

CS 4531: Software Engineering (4)**Catalog Description:**

Formal methods of software design and development. Recognition of conditions for production of high quality software. Organization and management of software development projects. Introduction to design methodologies.

Textbook:

Timothy Lethbridge and Robert Laganier, *Object-Oriented Software Engineering: Practical Software Development Using UML and Java*. 2nd Ed., McGraw-Hill, 2005.

Cay Horstman and David Geary, *Core JavaServer Faces*, 2nd Ed., Prentice Hall, 2007.

Course Goals:

This course introduces students to the practice of software engineering, or “the process of solving customers’ problems by the systematic development and evolution of large, high-quality software systems within cost, time, and other constraints.” The objective is to give students exposure to software design tools, software process models, an appreciation for the role of the user, and the implementation of complex designs.

Prerequisites by Course & Topic

CS 2511: Software Analysis and Design – Object-oriented programming and design

MA 3355: Discrete Math – Formal logic description language

Major Topics Covered in the Course

- User-centered design
- Web Engineering
- Developing Requirements
- Static Class Modeling and Design Patterns
- Dynamic Modeling of Behavior
- Software Architecture

Class/Laboratory Schedule: Lecture: 3 hours per week, Laboratory: 1

Course Outcomes

1. Understand factors that affect software quality.
 - a. Identify and evaluate external software qualities such as reliability, efficiency, testability, and maintainability.
 - b. Identify and evaluate internal software qualities such as simplicity, coupling, and cohesion.
 - c. Be able to apply knowledge of software qualities to actual software projects.
2. Understand and implement an effective software development process.
 - a. Be able to apply principles that guide choices regarding software development process selection.
 - b. Be able to perform domain analysis.
 - c. Be able to write a requirements document.
 - d. Be able to statically model software with class diagrams incorporating design patterns.
 - e. Be able to dynamically model software with interaction, state, and activity diagrams.
 - f. Be able to implement design models in a chosen programming language.
 - g. Understand the role of management.
 - h. Be able to apply knowledge of software development process to actual software projects.
3. Understand the role of the user in software development.
 - a. Be able to produced user-centered designs.
 - b. Be able to perform use case analyses.
 - c. Be able to evaluate and build effective graphical user interfaces

4. Be able to work successfully on a team.
 - a. Meet with fellow members of a class team on a regular basis to choose leadership, delegate responsibility, and see a large software development project through to completion.
 - b. Communicate regularly with team members.

Assessment Plan for Course:

This course is assessed every third year by the instructor and a course assessment document covering all of the course outcomes and their effect on the program outcomes is prepared.

Estimate CSAB Category Content

	CORE	ADVANCED		CORE	ADVANCED
Data Structures			Computer Organization and Architecture		1
Algorithms			Concept of Programming Languages		
Software Design		3			

Coordinator/Prepared by: T. Colburn