Class meeting times:  T, Th  10:00 – 11:15  
Meeting Room:  Computer Science 142  
Course Web Site:   http://laser.cs.umass.edu/courses/cs520-620.Spring15/  
TA:  Hee Tae Jung (htjung@gmail.com)  
TA Office Hours:  Mon. 1:00-2:00pm, LGRC A307 
Text:  None Required, but  
   Ghezzi, Jazayeri, and Mandrioli, Fundamentals of Software Engineering  
   has helpful supplementary reading.  
   Some Course materials will be made available online.  
   Readings will also be suggested.  
Instructor slides:  Will be on course web site (hopefully, usually) before lecture  
Instructor Office Hours:  by appointment

Course Summary:  This course examines the varied approaches to the development of computer software.  The course focuses on using software processes to produce software products.  We examine various ideas about how software products should be structured and function.  We then examine how software processes serve as vehicles for manufacturing such products.  We view the processes as manufactured artifacts as well.  A key perspective of the course is to apply technologies for manufacturing software products to the manufacture of the processes that make these products.  This approach facilitates the direct study of different software development approaches, and a more direct study of their effects on the products they produce.

Educational Objectives:  The course will acquaint students with various approaches to developing complex software systems.  Of more importance, however, the course is intended to help students develop skills in analyzing and evaluating different approaches to software development and making prudent choices for how to select development approaches that are most appropriate for different software projects.

Evaluation:  The primary basis for evaluation will be 3-4 major projects.  620 students will be required to demonstrate a broader and deeper grasp of subject material in their project submissions (e.g. through more effective use of mathematics and rigor).  Additional project sections may be required of 620 students.

Examinations:  There will be no mid-term examination.  There will probably be a final examination, which will probably be a take-home exercise.

Course Projects:  There will be 3-4 large course projects.  The projects will NOT be group projects, but will represent individual effort.
620 Students (only): Will be required to write a research term paper. The paper will probably have to be presented in class at the end of the semester.

Grading: The student’s final course grade will be based primarily upon the student’s project grades. The grade on the final examination will carry approximately the same weight as one of the projects. For 620 students, the grade on the research term paper will be weighted at least as heavily as one of the projects. Class attendance, presentations, and participation will also be taken into consideration in determining the final grade in the course.

Course Policies: You are expected to attend class. The official means of communication for this course will be in-class announcements. Missing class is not an excuse for failure to act as required by these announcements. In addition, announcements related to issues that arise between class meetings will be made electronically. In particular, clarifications of assignments, changes to due dates, etc. may be reported to you via postings to the course web site and/or electronic mail. It is your responsibility to log in and check the course web site, and your mail, regularly.

Copies of handouts, homework assignments, and course notes will be posted for downloading from the class website. The home page for this course is http://laser.cs.umass.edu/courses/cs520-620.Spring15/

IMPORTANT:
The work submitted for grading must be your own work. Submission of work that is not your own is considered academic dishonesty. Computer Science department policy specifies that the penalty for academic dishonesty is 1) a final course grade of `F', and 2) possible referral to the Academic Dishonesty Committee.

You may be using copyright-protected software in this course. United States Federal law and license agreements between the University and various software producers prohibit copying this software for any purpose. Such activity will be regarded as a form of academic dishonesty and will be dealt with as such.

An incomplete will usually be given only when documented, exceptional circumstances beyond your control have made it impossible to complete the assigned work before the end of the semester. It is your responsibility to contact the professor regarding any such problems as those circumstances develop. Note that the general rules of the University allow an incomplete only if most of the work has been completed before the end of the semester so that the incomplete can be finished within the first four weeks of the immediately following semester.

The instructor will be available immediately after class and by appointment
Shortened Outline of Course Topics:

1. Introduction to the course, to software, to software engineering
   a. Why is software important? Why is it hard?
   b. Products and Processes
   c. Synthesis and analysis

2. Software artifact representations
   a. Software Models
      i. DFGs, FSAs, Petri Nets, CFGs, etc.
   b. Multirepresentation approaches
      i. UML, Statemate, etc.

3. Anatomy and Physiology of a software product
   a. Requirements, Architecture, Design, Code, Testcases, etc.
   b. Constraints and relations

4. Analysis, constraints, and relations
   a. Why, when, and how
   b. When to do what

5. Processes and their models
   a. Lifecycle processes
   b. Modeling lifecycle processes
   c. Microprocess vs. macroprocess
   d. Some process modeling approaches
      i. Funsoftnets, Marvel, Slang
   e. Little-JIL process definition language

6. Specific software processes and their products
   a. Requirements definition
   b. Design
   c. Architecture
   d. Coding
   c. Extreme Programming
   d. Testing and Analysis
   e. Formal verification
   f. Rework
   g. Evolution
   h. Configuration management
   i. Reengineering, reverse engineering

7. Additional topics (as time allows)
   a. Software project management
   b. Software cost estimation
   c. Software economics
   d. Global software development