CS 410-02: Introduction to Software Engineering

FALL 2026 26 sep 2026 Rev2

Dr. J. Holly DeBlois

Office: McCormack, 3rd floor, room M-3-201-32, phone 617 287-6490

Office hours: Tue and Thurs 11:00-12:30pm and by appt

Lectures and Class:	Tues & Thurs, 4:00-5:15pm, Wheatley W01-0004 (course 3686)
Instructor Email:	jane.deblois@umb.edu
Instructor Website:	Lecture notes and assignments: https://www.cs.umb.edu/~hdeblois/cs410/f26/
Grades:	Grades are on canvas: https://www.umb.edu/canvas/
Grader:	Andrew Truong
Portal:	Register for CS410 to create your course directory at https://portal.cs.umb.edu
Piazza:	Join Piazza: https://piazza.com/umb/fall2026/cs41002

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 as teams of 5-7 students develop a long project for a client.

Pre-requisites: CS 310 Algorithms, CS220/CS320L Discrete Math and any CS400 level course, or permission of the instructor. We offer optional hw0 to review CS Server skills.

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. **If you miss >5 classes, your final grade will be lower.** If you miss 6 or 7, you lose 1 letter grade step (e.g., B to B-). If you miss >7, you lose 2 letter grade steps (e.g. B to C+). Because you are allowed to miss 5 classes, requests to be excused from class are rarely needed. If you need to be excused more, you must submit documentation. See https://www.umb.edu/registrar/policies/attendance/. Being late to class or leaving class for more than a few minutes will lower your participation points. See Participation, p3.

Course Objectives and Outcomes: We cover the following general topics:

- **Writing code for a client as part of a team of software engineers –** requires new skills, tell us your aims in your starter resume, learn to define what you like while working with a team
- **Requirements specification** – translate what the client wants into various requirements, identify buildable subsets, split them up so team members have different tasks
- **Websites on the CS server** – draw and build singly and in teams on the CS servers to practice delivering a small website product, singly and then for a team
- **Git, github and version control** – be able to use git and github to collaborate, branches for individuals and branches for team new features, team testing and team presenting to client

- **Use Minnie** to understand front-end and back-end, build both, test a complete function for the client
- **Follow agile methods, set up sprints and code in class –** limit time spent talking, discern ready to build items for a sprint, design, build, test and report or do knowledge acquisition
- **Test-driven code –** save time by identifying tests first, using them to drive what and how you build; tests must be repeatable by a test team (they cannot be on your localhost/local machine)
- **Ask for a CS server location** and deploy your team's product there first
- **Understand and document full stack of your product** so do not use containers or proprietary software that require login and payments; after your features work on the CS server, if client requests use of proprietary software, request instructor permission to use software that is not open-source
- **Artifacts –** store the your part of team work items that are not code as artifacts – such as list of product backlog not yet implemented and test documents that you create separate from code
- **Presentations –** present single slides several times to practice presenting; present three slides for your own work on the long project and add it to your resume
- **Demo the part of the project you worked on –** speak to the class about your contribution(s)

Bring Suitable Computer and Pencil to Class: You must bring a computer to class each day that is capable of accessing your CS course directory and doing in-class assignments there, which will be script-collected and graded. For handwritten work, bring a pencil or a pen. A page of paper with the assignment will be provided. Fill in your name and student id as it is in Wiser. Handwritten pages will be scanned into gradescope/canvas. We write tests and code, and compile and run in class. If you do not have a suitable laptop, contact your advisors or you will receive scores of zero on in-class work.

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

Access the CS Servers: We offer optional ungraded hw0 to assure that you can work on the CS servers. Be careful to 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>).

AI Policy. This course DOES allow the use of AI but it requires listing prompts and dates, since this is AI use in an academic setting. In addition, since you are required to understand the work you submit, some handwritten assessments will be given. Code submitted to github must be identified as yours by the branch, your github name and date. You are not allowed to ask AI to submit code to github for you.

Provost Policy on AI. AI is allowed with attribution: Use of AI tools, including ChatGPT, is permitted in this course on certain assignments [either detail assignments or types of assignments here, or clarify how they will know which ones! **See above**.] To adhere to our scholarly values and to the Student Code of Conduct, students must cite any AI-generated material that informed their work; citations should include not only in-text citations and listing in the references, but also the full text of cited ChatGPT (or other Large Language Model (LLM) generator) as an appendix to the assignment. Using an AI tool to generate content without proper attribution qualifies as academic dishonesty. Students are also responsible for making sure that any AI generated text does not contain false or erroneous information. If students are unsure about whether or not a source is appropriate to use in the assignment, they should contact the instructor.

Grading: Your score in the course is a weighted average of assignments in 7 groups, defined here and listed in canvas. In each group, your score is a fraction, the sum of points earned divided by the total points on all assignments in the group. See the table for conversion of score to grade at end of syllabus. Here are the groups and weights:

Startup (11%);
Initial projects (12%);
Homework (25%);
Long project work on team (25%);
Finishing up long project work (15%);
Participation (2%);
Attendance depending upon whether you miss more than 5 classes,
explained above (10%).

Daily reports cover scrum meeting, what you designed, tested and 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 in a manner that allows others to verify your test results. Assignments may change if necessary. Students are graded on the quality of their work on requirements, code and artifacts as part of the team's work.

Academic references are required for your team project. You are required to check any advice given casually, by a fellow student or by an AI agent. You must understand citing sources correctly. Appreciating acclaimed academic work is important to the success of any tech endeavor.

