COMPSCI 320- Introduction to Software Engineering

Course Description:
Software engineering goes beyond designing and writing code. It involves communication with customers, users, and other technical and non-technical professionals. A software engineer must specify the behavior and operating parameters of a system based on the ideas of a customer who may not understand the technical aspects of software. A software engineer acts as a consultant to help elicit the customer's needs and translate them into formal specifications that can be used to design and build the required software system.

In this class you will participate in the specification, design, creation and testing of a software system for a customer. You will work as a member of a team under a project manager. You will gain the necessary background knowledge through the readings, in-class activities and homework assignments. You will gain practical skills by applying this knowledge to the development of the software system desired by the customer.

At the conclusion of the course each student should:

1. gain experience working in a software development team.
2. be able to analyze a proposed system and produce a set of requirements.
3. produce an efficient, modular design for a proposed software system.
4. understand the process of building software to specification on a schedule.
5. be knowledgeable about the economics of software development (the estimation and measurement of resources expended in building a software product).
6. comprehend the discipline of software engineering and the problems of developing hi-quality software with a limited amount of time and resources.

Prerequisites:
This course does not teach students to program. Software engineering is a larger concept than programming and both cannot be taught in a single semester class. Students should already be proficient in at least one programming language (such as Java or C++). Students should have successfully completed a course that deals with larger software programs that consist of and make use of code libraries and APIs to communicate with other systems. For example, an application that utilizes libraries to build a user interface and also communicates with a data base system.

Integrative Experience
CS 320 is an integrative experience course. It focuses on developing communication, reflection, and learning-aware learning skills. Each student will:

- engage in an in-class discussion at the start of the semester about what they will learn in CS 320 and how the skills they have acquired in their general education requirements will help them.
- engage in an a reflective in-class discussion at the end of the semester about what they actually learned, and which learning activities were the most effective.

- work extensively in groups.
- present his or her work orally in class.
- participate in active-learning activities with significant reflection components (including one on design, one on system understanding, and one on security in software).
- reflect on submitted assignments and improve and resubmit the work, and
- be guided by CS 529 students and their experience having previously taken CS 3**20.**

Course Structure and Attendance:
The course meetings consist of two "lecture" periods of 75 minutes each, and one "discussion section" of 50 minutes per week. There is no prescribed text-book for this course. Relevant material will be posted on Moodle and presented in lecture. In class, students will participate in activities that are designed to reinforce the concepts and techniques presented in the material and in lecture.

Attendance for lecture and discussion is mandatory, and is counted as part of the participation requirement for the course. Absences must be cleared with the instructor. Medical emergencies, religious or funerary events, university-related event (athletic event, field trip, or performance), extenuating non-academic reasons (military obligation, family illness, jury duty, automobile collision) will be accommodated with written documentation. It is expected that teams will schedule meetings and other development related activity outside of class.

Project Work:
This course has two main components: the development of a software system for an outside customer, and learning and applying software engineering concepts and techniques presented on the course website and in class.

Students will be divided into development teams. Each team will have a project manager assigned to it. Project managers are currently enrolled in CS529, software project management. Project managers will direct team members through the development process. Managers will provide the instructor with frequent updates on the team's progress and on individual team member's performance. The instructor will liaison with the customer who will attend all project deliverable presentations.

The project has the following parts: requirements specification, high level design, detailed design, test plan, beta product release. Each of these parts will be submitted and presented in class. Feedback from instructors and project managers as well as peers will be provided. Students will have a chance to refine and resubmit their work. Final project demonstrations will be given during final exam week. There are no exams in this course.

You will be assigned to a development team (10 members) as well as a team manager. Your manager's job is to learn the project development process from the management side. This process includes project planning by providing tasks for the team, monitoring quality and progress, and removing any obstacles to developing the system. Your manager is also learning to be a good motivator and communicator of the team members.

Each student's participation in project-based work will be evaluated by their team managers Although the instructor is responsible for your course grade, your manager is in the best position to assess your level of participation during the project's development. Your manager's assessment of your contribution to the project will weigh heavily on your project grade. It is in your team's interest to work with your manager toward the creation of a successful software system. If you encounter any difficulties with your manager or any team members, you are encouraged to discuss them with the individual in a civilized manner. If you are uncomfortable with this option, then you are encouraged to make an appointment with me to discuss any issues that arise. Any discussions of this nature will be held in confidence.

General Course Outline:

The course consists of a sequence of sections that roughly correspond to the phases of software development.

