Software Development Practices

Drs. Chang Yun and Venkat Subramaniam

Course Description: The objective of this course is to bring students up to speed on pragmatics of software development. In this hands-on course, the students will work towards developing industrial strength software systems using state of the art techniques and tools. Emphasis will be placed on estimation, planning, risk management, automated testing, development and deployment. Various good practices in software development, including pragmatic programming and agile software development practices will be emphasized.

Pre-requisite*:

- You must have completed COSC 4353 or equivalent
- You must have completed COSC 1320 or equivalent
- You must have completed COSC 2320 or equivalent

* Pre-requisites will be enforced (expected to have knowledge of OO, Data Structures, software development life cycle, with a strong emphasis on test driven development). If you have any questions, please contact the instructors. Please fill the pre-requisite certification form at http://bit.ly/2hsX77C well before the first day of the class.

Course organization: There will be lectures/presentations on the first day of the class and project sponsors' presentations in the second week. For the rest of the semester, the course will be primarily driven by student project iterations. There will be a project progress review every two weeks with deliverables at the middle and end of the semester.

Lecture Schedule:

January 15 - Introduction to course, expectations, process, project expectations, and student evaluations

January 22 - Project Sponsor Presentations

Project: Each student will participate in a software project which will have an external sponsor. The project will be designed and implemented in consultation with the sponsor. The instructors will oversee the project progress in terms of the requirements, progress of the work, and implementation. The success of the project is measured based on the number of criteria including, but not limited to, working software delivered, continued progress, and pace through out the semester. The process and practices followed and quality of code, the ability to accommodate reasonable change in requirements, feature completeness, and progress in terms of schedule will be considered. Use of tools, techniques, practices, and facilities that reduce risk and promote probability of success will be encouraged.

Team Requirements:

- Each team should have 4 to 6 students
- Each team is required to have a minimum of 2 undergraduate students
- Each team is required to have at least 1 graduate student
- Each student is required to be part of one and only one team
- Any exceptions from these requirements require the approval of the instructors.

Project Schedule:

- January 15: Class Introduction
- January 22 - Update Wiki with your team information. Name and email address of each member along with a designated team lead/contact person.
- January 22 - We will post project details by this date. Please review those before project presentations.
- January 22 - Project presentation by Sponsors.
- January 24 - Update Wiki with your project preferences by this date.
- January 25 - Your project assignment will be posted on the Wiki
- January 29 - Iteration starts - Iteration duration every 2 weeks
- February 12 - First Iteration ends

- February 26 - Second Iteration ends
- March 11 - Spring Break (March 9-13)
- March 18 - Third iteration ends; First incremental release
- April 1 - Fourth Iteration ends
- April 15 - Fifth iteration ends
- April 29 - Project Release
- April 29 - Final Demo and Discussions

Grading:

Grades for project components will depend on the overall group effort as well as the individual contribution of students.

Iteration demo and progress 50%
First incremental deliverable 10%
Final deliverable 15%
Final demo and report 15%

Students evaluation of project 5%

Individual Peer review 5%

Some details of grading may change but this should give you a good idea.