ENSE 281. Software Engineering Management. Winter 2026.

Instructors:

- **Lecture**: Dr. Tim Maciag | tim.maciag@uregina.ca | https://www.maciag.ca/
    - **Tim's office hours**: Wednesdays, 11:30 am – 1:00 pm (in ED 410) or by appointment
- **Lab**: Adam Tilson | adam.tilson@uregina.ca

Lectures:

- Wednesdays | 2:30 pm – 5:15 pm | ED 616.

Labs:

- Mondays (091): 2:30 pm – 5:15 pm | or | Tuesdays (092): 2:30 pm – 5:15 pm.
- **NOTE**: Both lab days/times offer hyflex options: ED 485.1 (in-person) and synchronous remote (Zoom).

Course description

Principles of software engineering: requirements, design, testing, and maintenance. Review of principles of object orientation and analysis using UML. Frameworks and APIs. Introduction to the client-server architecture. Analysis, design and programming of simple servers and clients. Introduction to user interface technology.

Texts & resources

Course content will be loosely based on the following texts. Although you don't need to buy these, they are good to have on your (digital) bookshelf if you are serious about software engineering (software design & development). Most of these textbooks are available in O'Reilly Books Online (as of January 5, 2026, see: "**"). Log in via your URegina credentials.

- **A. Hunt and D. Thomas. *The Pragmatic Programmer, First & Second Edition*. Addison-Wesley. 2000/2019.
- D.C. Barrett. *Understanding Project Management, Third Edition*. Canadian Scholars' Press. 2024.
- **R.C. Martin. *Clean Code*. Pearson Education. 2009.
- **S. McConnell's. *Code Complete, Second Edition*. Microsoft Press. 2004.

Course evaluation

Lab work (individual)	Project work (team)	Mid-point assessment	End-point assessment	Professionalism, quizzes, & participation	Instructor discretion
25%	25%	20%	20%	10%	+/- 5%

Important notes

Class structure & minimum requirements

- The course instructor will facilitate an in-person education and learning environment.
- Students should have access to a laptop or portable technology (mobile phone, iPad, etc.) as there will be significant in-class active learning and exploration.

High-level information about all education & learning activities in the class & lab

- Professionalism, attendance, and participation will be observed in all class activities. These will be monitored and represented in the "Professionalism, quizzes, & participation" grade allocation.
- Late submissions to all coursework will not be accepted and will be graded as a non-submission.

Labs:

- The lab instructor will facilitate a “hyflex” synchronous remote and/or in-person format/approach to education and learning. For labs, students may attend in person or remotely via Zoom at the specified lab times.
- Students must obtain at least 60% in the lab. If they fail to achieve this benchmark grade, they will receive a 0/25% for their lab grade allocation.

Course project:

- Participation in the course project is mandatory. If students fail to participate in the course project, the student(s) will receive an NP in the course. Participation will be evaluated based on project insights (via GitHub), information obtained in team/instructor conversations (scrums), and/or on self/peer reviews for all team-based project work (and all as per the instructor’s discretion).
- Throughout the semester, team/instructor “scrums” (meetings) and dedicated “baked-in” project work time will occur. Attendance at these events is mandatory, and non-attendance may result in a grade of 0 for the specific project deliverable. Also, tardiness will be observed, and individual deductions will apply upward to the complete specific project deliverable allocation (as per the instructor’s discretion).
- All students must attend (and be on time) for the customer scrums and final project presentations at the end of the semester. Students who fail to attend and be on time for all final presentations will receive a 0% on their project allocation (0/25%).

Mid-point & end-point assessments:

- Deferred assessment requests will not be accepted. Assessment dates are set per the course schedule and are non-negotiable (except in extreme circumstances, at the instructor’s discretion).
- Students who have approved and agreed upon accessibility needs/requirements related to exam-type educational activities and who need more time and/or a quiet space to write knowledge checks will be required to arrange things with the University of Regina Accommodations Test Centre (ATC) and the instructor. ATC-approved students will be responsible for ensuring proper communication with the ATC and the instructor(s) to ensure the necessary preparations have been made at least 120 hours (5 days) before the pre-scheduled assessment (exam/test) day. Students who fail to provide said preparations must write assessments as scheduled.
- Students must achieve an overall grade of 60% or higher by combining both assessments (mid- and end-point). If students fail to achieve a grade higher than 60%, as indicated, they will receive a 0% for their cumulative observed assessment grade allocation.

Lectures:

- Students should bring a laptop or portable technology (mobile phone, iPad, etc.) to lectures, as there will be significant in-class active learning activities and exploration.
- Lectures will be in person and consist of lecture content with embedded “knowledge checks” to help students reinforce their understanding of key lecture topics and discussions. Student participation in knowledge checks will be included in their “Professionalism, quizzes, & participation” grade allocation.
- Students are expected to attend lectures on time. Attendance and course participation will be graded in the “Professionalism, quizzes, & participation” grade allocation.