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 as a team are required and each student must speak about their contribution.

Participation is based on time in class and contributions. The instructor may deduct participation points if you are late, leave early or leave the classroom for more than a few minutes. It is participation to be sure the instructor and your classmates know your name, interests and discoveries. Please say your name ahead of asking or answering questions or requesting or giving help to others. We use 1 pt per 10 minutes on time in class so you earn 8-9 points per session minus the deduction (sorry to be so persnickety but you are in class to code and test work so time matters to you and others).

Attendance is recorded on the main clipboard, but you have 5 missed classes allowed.

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. The max points lost is 50% of the total.** Score remains 0 unless you resubmit and earn points which are then reduced by the max late penalty. 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 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.

Student Assistance for Health, Well-being and Success:

- Are you in emotional distress? Call 617.287.5690 to speak with a licensed clinician 24/7 who can offer support, crisis recommendations, and assistance with finding resources.
- Have a campus question or issue? Use Here4U in the UMass Boston app or via www.umb.edu/here4U.
- Want advice in navigating a university or life situation? Contact the Dean of Students Office at www.umb.edu/deanofstudents.
- Want to connect with housing and food insecurity support, student life groups and events, or recreation activities? Visit Campus Life - UMass Boston.
- Want to access resources specifically for immigrant-origin, DACA, TPS, and undocumented students? Visit Immigrant Student Programs - UMass Boston.
- Want to make the most of your academic experience? Visit Student Equity, Access & Success.

- Looking for additional identity-based community support? Find more resources at Inclusive Identity - UMass Boston.
- Unable to attend class on a specific date or participate in an exam or class requirement due to a religious observance? Fill out the excused absence form (requires 2-weeks' notice) to request religious accommodation at Right To Excused Absence Because Religious Belief - UMass Boston

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 (27 classes over 14 weeks):

Week	class#/date	reading	topic	assignment/homework posted More report numbers due will be posted
1	#1-Tue Sep 8	Syllabus, PG ch1-2	introduction, agile principles	Syllabus, in-class draw1, hw0
	#2-Thu Sep 10	ES ch2-3, Fig 1-3 p7	hw1, websiteproj	Post hw1, login, w, website part1, tests, public_html, index.html, report R01
2	#3-Tue Sep 15 **Tue Sep 15 add/drop ends**	ES ch4-5 PG ch3-4	team website proj, artifacts, permissions	Website part2, html, group cs410-02, cp -r <file>, nnnn.txt, R02
	#4-Thu Sep 17	ES ch6-7 PG ch5-6	scrum roles, website design, demo the proj	Website part3, nnnn.txt symlinks, software as dir for website pointed to by https://www.cs.umb.edu/cs410, R03, hw1 due
--	--	--	--	--
3	#5-Tue Sep 22	ES ch8-9	agile principles, sprint review for website proj, backend/frontend initproj1	Post hw2, sprint review, Minnie, R04, Celtic Knot, git local, presentations of websites and resumes
	#6-Thu Sep 24	ES ch10-11	initproj2	Client presentations, emacs, passwordless login, product backlog, R06, hw2 posted including the requirement for R05
4	#7-Tue Sep 29	ES ch19-20	initproj2 sprint review, long proj preferences submitted	initial team proj, R07
	Wed			hw2 due long proj teams assigned

--	--	--	--	--
	#8-Thu Oct 1	client zoom	longproj sprint1, prodbacklog, artifact, roles	Teams meet, long proj start, github, artifact, R08
5	#9-Tue Oct 6	client zoom	longproj backlog	discuss code on github by student, R09
	#10-Thu Oct 8	acad. refs reqts list	sprint1 review, presentations	present, sprint2, all teams, R10
6	**Mon Oct 12 Indigenous Peoples Day**			
	#11-Tue Oct 13	git review deploy loc.	use Healey Lib sprint2 starts	sprint2, R11
	#12-Thu Oct 15	Client zoom	artifacts team reviews	post hw3, R12
7	#13-Tue Oct 20			sprint more, R10
	#14-Thu Oct 22			presentations
8	#15-Tue Oct 27			sprint more, R11
	#16-Thu Oct 29			sprint more, R12
--	--	--	--	--
9	#17-Tue Nov 3			sprint more, R13
	#18-Thu Nov 5			presentations
10	#19-Tue Nov 10			sprint more, R14, schedule client validation
	Wed Nov 11 Veterans Day			
	#20-Thu Nov 12			hw3 due, sprint more, R15
11	#21-Tue Nov 17			sprint more, R16
	#22-Thu Nov 19			sprint more, R17
	Pass/Fail/Withdrw Deadline Nov 19			
12	#23-Tue Nov 24			
	HolidayThu Nov26-29			
--	--	--	--	--
13	**Classes resume Nov 30 Mon**			
	#24-Tue Dec 1			Presentation slides, 3 per student, due R18
	#25-Thu Dec 3			Presentations
14	#26-Tue Dec 8			Presentations and Demos
	#27-Thu Dec 10		last class	special topic
	Semester Ends Dec 11 Fri			

	14 weeks, 1 holiday affecting class, so 27 class meetings			

Additional references:
www.w3schools.com for html and javascript

Revisions: Rev1: fix weighted attendance % error, add student assistance and revise schedule
Rev2: set max late penalty 50%, revised schedule of reports and hw2

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