CS 410-02: Introduction to Software Engineering
FALL 2025

Dr. J. Holly DeBlois
Office: McCormack, 3rd floor, room M-3-201-32
Office hours: Tues/Wed/Thurs 12:30-1:30pm

Lectures and Class: Tuesday & Thursday, 4:00-5:15pm, H04-0031 (course 3983)
Instructor Email: jane.deblois@umb.edu
Instructor Website: Lecture notes and assignments: https://www.cs.umb.edu/~hdeblois/cs410/f25/
Canvas: Grades are on canvas: https://www.umb.edu/canvas/
Portal: Register for CS410 to create your course directory at https://portal.cs.umb.edu
Piazza: Join Piazza: https://piazza.com/umb/fall2025/cs410/

Course Description: In CS410, we present all aspects of the software development process from initial specification to final validation using different approaches to the development process, contrasting the older "waterfall" approach to the newer "agile" methods, especially "scrum."

Textbook: The main reference in this course is *Essential Scrum, A Practical Guide to the Most Popular Agile Process*, (Pearson, 2013) by Kenneth S. Rubin.

Attendance: Mandatory. **If you miss more than 5 classes, your final grade will be lowered: miss 6-7, lower by one step; miss 8 or more, lower by two steps.** See grade steps in chart on last page.

Topics: We cover the following general topics:

- **Writing code for a client as part of a team of software engineers –** how it is different
- **Requirements specification** – what does the client want and what is also a good idea
- **Websites on the CS server** – yours and for a team
- **Git, github and version control** – to be able to integrate code written by different people
- **Design and code in class –** while working on your longer team project
- **Test-driven code –** on a team, testing identifies strong options for the design objectives
- **Adopt a code style for the team –** write so others can read it
- **Implementation methodology** - instead of waterfall we mostly use agile methods and scrum
- **Oral presentations –** you present what you built for the small projects and the longer project
- **Presenting the code to the client for validation** - the team goal

In-Class: Bring a computer capable of accessing the CS Linux servers. We write tests and code, and compile and run in class. If you do not have a suitable laptop, let me know. We'll provide recharging.

Note: No courses required by the CS major, minor or certificate may be taken pass/fail.

Prerequisites: CS 310 Algorithms, CS 220/320L Discrete Math and any CS 400 level course or permission of the instructor.

Evaluation: Your grade in the course is determined by: Daily reports (10%, two lowest report grades are dropped), individual startup project (10%), small team projects (20%), homeworks (50%), long project final presentations (10%).

Daily reports cover scrum meeting, what you designed, tested and coded during class, your tests and your comment.
Startup project covers building your CS website and symlinking it to your resume.
Small team projects include creating a corporation website and a project that uses git as a team. You individually present your personal work and update your resume.
Homework will cover the textbook and applications of scrum. Three will be on textbook topics. The other two will be on personal work for sprint1 and sprint2 of the long project.
Long project will be for a client with about 6 students per team. We assess the team's work: requirements, test plans and test results, which must be uploaded to the CS servers, and code, both on git and deployed, which must be clearly written and accessible from the CS servers as well as from the client desired location. Assignments may change if necessary.
F**inal presentations** cover the personal work of each student: three slides and a five-minute presentation. The long project final result itself is not graded, but it will be presented to the client for validation. Long projects will be demonstrated in the final class.
Attendance: counts as explained on page 1.

Collaboration policy: Most of the work is collaborative, but you must be sure the homework you submit is your own ideas in your own words. On the team projects, collaboration may be needed to decide who builds what. Note: if someone's work is going slow, the work can be redesigned to be done by several people or several ways. Collaboration means deliver what the client wants.

Late Submission or Late Arrival: Homework and short project submissions are scheduled for precise times. **Late submission loses 1% of the score per hour late.** Please be on time to class since meetings are held in the first 15 minutes and are to be reported in your daily reports.

