

# Department of Computer Engineering and Computer Science

## CECS 590 Course Outline

### Requirements Engineering

#### DESCRIPTION

This course aims to equip students to develop techniques of software-intensive systems through successful requirements analysis techniques and requirements engineering. Students learn systematic process of developing requirements through cooperative problem analysis, representation, and validation. Lecture 2 hours. Semester long team project plus final exam. Letter grade only (A-F).

#### I. PREREQUISITE TOPICS

CECS 343 or CECS 543 or other basic knowledge about the principles of software engineering and the software lifecycle.

#### II. COURSE TOPICS

This course exposes students to the problem of determining and specifying *what* a proposed software system should do, *why* and for *whom* the system is needed, not *how* the system should do it, which is the topic of downstream software engineering activities such as design and coding. There are some nontechnical aspects of the course, with respect to communication and negotiation with multiple stakeholders. Most of the course covers technical approaches to the requirements problem, such as techniques for eliciting stakeholder goals and requirements, notations and models for documenting and specifying requirements, strategies for negotiating requirements, and techniques for analyzing documented requirements.

In detail, the course covers:

- WHY do we need Requirements Engineering?
- Principles: Definitions, process, roles
- System Models: Decomposition and abstraction, system views
- Frameworks: What reference structures can I use for requirements?
- Business Case Analysis: Why are we building this system?
- Stakeholders: Who are the people to talk to about requirements?
- Goals and Constraints: What are the major objectives for the system?
- System Vision: What exactly do we want to achieve?
- Domain Models: What are the surrounding systems ours interacts with?
- Usage Models: How will the system interact with the user?
- Software quality models: How to determine the quality characteristics?
- Quality requirements: How to specify which qualities need to be met?
- Quality assurance: How to ensure that RE is done in a good way?
- Change management: How to evolve requirements?

#### III. COURSE OBJECTIVES

- Overall: A general introduction to requirements engineering.
- Specific: a knowledge of and an ability to apply:
  - Stakeholder analysis
  - Goal analysis
  - Creating a system vision
  - Developing a domain model
  - Developing a usage model (UML use cases)

- Eliciting and specifying quality requirements
- Quality assurance techniques
- Requirements management

*Sample assignments:*

- Eliciting and documenting the stakeholders for a software system.
- Developing a use case in UML.
- Performing a review of quality requirements.

## **V. METHODS FOR ASSESSING STUDENT LEARNING**

- A semester-long requirements engineering project, composed of individual, written assignments (to practice and demonstrate the skills from the course objectives above).
- A final examination.

## **VI. FURTHER READING**

Please note that these books are not a script for the class, but just additional information.

- Karl Wieggers and Joy Beatty: Software Requirements
- Axel van Lamsweerde: Requirements Engineering

Prepared by Birgit Penzenstadler