CS 410: Introduction to Software Engineering

SPRING 2026v6

Dr. J. Holly DeBlois

Office: McCormack, 3rd floor, room M-3-201-32

Office hours: Tues/Wed/Thurs 2:30-3:30pm

Lectures and Class:	CS410-01: Monday & Wednesday, 4:00-5:15pm, McC M02-0404 (course 1248) CS410-02: Tuesday & Thursday, 12:30-1:45pm, McC M02-0116 (course 3631)
Instructor Email:	jane.deblois@umb.edu
Instructor Website:	Lecture notes and assignments: https://www.cs.umb.edu/~hdeblois/cs410/s26/
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/spring2026/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." We code several short projects and then a long project for a client.

Textbooks: The main reference in this course is the textbook: E*ssential Scrum (ES), A Practical Guide to the Most Popular Agile Process*, (Pearson, 2013) by Kenneth S. Rubin. The second reference is: *Manifesto for Agile Software Development* (2001**) - http://agilemanifesto.org/** . The third reference is: *Pro Git (PG): Everything You Need to Know About Git, 2nd Ed.*, Scott Chacon, Ben Straub, (APRESS, 2014) https://git-scm.com/book/en/v2 .

Attendance: Mandatory. Only university designated excused absences are addressed.
Note: 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 –** requires new skills
- **Requirements specification** – translate what the client wants into various requirements
- **Websites on the CS server** – draw and build singly and in teams
- **Git, github and version control** – be able to use git and github to collaborate
- **Design and code in class following agile methods –** select, design, build, test and report daily
- **Test-driven code –** save time by identifying tests first, to drive what and how you build
- **Artifacts –** store the work items like product backlog and tests that you create along with code
- **Presentations –** present single slides to practice presenting and then present three slides for your work on the long project
- **Prepare an academic research paper –** show research side of how you build team software

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, the library IT can provide one.

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. We offer optional ungraded hw0 to assure that you can work on the CS servers. Access your course directory without changing privacy settings -- do not use a linux move (mv) command, instead use a copy (cp) command and preserve timestamps (cp -p <file>).

Evaluation: Your grade in the course is determined by: daily reports (20%), startup website projects (10%), small team projects (20%), homework (30%), long project slides and presentations (10%), a research paper about your work (10%) and whether you miss more than 5 classes (explained above).
Daily reports cover scrum meeting, what you designed, tested, coded during class, and test results.
Startup website projects cover building CS websites several ways using agile methods, a symlink to nnnn.txt (short resume), and making a slide about your work, which you present to the class.
Small team projects include using git and github to develop code together, and focus on an integrated graphics idea, rudimentary front-end and back-end ideas and using system diagrams to develop requirements and build code and artifacts. You update your resume as your skills expand. Sprint 1 activities, slide and presentation are included as part of small team projects.
Homework will cover the textbook information applied to code developed using agile methods.
Long project will be for a client with 6-8 students per team. You will translate what the client desires to have built into requirements, tests and code. You will follow the agile methodology producing runnable code in increments, stored both on github and deployed. Assignments may change if necessary. Sprint 1 slide, presentation and activities are graded as small team projects. The long project itself is not graded, but it will be presented to the client for validation. Demos may be given in the last class. Students are graded on the quality of their requirements, code and artifacts.
Research paper on your long project work, to include topic, references, introduction, 3-5 sections, conclusion and footnotes. You will submit two drafts, get comments back and submit a final report.
F**inal presentations** cover the personal work of each student on the long project: it will include three slides and a three-minute presentation. Demos of the long projects in the last class are for fun!

Collaboration policy: Most of the work is collaborative. Be sure the homework, reports, academic paper and presentations you submit are your own ideas. Note: if someone's work is going slow, the work can be reassigned. You get credit for your code even if it is not included in the final project.

Late Penalties: Homework, reports 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 scrum meetings are held in the first 15 minutes and are to be reported in your daily reports.

Academic Integrity and Student Conduct 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 the 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/** . In addition, since neither has guidance for citing chatGPT or other large language model generated code, **document any code you generated and used by marking START, citing the prompt you entered and further down marking END.** We recommend you take steps to not allow chatGPT or any large language model to store your results. If you have a series of prompts, list them in your readMe.txt and put only the last one in the code.
Student Conduct: You must be honest in all your conduct. 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/.

