**CECS 542 Requirements Engineering**
**Course Syllabus, Spring 2017**

**Professor**: Dr. Birgit Penzenstadler
**Class meetings**: Lecture: Mon/Wed 5:30pm-6:20pm in ECS-302
Lab: Mon/Wed 6:30-7:45pm in ECS-403
**Office**: 536 ECS
**Email:** birgit.penzenstadler@csulb.edu
**Office Hours**: 4:00 – 5:00 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.
Lab: 4 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?
- 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 Exams**: There are two individual written exams. Each will be 25% of your grade. These exams plus the team assignments sum up to 100%.

**Week Plan**

| | Date | Lecture | Lab Assignments |
|---|---|---|---|
| 1 | 23-Jan | Introduction & RE main terms | *Due dates Monday 9 am* |
| | 25-Jan | EARS – Easy Approach to Requirements Syntax | Review HFOSS projects for interest: http://openmrs.org/ Practice EARS for their requirements |
| 2 | 30-Jan | Principles: Definitions, process, roles, problem/solution view, artifact orientation | Identify equivalents or instances of these terms and concepts in HFOSS project; Research the NaPiRE study |
| | 1-Feb | Frameworks in RE | Research the ISO RE standard 29148 |
| 3 | 6-Feb | AMDiRE | Take a standard requirements specification and sort information into AMDiRE model |
| | 8-Feb | Business Case Analysis | Reengineer a business case (pitch) for a system |
| 4 | 13-Feb | Stakeholders | Reengineer list of stakeholders in HFOSS project |
| | 15-Feb | Stakeholders | Make stakeholder model (*deliverable*) |
| 5 | 20-Feb | Goals and Constraints | Reengineer list of goals in HFOSS project |
| | 22-Feb | Goals and Constraints | Make goal model (*deliverable*) |
| 6 | 27-Feb | System Vision | Research system vision, read “rich picture” paper & tutorial |
| | 1-Mar | System Vision | Reengineer system vision (*deliverable*) |

| | | | |
|---|---|---|---|
| 7 | 6-Mar | Domain Models | Research different types of domain models |
| | 8-Mar | Review - Q & A | Time for more questions |
| 8 | 13-Mar | Midterm | - |
| | 15-Mar | *Feedback* | Time for questions |
| 9 | 20-Mar | Usage Models | Reengineer use cases (draft) |
| | 22-Mar | Usage Models | Refine/rework use cases (*deliverable*) |
| -- | 27-Mar | Spring break | |
| | 29-Mar | Spring break | |
| 10 | 3-Apr | Scaling RE & Refining requirements | Practice requirements refinement on project |
| | 5-Apr | Software tools in RE | Continue requirements refinement |
| 11 | 10-Apr | Quality requirements | Reengineer quality requirements |
| | 12-Apr | Quality requirements | Finalize quality requirements (deliverable) |
| 12 | 17-Apr | Quality assurance | Perform peer review (Understanding Design and Checking Compliance) |
| | 19-Apr | Quality assurance | Receive peer review feedback |
| 13 | 24-Apr | Requirements management and change management | Bug tracker activity |
| | 26-Apr | Requirements management and change management | Bug tracker activity |
| 14 | 1-May | Advanced topics in RE | Research agendas in RE |
| | 3-May | Final presentations | - |
| 15 | 8-May | Final presentations | - |
| | 10-May | Review - Q & A | Time for more questions |
| Finals week | tbd | Final exam | - |

**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"

**Academic Honesty:**

***Cheating* and *plagiarism*** will not be tolerated in this course. Any individual caught cheating on quizzes, homework, lab projects, or the final exam will be punished to the full extent allowed under University regulations. Plagiarism on papers or assignments is not acceptable and work that is plagiarized will not receive credit. Plagiarism is considered cheating. *Note:* any time another person's work is used without giving them proper credit, it is considered plagiarism and cheating. It is also considered plagiarism if you try to reuse work from other courses for the deliverables in this course.
*At a minimum,* any student caught cheating will receive no credit for the work concerned, and will receive a reduction of one letter grade from their final course grade. The official CSULB Policy on Cheating and Plagiarism can be found here:
**http://web.csulb.edu/divisions/aa/catalog/current/academic_information/cheating_plagiarism.html**

**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.

**COE Tutoring Center Announcement**

Take advantage of free peer tutoring to keep up your grades in the most challenging classes. Tutoring is available for undergraduate engineering students in departmental courses for Electrical Engineering, Mechanical & Aerospace Engineering, Civil Engineering, Computer Engineering/Science, and Chemical Engineering.

Tutoring is on a first-come-first-serve, walk-in basis. Tutors are available Monday-Thursday in the Fall and Spring terms. All tutoring sessions take place in the Engineering Student Success Center (ESSC) in EN2, Room 300 between the hours of 9:00 a.m. – 6:00 p.m.

Visit the website for detailed tutoring schedules:

**http://web.csulb.edu/colleges/coe/views/essc/academic_success/engineering_tutor.shtml#asp_ETP**