Academic integrity will be strongly enforced. Your requirements, homework and reports must be your own work, not AI-generated. See **https://www.umb.edu/academics/provost/academic-integrity/**
Make personal use of AI carefully. Please ask questions about what is allowed. We recommend using free access to chatGPT, not saving prompts and if you generate code, you are required to record your prompts in the code. Since UMB does not have any particular guidance for sourcing code yet, we will use the MIT guidance: **https://integrity.mit.edu/handbook/writing-code/**

Additional references:
Manifesto for Agile Software Development (2001**) - http://agilemanifesto.org/**
Pro Git: Everything You Need to Know About Git, 2nd Ed., Scott Chacon, Ben Straub, (APRESS, 2014)
The Internet Book: Everything You Need to Know about Computer Networking and How the Internet Works, 5th Ed., Douglas E. Comer, (CRC Press, Taylor & Francis Group LLC, 2019)
www.w3schools.com for html and javascript
https://www.gnu.org/software/emacs/refcards/pdf/refcard.pdf explains the emacs editor

Accommodation: Section 504 of the Rehabilitation Act of 1973 offers guidelines and support for curriculum modifications and adaptations for students with documented disabilities. Contact the Ross Center at 617-287-7430 and please discuss your accommodations with the instructor.

Student Conduct: You must be honest about how long tasks take, creative about design and team assignments and work well with others as software engineer or servant leader. The University

presupposes that work for academic credit is the student's own and complies with University policies above and here: **https://www.umb.edu/camp-life/dean-of-students/student-conduct-process/.**

Syllabus and Schedule Subject to Change: The instructor reserves the right to change the syllabus when necessary and will let you know. Here is the tentative schedule (28 classes over 15 weeks):

week	class/date	chapter & pages	lecture topic & class work	assignments due
1	#1-Tue 9/2	ES2-4, Fig1.3	Syllabus, Agile Principles, Servant Leader	r1, handwr1, hw0, hwk1 posted
	#2-Thu 9/4	ES5-6	w3sch:html, javascript, Comer Ch21-22	r2, classproj1 starts, handwr2
2	#3-Tue 9/9, add/drop ends	ProGit 1-6,		r3, classproj1due
	#4-Thurs 9/11	ES9-13	corporations, corp websites, Comer Ch25,29	r4, hwk1 due, classproj2 starts
3	#5-Tue 9/16	ES14-15		r5, hwk2 posted, classproj2 due
	#6-Thu 9/18	Comer,part1		r6, classproj3 starts
4	#7-Tue 9/23	ProGit 7-8		r7, classproj3 due
	#8-Thu 9/25	Comer,part3	do parts, integrate	r8, hwk2 due
5	#9-Tue 9/30	ES19-23		r9, hw3 posted
	#10-Thu 10/2, NA grades due			r10
6	#11-Tue 10/7		client projects presented	r11
	#12-Thu 10/9		meet as teams, plan sprint	r12, hwk3 due
7	#13-Tue 10/14		Sprint1	r13
	#14-Thu 10/16	prod bcklog	Sprint1	r14
8	#15-Tue 10/21		Sprint1	r15
	#16-Thu 10/23		Present	r16, hwk4 posted
9	#17-Tue 10/28		Sprint2	r17
	#18-Thu 10/30		Sprint2	r18
10	#19-Tue 11/4		Sprint2	r19, hwk4 due
	#20-Thu 11/6	No class	Present	r20, hwk5 posted
11	Tue 11/11, VeteransDay		Continue sprints	
	#21-Thu 11/13			r21
12	#22-Tue 11/18		Holiday	r22, hwk5 due
	#23-Thu 11/20			r23
13	#24-Tue 11/25	No class		r24,final slides due
	Thu-Sun 11/27-30		Pass/Fail/WithdrawDeadline	
14	#25-Tue 12/2			r25,schedule client validations
	#26-Thu 12/4		Thanksgiving Recess	
15	#27-Tue 12/9			
	#28-Thurs 12/11	Last class	Final Presentations Final Presentations Project demos, course eval.	

Table of score to grade conversions (default grading scheme called “UMB letter” in canvas):

93 <= S = A
90 <= S < 93 = A-
87 <= S < 90 = B+
83 <= S < 87 = B
80 <= S < 83 = B-
77 <= S < 80 = C+
73 <= S < 77 = C
70 <= S < 73 = C-
67 <= S < 70 = D+
63 <= S < 67 = D
60 <= S < 63 = D-
S < 60 = F