Development Phase	***Topics***	***Activities***
Project Plan	Process planning; effort, schedule, and resources estimation; quality plans and risk management.	Customer presents project.
System analysis	Analysis techniques such as diagramming, questionnaires, interviews.	Determine what the system has to do, any constraints.
Requirements and Specification	Identify functional and non-functional requirements, Creating a formal document, Use cases.	Convert analysis results into formal specification.
High-Level Architecture Design	Choice of architecture, use of libraries, frameworks, APIs.	Define the high-level modules of the system and how they communicate.
Low-Level Design	OO code design, choose data structures, algorithms, interfaces between classes.	Define the detailed structure of high level modules
Testing	Levels of testing, why testing cannot be 100%	Create a test plan for the system.
System Implementation and Presentation	Writing code to a spec/framework. Good coding and documentation practices, presentation of system.	Coding and testing of the system with alpha and beta releases.

Course Grading:
Your grade will consist of individual and group evaluations. Individual grade consists of the scores for homework assignments, in-class presentations, peer evaluations, as well as class attendance. Group work consists of four project deliverables. The breakdown of the evaluation components and their grade weight is as follows:

Intended Learning Outcome	**Assessment Type/Deliverable**	**Individual Grade 50%**	**Project Grade 50%**
Demonstrate skills that support the project development: Recall, recognition, comprehension of concepts and principles of software engineering.	Homework assignments	20%	
Present orally with clarity of material and supported by visual material.	In-class presentations of project deliverables and presentations to team.	15%	
Show evidence of attendance and participation at group meetings, timeliness and availability, communicating, as well as contribution to the project.	Peer (team member) grade	10%	
Show evidence of in-class attendance participation in class activities, and contribution to discussions and reflections.	Assessment by instructor	5%	
Develop a project management plan that includes scheduling, quality, and risk assessment estimates.	Project Plan document		5%
Develop statements of requirements and specification based on the true needs of users and other stakeholders.	Requirements Specification document		10%
Define the high-level architecture modules of the system and how they communicate.	High level design document		10%
Design the detailed structure of high-level modules. Apply and evaluate design and testing strategies.	Low level design document and Test Plan		10%
Implementing a specified functionality and presentation of system.	Working prototype		10%
Demonstrate participation in group meetings, responsiveness to and use	Manager's assessment of team member's work		5%

of communication, as well as quality of individual contribution to the required project deliverables.			

Grading policy
Grades will be available through Moodle and you should check them regularly and review any provided feedback. If you encounter any issues with your grades, you have 1 week past the return of your grades to contact the course staff so we can investigate. We will not accept questions about individual grades beyond one week, so you must be prompt.
Final project grades are based on the quality of the product deliverables and on teamwork processes. Note that grades for project deliverables are shared by the team weighted by your manager's evaluations of your individual performance.

Late submissions
Late work will not be accepted. If you need an extension for an assignment, contact me at least 24 hours before the assignment is due. Medical conditions, religious or funerary events, university-related event (athletic event, field trip, or performance), extenuating non-academic reasons (military obligation, family illness, jury duty, automobile collision) that need extension will be accommodated with written documentation. Problems with computer or internet access, holiday or family travel are not valid excuses.

Accommodation Statement:
Any student who requires an accommodation due to a disability is directed to contact the UMASS Disability Services: http://www.umass.edu/disability/ to obtain the appropriate accommodation forms. If you have a documented disability that requires an accommodation, please notify me within the first week of the semester. My goal is to provide every student with a high-quality learning experience. Please contact me if you have any questions or concerns about disabilities or any issue that may impact the quality of your learning..

Academic Honesty Statement:
In general, collaborative work is encouraged in teams. For individual assignments, please refer to UMASS Academic Honesty Policy and Procedures for guidelines on what constitutes academic dishonesty and the sanctions that may be imposed on any student who has committed an act of academic dishonesty (http://www.umass.edu/honesty/).

Classroom conduct
In this course, each voice in the classroom is valued. We honor UMass's commitment to embrace diverse people, ideas, and perspectives to create a vibrant learning and working environment. This course requires students to work in groups and to present material in class. As such, it is expected that all students will observe a "workplace" level of social decorum at all times when interacting with managers and peers. Students are expected to treat their work in this class in a professional manner and to treat all individuals with dignity and respect. Please consult the UMass Guidelines for Classroom Civility and Respect:

http://www.umass.edu/dean_students/codeofconduct/classroomcivility.