequence diagrams
 - 3.4.4. State chart diagrams
 - 3.4.5. Activity diagrams
 - 3.4.6. Component diagram
 - 3.4.7. Deployment diagram
 - 3.4.8. Diagram extensions

Chapter 4: Requirements Elicitation

- 4.1. An overview of requirements elicitation.
- 4.2. Requirements elicitation concepts
 - 4.2.1. Functional requirements
 - 4.2.2. Nonfunctional and pseudo requirements
 - 4.2.3. Levels of description
 - 4.2.4. Correctness, completeness, consistency, clarity, and realism
 - 4.2.5. Verifiability and traceability
- 4.3. Requirements elicitation activities.
 - 4.3.1. Identifying actors
 - 4.3.2. Identifying scenarios
 - 4.3.3. Identifying use cases
 - 4.3.4. Refining use cases
 - 4.3.5. Identifying relationships among actors and use cases
 - 4.3.6. Identifying initial analysis objects
 - 4.3.7. Identifying nonfunctional requirements
- 4.4. Managing requirements elicitation
 - 4.4.1. Eliciting information from users:
 - 4.4.2. Validating requirements: Usability testing
 - 4.4.3. Documenting requirements elicitation

Chapter 5: Software Project management

- 5.1. Responsibility of Software Project Managers
- 5.2. Project Planning
- 5.3. The organization of SPMP document
- 5.4. Project Size Estimation Metrics
- 5.5. Project Estimation Techniques
- 5.6. Scheduling, Organization and Team Structures
- 5.7. Staffing
- 5.8. Risk Management
- 5.9. Quality Assurance
- 5.10. Project Monitoring Plans

Chapter 6: Object Oriented Requirement Analysis

- 5.1. Analysis Concepts
 - 5.1.1. Entity, Boundary, and Control Objects
 - 5.1.2. Association Multiplicity Revisited
 - 5.1.3. Qualified Associations
 - 5.1.4. Generalization
- 5.2. Analysis Activities: From Use Cases to Objects
 - 5.2.1. Identifying Entity Objects
 - 5.2.2. Identifying Boundary Objects
 - 5.2.3. Identifying Control Objects
 - 5.2.4. Modeling Interactions between Objects: Sequence Diagrams
 - 5.2.5. Identifying Associations.
 - 5.2.6. Identifying Attributes
 - 5.2.7. Reviewing the Analysis Model

Chapter 6: Object Oriented System Design

- 6.1. An overview of system design.
 - 6.1.1. System design concepts.
 - 6.1.2. System design activities: From objects to subsystems
 - 6.1.3. Documenting system design
 - 6.1.4. An overview of object design
 - 6.1.5. Object design concepts
 - 6.1.6. Object design activities
 - 6.1.7. Managing object design
 - 6.1.8. Documenting object design

Chapter 7: Software Quality Assurance

- 7.1. An overview of testing
- 7.2. Testing concepts
- 7.3. Testing activities
- 7.4. Managing testing
- 7.5. Impact of object oriented testing
- 7.6. Types of Testing

Text Book

- 1. Brahmin, Ali (1999), Object oriented System development, McGraw Hill, USA.

References

- 1. **Martina Seidl, Marion Scholz, Christian Huemer, Gerti Kappel. UML @ Classroom: An Introduction to Object-Oriented Modeling. 2012. Springer International Publishing AG.**
- 2. Scott, Kendall (2004) Fast Track UML 2.0 Apress USA

3. Booch, Grady Rumbaugh, James Jacobson, Ivar (2005) The Unified Modeling Language User Guide second edition Addison Wesley Professional USA

Evaluation Schemes

➤ Quizzes, assignments and tests	30%
➤ Projects	20%
➤ Final Exam	50%