

Syllabus for CSC-383 Software Engineering II Fall 2005

Instructor Information:

- Dr. Michael Farmer
- Office Hours: Tue. 1:00-4:00
- Office: 214 MSB
- Phone: (810) 762-3423
- Email: farmerme@umflint.edu

Course Information:

- To date most of your classes have required you to write many lines of code. You have spent many late nights searching for syntax errors, etc. In other classes such as the algorithms class you have learned the detailed design of some common algorithms. For example, you have learned the cost in terms of memory and speed of many standard tasks, such as searching and sorting. Recall in CS-382 we learned the engineering **methods and techniques** associated with the key stages of engineering a software system. Now in CS-383, we will address the aspects of **management and control** of the development of a software product. The objective of this class is to provide the student with the knowledge of a variety of process models within which to apply the tools and techniques developed in 382. It also will provide the student with the skills to estimate the costs of a development, develop metrics to manage the development, and the standards with which to document the software development.

Course Objectives:

- The goal of the class is to continue the evolution from programmers to software engineers to ultimately software development leaders. You will learn the management techniques needed to accomplish this. Our final products are project management plans which define our process, our metrics for monitoring progress, and our documentation for capturing the essential knowledge of the project. The objective is to provide the students with a good understanding of the standard process models that professional engineers use to minimize the development risk of a software product. In industry this skill is what separates a typical programmer from a potential project leader.
- As with CSC-382, there will be **no code written** in this class. We are focusing on understanding how to establish, manage, and document a software development project. The tools and methods to make this job easier are a key element of what will be learned in this class

Class Format:

- The best way to learn Software Engineering is by **doing**. We will use lectures to explain techniques and methods we will be using. Then we will use on-going in-class case study to try our hand at the techniques. This is where class participation is critical. We will be heavily discussing the merits and shortcomings of the various software management processes, and everyone’s contributions are expected.

Prerequisites:

- CSC-382 or Consent of Instructor

Book:

- *Software Engineering: A Practitioner’s Approach, 6/e*, by Roger Pressman.

Other Good Reference Books:

- *Software Engineering: 7/e*, by Ian Sommerville
- *Introduction to Software Engineering*, by Panjab Jalote
- *Software Management* by Donald Reifer
- *The Personal Software Process*, by Watts Humphrey
- *The Mythical Man-month* by Frederick Brooks Jr.
- *Extreme Programming*, by Kent Beck

Tools:

- *Microsoft Project*
- *Microsoft Word* – available everywhere
- *COCOMO* (cost estimating software)

Grading:

- Homework (20%), Mid-term (25%), Project (25%), Final (30%)
- Class participation is an expected portion of the course work
- Scale:

A’s	B’s	C’s	D’s
97-100: A+	87-89: B+	75-79: C+	65-69.9: D+
92-96.9: A	82-86.9: B	72-74.9: C	62-64.9: D
90-91.9: A-	80-81.9: B-	70-71.9: C-	60-61.9: D-
			0-59: E

Attendance:

- Regular class attendance is essential. The discussions will evolve from week to week and you will have trouble participating effectively without regular attendance.

Late Homework Policy:

- Having worked for 20 years to schedules devised by others, I know things happen that sometimes prevent on-time delivery. I will not take-off points for late assignments, but please note that *once the answers are provided I can no longer accept papers*. I will post solutions by the **next** class after the assignments are due. Please note a change from last semester – *No emailed homeworks assignments will be accepted*. Please turn all homework assignments directly into me in class.

Academic Misconduct:

- Copying others work, plagiarizing external references without giving due credit, and cheating in exams are strictly forbidden. Please take pride in your own work, and feel free to ask me if you need help or assistance to maximize your learning.

Proposed Schedule: (Note this schedule is subject to change)

Week	Monday	Wednesday
Week 1 (8/29)		Overview & Introduction to SW Engineering (Ch. 1)
Week 2 (9/5)	<i>Labor Day</i>	Review of Last Semester (the tools of the trade)
Week 3 (9/12)	Overview of the software engineering problem (from Mythical Man-Month)	Software Quality Systems CMM, ISO-9001, SPICE, etc(Ch. 2)
Week 4 (9/19)	Personal and Team Software Processes (Ch. 2)	Common Process Models (Ch. 3)
Week 5 (9/26)	Adaptive and Agile Process Models (Ch. 4)	Extreme Programming (Ch. 4 plus “Xtreme Programming”)
Week 6 (10/3)	Project Management Concepts (Ch. 21)	Project Management Concepts (Ch. 21)
Week 7 (10/10)	Software Metrics (22.1, 22.2) and (15.1 and 15.2)	Software Product Metrics (Ch. 15.3,15.4, and 15.5)
Week 8 (10/17)	Review for Midterm	Mid-term
Week 9 (10/24)	Software Quality and Metrics (Ch. 22.3-22.6)	Software Cost Estimation (Ch. 23)
Week 10 (10/31)	Software Cost Estimation (Ch. 23) & COCOMO Cost estimation tool	Software Project Scheduling (Ch. 24)
Week 11 (11/7)	Risk Management (Ch. 25)	Software Quality Management (Ch. 26)
Week 12 (11/14)	Software Quality Management (Ch. 26)	Change Management (Ch. 27)

Week 13 (11/21)	Change Management (Ch. 27)	<i>Thanksgiving</i>
Week 14 (11/28)	Software Documentation Standards	Building the SW Development Environment – CASE Tools
Week 15 (12/5)	Advanced Topics I	Advanced Topics II
Week 16 (12/12)	Project Presentations	Class Review