CS460: Software Engineering
Spring 2020 Semester

Instructor:
Soraya Abad Mota   (email: soraya@cs.unm.edu)
                          Office: Farris Engineering Center 2040. Phone: 277-3052
                          Office hours: Monday 2:30-4:30 pm, Thursday 11:00 am - 12:30 pm
                                      and also by appointment.
Teaching Assistant:
Catherine Wright   (email: wrightc@unm.edu)

                          Office Hours: tbd

Class meets: Tuesday and Thursday 2:00 - 3:15 pm, Dane Smith Hall 327

1   Course Description

The UNM catalog entry for this course states: "Software engineering principles will be discussed and applied to a large team developed project. Other topics relevant to the production of software will also be covered, including ethics, legalities, risks, copyrights and management issues. Prerequisite: 351L."

To implement the goal stated above, state-of-the-art, industrially tested, specification and design techniques are presented, illustrated on small examples, and applied to realistic problems. The objective of the course is to develop an understanding of the technical and organizational complexities involved in software development and to teach key concepts and techniques used to manage these complexities. The students are required to design and build systems through team effort. Emphasis is placed on emulating the realities of an industrial organization. The classes are conducted in an intensive workshop atmosphere. The projects cover the principal system development life-cycle phases from requirements analysis, to software design, and to final implementation. Issues relating to human factors, dependability, performance, operating costs, maintainability and many others are addressed and resolved in a reasonable manner.

This course has been designed with the following philosophy:

- solid foundation

- software skill development – analysis, design, coding

- communication skill development – written and oral (between peers and with "clients")

- professionalism

- real-life experience enhanced by proper use of effective methods

- pragmatic ways to approach problem solving in a studio style atmosphere

- no reliance on specific tools or limiting methods

1.1 Desirable Entry Level Skills

This section provides examples of skills required or desirable for this class. These are skills that students should have acquired by now. They are grouped into basic requirements and elementary software engineering skills.

Basic requirements

- programming proficiency (e.g. 100-300 lines of dependable code in Java or C++)
- operating systems principles (drivers, mutual exclusion, scheduling)
- database systems
- socket-based communication
- specification, design, implementation distinctions
- basic technical writing abilities
- low-level code documentation (headers, in-line assertions)
- sets, functions, predicates, state machines
- semantic models

Elementary software engineering skills

Basic functional design: data structures, high-level description using words and pictures, initial state, commonly used operations, procedure, high-level algorithm specification, pseudocode (10/1 ratio of code to text), procedural abstraction, axiomatic specification (pair of assertions), operational specification (pseudocode), functional decomposition and recursion.

Object-oriented concepts: simple objects, proper encapsulation of data structures and devices, clean abstraction (application-oriented), specification: abstract state, operations over the abstract state, design documentation: data structures and procedures.

1.2 Learning Outcomes

At the end of the term, the student should have acquired the following software engineering skills.

Object-oriented design: composite objects, design diagrams, design strategies, design patterns, accommodating languages (e.g., C++ vs. Java), coping with design constraints (e.g., off-the-shelf packages), factoring performance, generalization (classes and templates).

System level design: design rules, processing and data allocation, design and specification of system interfaces, distributed algorithms design and verification, performance evaluation and estimation, feasibility analysis.

System level requirements: elicitation strategies, models, documentation methods, validation, formal specification, human interface design.

Technical management of large software projects: life-cycle, software development strategies (macro-level), technical reviews, written and oral communication, planning and commitments, group and corporate dynamics, system integration.

2   Logistics

All work this semester will be done in **teams**. The characteristics of the teams are:

- 4 to 5 students on each;

- one manager per team, with coordination responsibilities.

Each student must select a team on the first lecture. Team work puts emphasis on cooperative work. Industrial success is based on team performance and communication. One of the goals of process reviews is to avoid reliance on individuals. A team must blend and exploit the strengths of the individuals.

The instruction schedule is as follows:

- Concentrated lectures in the first 3 weeks. Other lectures will be distributed in the semester.

- Workshops: starting on week 4, students will participate in these activities every week, except when there are some additional lectures.

  - On Tuesday workshops: a document (TFS, RDD, SRS, or SAD) for a specific project will be due and each team will present their own document or read other groups documents. Ocassionally, a document from one team (only) will be selected to be refined. After refinement all teams will use that document as input for the next document.

  - On Thursday workshops: the revised documents are presented and teams will ask as many questions as needed in order to continue with the project using the revised document possibly authored by other team.

  - Each team will have to give at least two formal presentations during the semester. One for a design phase of one of the projects, the other one will be the final presentation of the second project which includes a demo of the prototype/simulator implemented for the second project. All teams will present informally during the phases of the two projects.

There are some exceptions to the activities during the workshops which are indicated in the tentative calendar attached.

- Each team will meet frequently (at least once per week) to plan the work or discuss specific aspects of the work. Each student of each team must invest between 6 and 9 hours per week of effective work to the current project. A portion of the grade depends on how much effort is invested in the project. To facilitate the evaluation of this aspect, by the instructor, the TA, and by the team coordinator, each student must record their time in a timesheet. The timesheets will be submitted to the coordinator who will register each team member's total of hours per week and total hours per project.

During the semester, there will be two projects. These will serve as the *Case Studies* in which students will practice the skills being developed. Tentatively, the two projects are:

1. Traffic Intersection Control System (TIC)

2. Cretaceous Garden Control System (CGC)

Each student is required to invest between 6 to 9 hours per week in the projects of this class. This time is outside the lecture or workshop attendance. This is the equivalent to 2-3 hours per credit hour per week. We suggest that each student reports to the manager of the team in a timesheet, the activities performed for the projects and the time invested each week.