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.

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 (29 classes over 15 weeks):

week	class/date	reading	tracker	assignment
1	#1-Mon 1/26 or Tue 1/27 **Snowdays**			
	#2-Wed 1/28 or Thu 1/29	Syllabus, PG ch1-2	introduction, git, agile principles	Website part1, tests, in class draw1, login, w, emacs, nnnn.txt, Report R01
	Thurs 2/5(changed) add/drop ends			
2	#3-Mon 2/2 or Tue 2/3	ES ch2-3, Fig 1-3 p7	hw1, websiteproj	Post hw1, website part2, tests, in class draw2, emacs, cp data file, git, report R02
	#4-Wed 2/4 or Thurs 2/5	ES ch4-5	proj artifacts	Website part3, html, git, symlink file, R03
3	#5-Mon 2/9 or Tue 2/10	PG ch3-4	update nnnn.txt	Prepare presentation, present, **no R04**
	#6-Wed 2/11 or Thu 2/12	ES ch6-7	hw2, team proj1	Submit hw1, post hw2, Bend, git local, R05
4	**Mon2/16Holiday**:no class cs410-01			
	#7-Tue 2/17	PG ch5-6	artifacts, tproj2	Celtic html, w3school or jsfiddle, R06
	#8-Wed 2/18 or Thu 2/19	ES ch8-9		Celtic html2 and R07 (Wed for cs410-01)
5	#9-Mon 2/23 or Tue 2/24	ES ch10-11	update nnnn.txt	Submit hw2, do slide, present, **no R08**
	#10-Wed 2/25 or Thu 2/26	ES ch19-20	res. paper req	Post hw3, do paper topic, intro,refs,**noR09**
	Fri 2/27 N/A grades due			
6	#11-Mon 3/2 or Tue 3/3	client zoom	long proj teams	long proj start, sprint planning, github, R10
	#12-Wed 3/4 or Thu 3/5	client zoom	longproj backlog	Submit hw3, sprint1, all teams, R11
7	#13-Mon 3/9 or Tues 3/10	client zoom	sprint meetings	sprint1b, R12
	#14-Wed 3/11 or Thu 3/12	res. paper	intro/flow/conclu	sprint1c, individual slide, Draft1due, **noR13**
-	**3/15-3/22 Spring Break**			
8	#15-Mon 3/23 or Tue 3/24	cl. zoom...		Presentations, Comments given, **no R14**
	#16-Wed 3/25 or Thu 3/26	adjust lp...	product backlog	sprint2, all teams, R15
9	#17-Mon 3/30 or Tue 3/31		artifacts	sprint2b, R16
	#18-Wed 4/1 or Thu 4/2	res. paper	system diagrams	sprint2c, R17
10	#19-Mon 4/6 or Tue 4/7		sections 1-3	Draft2 due, sprint more, R18, **no R19**
	#20-Wed 4/8 or Thu 4/9			Comments given, sprint more, R20
	#21-Mon 4/13 or Tue 4/14	res. paper	individ. meetings	sprint more, R21
11	#22-Wed 4/15 or Thu 4/16		individ. meetings	sprint more, R22
	Mon4/20Holiday:no class cs410-01			sprint more, no R23
12	#23-Tue 4/21			sprint more, R24
	#24-Wed 4/22 or Thu 4/23			Research paper due
	Thu4/23pass/fail/withdr. deadline			
13	#25-Mon 4/27 or Tue 4/28			sprint more, schedule client validation, R25
	#26-Wed 4/29 or Thu 4/30			sprint more, slides due, R26
14	#27-Mon 5/4 or Tue 5/5			Presentations
	#28-Wed 5/6 or Thu 5/7			Presentations
15	#29-Mon 5/11 or Tue 5/12		last class -02	Presentations and Demos
	#30-Wed 5/13		last class -01	special topic

Additional references:
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

Revisions: Evaluation point totals clarified. Wording was updated in topics, evaluation and schedule. Links corrected Reposted 2/2/2026; v6: typo corrected, date to submit hw1 corrected, reported 10feb2026.

Table of score to grade conversions (default grading scheme "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