



COURSE TITLE	:Software Engineering
COURSE PREREQUISITE	:Database Concepts
COURSE DURATION	:16 weeks (2 hours lecture; 3 hours laboratory per week)
COURSE METHODOLOGY	:Combination of lectures, laboratory exercises and software project

Course Description

This course introduces students to the basic concepts, principles and dynamics of software engineering. It involves the study of methodologies and techniques, and the construction of models at each major software development phase, namely, requirements analysis, design, testing and implementation. At the end of the course, the students should be able to submit a software project complete with technical documents.

Course Outline

Week	Topics
1	<p>Introduction to Software Engineering</p> <ul style="list-style-type: none"> • Software Engineering- A Layered View • Quality within the Development Effort • Software Quality Assurance and Techniques • Software Process • Understanding Systems • Understanding People in the Development Effort • Documentation in the Development Effort
2	<p>Object-oriented Software Engineering</p> <ul style="list-style-type: none"> • Object-oriented Concepts • Object-oriented Process Model • Object-oriented Analysis and Design • Unified Modeling Language (UML) <p>Requirements Engineering</p> <ul style="list-style-type: none"> • Requirements Engineering Tasks <ul style="list-style-type: none"> • Inception • Elicitation • Elaboration • Negotiation • Validation

3	Requirements Engineering <ul style="list-style-type: none"> • Requirements Analysis and Model <ul style="list-style-type: none"> • Scenario-modeling • The Requirements Model <ul style="list-style-type: none"> • Use-case Model • Supplementary Specifications • Glossary
4	Requirements Engineering <ul style="list-style-type: none"> • Requirements Specifications <ul style="list-style-type: none"> • Use-case Analysis Technique • The Analysis Model <ul style="list-style-type: none"> • Object Model • Behavioral Model
5	Requirements Engineering <ul style="list-style-type: none"> • Requirements Traceability Matrix • Requirements Metrics
6	Design Engineering <ul style="list-style-type: none"> • Design Engineering Concepts <ul style="list-style-type: none"> • Quality Guidelines for Design • Abstraction • Modularity • Refinement • Refactoring • The Design Model <ul style="list-style-type: none"> • Software Architectural Design • Design Patterns
7	Design Engineering <ul style="list-style-type: none"> • The Design Model <ul style="list-style-type: none"> • Data Design <ul style="list-style-type: none"> • Persistence and Database Design
8	Design Engineering <ul style="list-style-type: none"> • The Design Model <ul style="list-style-type: none"> • Interface Design Elements <ul style="list-style-type: none"> • Report Layout • Form Layout • Dialogue Design and Screen Design
9	Design Engineering <ul style="list-style-type: none"> • The Design Model <ul style="list-style-type: none"> • Component-level Design Elements <ul style="list-style-type: none"> • Component-level Design Guidelines • Deployment-level Design Elements • Map Design Deliverables with Requirements Traceability Matrix • Measuring Design

10	Implementation <ul style="list-style-type: none"> • Programming Standards and Procedures • Programming Guidelines • Implementing Packages • Implementing Java Database Connectivity (JDBC) • Implementing Graphical User Interface • Map Implementation Deliverables with Requirements Traceability Matrix • Implementation Metrics
11	Software Testing <ul style="list-style-type: none"> • Software Testing Concepts • Software Testing Design Concepts • Testing the Programs <ul style="list-style-type: none"> • Unit Testing • Integration Testing • Test-driven Development
12	Software Testing <ul style="list-style-type: none"> • Testing the System <ul style="list-style-type: none"> • Types of System Testing <ul style="list-style-type: none"> • Function Testing • Performance Testing • Reliability, Availability and Maintainability • Acceptance Testing • Installation Testing • Generating System Test Cases • Map Test Cases with Requirements Traceability Matrix • Test Metrics
13	Basic Software Project Management <ul style="list-style-type: none"> • Software Project Management Concepts • Project Identification and Definition • Project Organization • Project Scheduling and Work Breakdown • Resource Allocation
14	Basic Software Project Management <ul style="list-style-type: none"> • Software Metrics and Estimation • Software Project Plan • Risk Management • Software Configuration Management
15	Software Development Tools <ul style="list-style-type: none"> • Collaborative Software • Version Control Software • Bugs, Issues and Fault Management Software • Diagramming and Code Generation Tools • Software Quality Assurance Tools
16	Software Project Presentation

Software Project Deliverables

The list below contains the deliverables of the software project. The *Projected Delivery Date* specifies when deadlines can be set.

- I. Requirements Model (*Projected Delivery Date: After Week 3*)
 - A.Actor Description
 - B.Use-case Model
 - i. Use-case Diagrams
 - ii. Use-case Specifications
 - C.Supplementary Specifications
 - D.Glossary
- II.Analysis Model (*Projected Delivery Date: After Week 5*)
 - A.Object Model
 - i. Analysis Classes
 - B.Behavioral Model
 - i. Sequence Diagrams
 - ii. Collaboration Diagrams
 - C.Requirements and Analysis Metrics
- III.Design Model (*Projected Delivery Date: After Week 9*)
 - A.Software Architecture
 - B.Interface Design
 - i. Dialogue Design using State chart Diagrams
 - ii. Screen Design
 - iii.Software Screen Prototypes
 - iv.Report and Forms Designs
 - C.Component/Controller Design
 - i. Behavior of components using State chart Diagrams
 - D.Deployment Model
 - E.Design Metrics
- IV.Software Testing (*Projected Delivery Date: After Week 12*)
 - A.Test Plan
 - B.Test Cases
 - C.Test Metrics
- V.Implementation (*Delivery Date: Week 16*)
 - A.Source Codes and Executable Codes
 - B.Schema of the Database
 - C.Backup of the Database
 - i. Blank Database
 - ii.Clean Database (ready for production where pre-defined codes are inserted into tables)
 - iii.Scenario Databases
 - D.Installers
 - i. Installer of software project done
 - ii. Installer of technology used such as database, web server etc.
 - iii.Installation and Configuration Procedures
- VI.Requirements Traceability Matrix (*Delivery Date: Week 16*)
- VII.Project Paper (*Delivery Date: Week 16*)
 - A.Background of the Study
 - B.Statement of the Problem
 - C.Scope and Limitations
 - D.Project Objectives
 - E.Project Results
 - F. Conclusion
 - G.Recommendations

Requirements

To follow.