3   Grading

Since most of the work in this class is done in teams, there are no tests or exams. Student input will be sought regarding the grades of the teammates. Presence in class and at meetings is compulsory for lectures and workshops. In-class participation is part of the grade. Appropriate time invested in the course is part of the grade.

Everyone starts with a B in the class. Students' final grade may go up or down according to the following grading guidelines:

```
A - exceptional individual contributions OR outstanding team work
B - good work - but the minimum expected
C - neglectful in responsibilities toward the team or low quality work on the
    projects
```

4   UNM learn platform

For all announcements and submissions of assignments we will use UNM learn available with your Net ID at learn.unm.edu. When you register for this class your UNM id is automatically included in the course platform list and this will allow you access to all the course materials. There will be no other formal website for this class. Students should be up to date with the announcements and material published in this platform.

5   Course and UNM Policies

This section contains the most important policies students are expected to comply with.

5.1   Specific Course Policies

1. Communication with instructor and TA will always be respectful. The instructor requires that you use the email address `soraya@cs.unm.edu` and **add CS460 to the subject line**.

2. Assignments will be handed out and collected using UNM Learn; assignments should only be submitted through learn, not email or other means. If you are unable to submit assignments on Learn due to technical difficulties, please email me the submission on time and we will coordinate later submission through Learn once the technical difficulties are resolved.

3. This course falls under all UNM policies for last day to drop courses, as described at http://www.unm.edu/studentinfo.html and in the UNM Course Catalog. Please see the UNM academic calendar for course dates, the last day to drop courses without penalty, and for financial disenrollment dates.

4. Any requests to drop the class or change grade mode (e.g. audit, CR/NC) with instructor permission must be made on or before the last regular class/lab meeting. Such request made after this date will not be approved except in the case of documented, extraordinary circumstances.

5.2   Academic Honesty

The university policy on academic honesty is contained in the Pathfinder; you should review this policy if you are unfamiliar with it. Any academic dishonesty will result in an automatic F for the entire semester and will be referred to the UNM Dean of Students for further disciplinary action as they deem appropriate. There will be no second chances or extra warnings.

5.3 Copyright issues

All materials in this course fall under copyright laws and should not be downloaded, distributed, or used by students for any purpose outside this course.

5.4 Title IX

Our classroom and our university should always be spaces of mutual respect, kindness, and support, without fear of discrimination, harassment, or violence. Should you ever need assistance or have concerns about incidents that violate this principle, please access the resources available to you on campus, especially the LoboRESPECT Advocacy Center and the support services listed on its website (http://loborespect.unm.edu/). Please note that, because UNM faculty, TAs, and GAs are considered "responsible employees" by the Department of Education, any disclosure of gender discrimination (including sexual harassment, sexual misconduct, and sexual violence) made to a faculty member, TA, or GA must be reported by that faculty member, TA, or GA to the university's Title IX coordinator. For more information on the campus policy regarding sexual misconduct, please see: https://policy.unm.edu/university-policies/2000/2740.html.

5.5 ADA

In accordance with University Policy 2310 and the Americans with Disabilities Act (ADA), academic accommodations may be made for any student who notifies the instructor of the need for an accommodation. It is imperative that you take the initiative to bring such needs to the instructor's attention, as I am not legally permitted to inquire. Students who may require assistance in emergency evacuations should contact the instructor as to the most appropriate procedures to follow. Contact Accessibility Resource Center at 277-3506 for additional information.

5.6 Citizenship and/or Immigration Status

All students are welcome in this class regardless of citizenship, residency, or immigration status. Your professor will respect your privacy if you choose to disclose your status. As for all students in the class, family emergency-related absences are normally excused with reasonable notice to the professor. UNM as an institution has made a core commitment to the success of all our students, including members of our undocumented community. The Administration's welcome is found on our website: `http://undocumented.unm.edu/`.

Tentative Calendar CS460 - Spring 2020

Week	Tuesday	Thursday
1 January	**21** Introduction. Syllabus discussion. Team constitution. Project 1 description. L1	**23** Software Life Cycle - fundamental concepts L2
2	**28** Requirements Engineering (1): Process, products. TFS research. L3	**30** RE(2): Requirements Definition Document (RDD) L4
3 February	**4 Technical Feasibility Study (TFS) due for Project 1**	**6** Methods and Models (1) L5
4	**11** Workshop #1: **RDD for Project 1 (TICS) due**	**13** Workshop #2: **Revised RDD for TICS**
5	**18** The Software Requirement Specification document (SRS) L6	**20** Software Architecture Design (SAD) and its document L7
6	**25** Workshop #3: **SRS for TICS due**	**27** Workshop #4: revised SRS for TICS
7 March	**3** Workshop #5: **SAD for TICS due** Shell for implementation	**5** Workshop #6: revised SAD for TICS
8	**10** Workshop #7: Prototype Implementation of Project 1	**12** Workshop #8 **Demo of prototype for Project 1 due** (Assign Project 2. Peer evaluations due over the weekend.)
Spring break	**17 Spring Break**	**19 Spring Break**
9	**24** Lecture on Methodologies L8 **TFS for Project 2 due**	**26** Lecture on Agile Methodologies L9
10 March/April	**31** Workshop #9: **RDD for Project 2 due**	**2** Workshop #10: revised RDD for Project 2
11 April	**7** Workshop #11: **SRS for Project 2 due**	**9** Workshop #12: Revised SRS for Project 2
12	**14** Lecture L10	**16** Workshop #13: **SAD for Project 2 due** + Implementation plan
13	**21** Workshop #14: revised SAD for project 2 + implemenation first report	**23** Lecture L11
14	**28** Workshop #15: Implementation	**30** Lecture L12
15 May	**5** Workshop #16: Implementation	**7** Final presentations