CECS 590 Requirements Engineering
Course Syllabus, Spring 2015

Professor: Dr. Birgit Penzenstadler
Office: 536 ECS
Email: birgit.penzenstadler@csulb.edu
Office Hours: 4:15 – 5:15 Monday, Wednesday

DESCRIPTION

Requirements Engineering lays the crucial foundation for successful software development. This course aims at equipping students with requirements engineering techniques for software-intensive systems. Students will learn a systematic approach to develop requirements through cooperative problem analysis, representation, and validation.
Lecture: 2 hours. Assessment: Team project plus exams. 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, and techniques for analyzing requirements.
In detail, the course covers:

- WHY do we need Requirements Engineering?
- Principles: Definitions, process, roles, problem/solution view, artifact orientation
- 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?
- Process requirements: How to specify constraints for development?
- System specification: How to hand over to design?
- 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. COURSE REQUIREMENTS AND ASSESSMENT

Semester Team Project: The students will undertake a semester-long requirements engineering project, composed of individual, written assignments (to practice and demonstrate the skills from the course objectives above). Students may form teams of no more than three members. All members must participate equally, although not necessarily doing the same jobs.
No late work accepted. Deadlines will be *strictly adhered to*. The project will require a number of deliverables. Almost every semester there is at least one team, and often more, that does a very poor job. In most cases a team member who has contributed little or nothing causes this. You have a responsibility to your team to contribute time and effort to the project. If I determine that you have not contributed, you will receive zero points for the project. The project is composed by 5 assignments:

- Stakeholder model (10% of the grade)
- Goal model (10%)
- System vision (10%)
- Usage model (10%)
- Quality requirements (10%)

Written Individual Assignments: There are two individual written assignments (short papers), a literature review and a research paper. Each will be 25% of your grade. These assignments plus the team assignments sum up to 100%.

Academic Honesty: Students are to do their own assignments. Cases of copying, cheating, and plagiarism of assignments and/or tests, and any other violations, will be pursued to the maximum extent permitted by the University, which can include expulsion from the University. This applies equally to students who intentionally assist other students in academic dishonesty. When you go on a job interview, you will very likely be asked to demonstrate your knowledge of your field. How will you do that if you cheated your way through?

Reasonable Accommodation: Individuals with disabilities who need assistance or modification to the University's programs and/or activities should inform the person(s) responsible for these programs and/or activities immediately upon knowing that such modification is necessary. Individuals registered with the California Department of Rehabilitation may be eligible for assistance through that agency. Students may be eligible for assistance through the Office of Disabled Student Services, located in Brotman Hall 270, telephone (562) 985-5401. For evaluation and service, contact that office directly. If the modification or accommodation provided is inappropriate or insufficient, you may seek the assistance of the Office of Equity and Diversity, located in University Student Union 301, telephone (562) 985-8256. If a reasonable accommodation has been requested but was not provided, the individual may access the complaint resolution process.

VI. FURTHER READING

Please note that these books are not a script for the class, but just additional reading suggestions:
Karl Wiegers and Joy Beatty: "Software Requirements"
Axel van Lamsweerde: "Requirements Engineering"