

**Course Syllabus**  
**Software Engineering I: Requirements Engineering**  
**CS4310 (12002)**

**Instructor:** Elsa Tai Ramirez  
**Office:** CRBL 404  
**E-mail:** [wyetai@miners.utep.edu](mailto:wyetai@miners.utep.edu)  
**Office hours:** Tuesday: 9:00 to 10:00 am; by appointment  
**Class Time:** 7:30 to 8:50 am TR, CCSB 1.0202/SE Lab in CCSB 1.0510  
**Prerequisites:** Senior Standing in CS

**TA:** Jose Cabrera  
**Office:** CCSB 1.0706  
**E-mail:** [jacabreramaynez@miners.utep.edu](mailto:jacabreramaynez@miners.utep.edu)  
**Office hours:** TBA

**Communication Network.** <https://www.ibm.com/cloud-computing/social/us/en/>

**Communication.** Communication of course announcements, assignments, and resources will be done through IBM Connections Cloud. You will be receiving an invitation as soon as the course roster is validated. Once you are assigned to a team, you will be given a space to post your weekly efforts. Please check IBM Connections Cloud on a **daily basis** and post questions about the course. This is the best way for us to know what parts of the course need clarification.

**Text Book.** Hull, E., Jackson, K., and Dick, J. *Requirements Engineering 3/E*. Springer, ISBN 978-1-84996-405-0 (<http://www.springer.com/computer/swe/book/978-1-84996-404-3>)

**Supplement:** Wiegers, K. and Beatty, J. *Software Requirements 3/E*. Microsoft, ISBN 978-0-7356-7966-5 (<https://www.microsoftpressstore.com/store/software-requirements-9780735679665>)

**Course Goals.** To prepare students to become proficient in applying software requirements engineering methods and techniques, working in cooperative teams, and managing projects. This course is also designed to promote your overall success, inside and outside the classroom. Our coursework will help you to improve in key areas such as Communication, Confidence, Critical Thinking, Leadership, Problem Solving, Social Responsibility and Teamwork. To find out more about the university's plan to improve student engagement and learning, visit the UTEP Edge.

This course is the first semester of a two-semester capstone project in which students work with a customer to capture and specify requirements for a real-world application. **In order to register for CS4311, students must pass this course with a C or better.**

<b>Grading.</b>	Assignments and quizzes	5% (includes participation)
	Connections Cloud entries	5%
	Project	45%
	Exams	45%

Handwritten notes in a spiral notebook will be allowed during all quizzes. Students are strongly encouraged to take notes during class and enhance their notes with other resources.

*Assignments and quizzes.* Assignments given as homework are **due at the beginning of the class. No late homework assignments will be accepted.** Problems that have been assigned as individual work can be discussed with other students in a general way, but the solutions must be done independently and the work must be unmistakably your own. Assignments completed as

a team, must include all student names on the first sheet. By signing your name, you are stating that you agree with the answers and can explain them to the TA or instructor. Students are expected to read assigned sections of the book and to be prepared to discuss the material in class. Quizzes will be given after each assignment to check for general understanding of the material.

*Project.* Draft project assignments are graded. Because a draft is work-in-progress, comments by the guidance team are **general**. The assigned letter grade, if applicable, reflects the effort *at that point in time*. The final grade is calculated independently of the rough draft.

Each team member is responsible for **documenting his/her work on the project** on Connections Cloud. The percentage of the project grade calculated towards the final course grade is determined through this documentation and the individual's contribution toward creating an effective team environment. The entries will be maintained through Connections Cloud. **No late project work will be accepted.**

Exams. **No make-up exams.** Exams are tentatively scheduled for:

Exam1: Oct 2, 2018  
Exam2: Nov 13, 2018  
Final Exam: Dec 13, 2018 7:00 am – 9:45 am (comprehensive Final)

**Attendance Policy.** Attendance is taken using iClicker. It is the student's responsibility to sign in for the class. A tardy will be recorded each time a student shows up **ten minutes after the start of class**. Your final grade will be lowered by **one point for every three tardies or for every two unexcused absences**. Students who exceed **three or more unexcused absences** will be dropped from the course.

**Standards of Conduct.** Students are expected to conduct themselves in a professional and courteous manner, as prescribed by the Standards of Conduct:  
<http://admin.utep.edu/LinkClick.aspx?link=HOOP-Section+II.pdf&tabid=30181&mid=63285>

Students may discuss assignments in a general way with other students, but the *solutions on individual work must be done independently*. Again, graded work must be unmistakably your own. Collusion and plagiarism will not be tolerated. Professors are required to report academic dishonesty and any other violation of the Standards of Conduct to the Dean of Students.

#### **Important Dates.**

Sept 3	Labor Day Holiday
Nov 2	Fall Drop/Withdrawal Deadline
Nov 22-23	Thanksgiving Holiday
Dec 6	Last day of classes
Dec 7	Dead Day
Dec 14	Final Exam (7:00am – 9:45am)

## Learning Outcomes

### Level 1. Knowledge and Comprehension

Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions. The material has been presented only at a superficial level. Upon successful completion of this course, students will be able to:

- a. Define basic software engineering concepts and principles (abstraction, anticipation of change, modularity, stepwise refinement, and separation of concerns).
- b. Define quality attributes such as availability, correctness, efficiency, interoperability, maintainability, portability, reliability, security, modifiability, availability, testability, and usability.
- c. State the main features of process improvement models, e.g., CMM, ISO, PSP, QPI, Plan-Do-Check.
- d. Define security design principles and the rule of least astonishment.

### Level 2: Application and Analysis

Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details. Upon successful completion of this course, students will be able to:

- a. Determine which life cycle model to use by analyzing different scenarios.
- b. Apply techniques for eliciting requirements.
- c. Analyze requirements to determine if they meet the attributes of well-written requirements.
- d. Identify risks in software development and project management.
- e. Analyze the course project and determine the local and global impact on computing on individuals, organizations, and society, including consideration of professional software engineering code of ethics.
- f. Relate the importance of professional societies.
- g. Engage in self-directed study to learn new techniques and tools for software requirements definition.

### Level 3: Synthesis and Evaluation

Level 3 outcomes are those in which the student can apply the material in new situations. This is the highest level of mastery. Upon successful completion of this course, students will be able to:

- a. Construct a feasibility report that demonstrates ability to engage in self-directed study.
- b. Conduct verification and validation using techniques such as inspections and walkthroughs.
- c. Construct a prototype, which adheres to basic HCI principles, to validate the user interface.
- d. Construct a software requirements specification.
- e. Analyze and model aspects of a problem by applying various modeling techniques.
- f. Demonstrate an ability to assemble and orally present technical work and compose technical documents that are grammatically correct and technically sound.
- g. Apply effective techniques for project management, collaboration, and problem-solving within groups.