Software Development Practices

Dr. Jaspal Subhlok

Dr. Venkat Subramaniam

Course Description: The objective of this course is to bring the students up to speed on the 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, 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, and software development life cycle). If you have any questions about them, please contact Jaspal. Please fill the pre-requisite certification form available on the course website and bring it to the class on the first day.

Topics:
Agile Software Development
Principles and practices
Tools for agile development
Unit Testing
Continuous integration
Continuous Feedback
Estimation
Measuring progress
Measuring Quality

Course organization: There will be lecture/presentations for the first 3 weeks. After that the course will be primarily driven by student project iteration with some guest lectures and other presentations. There will be a project progress review every two weeks with deliverables at the middle and end of the semester.

Lecture Schedule:

January 17 - Introduction to course, expectations, process, project expectations, and student evaluations

January 17 - Introduction to Agile Methodologies and Practices

January 24 - 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 the 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.

Project
 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 2 graduate students

   Each student is required to be part of one and only one team

Each project team will have a team lead

Any exceptions from these requirements require the approval of the instructors.

Project Schedule:

- January 19 - Update Wiki with your team information. Name and email address of each member along with a designated team lead/contact person.
- January 19 - We will post project details by this date. Please review those before project presentations.
- January 24 - Project presentation by Sponsors.
- January 26 - Update Wiki with your project preferences by this date.
- January 27 - Your project assignment will be posted on the Wiki
- January 31 - Status update by teams and lecture on presentations
- February 7 - Iteration starts - Iteration duration every 2 weeks
- February 21 - First Iteration ends
- March 20 - First incremental release
- May 1 - Project Release
- May 1 - 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%

Student evaluation of project 5%

Individual Peer review 5%

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

Recommended

 Text: There are no required texts for this course. Some recommended texts

 are:

Agile Software Development: Principles, Practices, and Patterns by Robert Martin

Pragmatic Project Automation: How to Build, Deploy, and Monitor Java Apps by Mike Clark

Test Driven Development: By Example by Kent Beck

Pragmatic Unit Testing by Dave Thomas and Andy Hunt