- Course materials are provided exclusively for students' use. They must not be uploaded to 'homework helper' platforms such as CHEGG, CourseHero, etc., nor to large language models (LLMs) like NotebookLM, ChatGPT, etc.

A note on professionalism, use of Generative Artificial Intelligence (GenAI) tools and the like, and academic misconduct:

- It is expected that students will be pragmatic and professional in all course communication and course learning activities (individual and team). The instructor reserves the right to apply a +/- 5% discretionary grade to all individual students as per the course grading scheme, based on the instructor's perceived pragmatism and professionalism.
- As mentioned under "Lectures," all course materials are provided exclusively for students' use. Provided lecture (and lab) content must not be uploaded to 'homework helper' platforms such as CHEGG, CourseHero, etc., nor to large language models (LLMs) like NotebookLM, ChatGPT, etc. Such actions constitute a violation of copyright and will be regarded as an act of academic misconduct.
- The instructor(s) reserve the right to interview students regarding their understanding of submitted works, and to alter individual grades accordingly (as per the instructor's discretion). In the perceived event that a student's submitted work appears to have been copied from a solution manual, classmates, or online sites such as Stack Overflow, CHEGG, Course Hero, etc., the student will receive a zero on the submitted work and be flagged to the Associate Dean of Academic for a warning letter. If the behaviour is repeated, the student will be flagged to the Associate Dean Academic for a formal academic misconduct investigation, which will be documented in the student's official university student file.
- Tools like ChatGPT, CoPilot, and open-source options such as PaLM+RLHF could transform software engineering and computer science. While independent learning is encouraged, students using these tools to enhance their knowledge must remain transparent and demonstrate professionalism. Students, individually or in teams, must justify their design and development choices based on all course activities, including lectures, labs, blogs, discussions, and more. Transparency and a sound rationale are required; simply claiming that a choice was generated by such tools is not sufficient and may lead to penalties or consultations with the Undergraduate Dean. Maintaining transparency is crucial.
- **More Tim being all *introspectivy* and such**: This new world feels strange and confusing (at least to me, though others might feel the same). These tools are available in their current form—they're interesting but not perfect. Still, they are likely to get better over time. The world may expect you to learn how to use them effectively. I'm fine with people exploring these tools to enhance their knowledge, but it's also okay if you feel cautious (which is preferred, quite frankly). Whether alone or in a team, you should reflect on your experiences and have honest, open conversations, focusing on transparency.

Other information:

- **Grading** (As per the academic calendar)

90—100: Outstanding	80—89: Very good	70—79: Above average
60—69: Satisfactory	50—59: Barely acceptable	0—49: Unacceptable

- **Student behaviour**: https://www.uregina.ca/university-leadership/governance/student-appeals.html
- **Health & safety**: https://www.uregina.ca/health-safety/index.html
- **Student Accessibility**: https://www.uregina.ca/accessibility/student/index.html

Course schedule

All topics/dates are tentative/subject to change minus assessment/exam dates

[1] A. Hunt and D. Thomas. *The Pragmatic Programmer, First & Second ed*. Addison-Wesley. 2000 & 2019
[2] S. McConnell's. *Code Complete, Second Edition*. Microsoft Press. 2004
[3] R.C. Martin. *Clean Code*. Pearson Education. 2009
[4] D.C. Barrett. *Understanding Project Management, Third ed*. Canadian Scholars' Press. 2024

Week #	Date	Lecture topics	Resources/Notes	Lab topic
Week 1	Jan 7	Course introduction; Syllabus overview; Teaching philosophy; Q/A		N/A
		A pragmatic philosophy for pragmatic programming	[1] Chapter 1 [2] Chapter 33—34	
Week 2	Jan 14	Understanding the project environment; starting the project; Start with Why	[4] Chapter 1—2	Tools
		The planning phase; the executing phase	[4] Chapter 3—14	
Week 3	Jan 21	Pragmatic team projects; Project discussion; Team introductions; "Teams that care" game; Project work session (ideation/activity 1)		View 1
Week 4	Jan 28	Activity 1: Project pitch, conversation, and, if needed, iteration		View 2
Week 5	Feb 4	Specifics to software design, architecture, & construction	[1] Chapter 3 [2] Chapter 1—5, 30, 34 Sutherland & Schwaber: Scrum guides	View 3
		Picturing architecture & design	[1] Chapter 3, 7 Allen Holub: website	
Week 6	Feb 11	Mid-point assessment		View 4
Week 7	Feb 18	Winter break, no class		N/A
Week 8	Feb 25	Model View Controller (MVC) design architecture; Intro to design patterns; Picturing architecture review; Activity 2 overview	[1] Chapter 2, 5 Sourcemaking website	Controller 1
		Activity 2 project work time		

Week 9	Mar 4	Activity 2 scrum/report-out: Requirements, system architecture, documentation, and plan		Controller 2
Week 10	Mar 11	Creating quality variables, functions, and classes	[2] Chapter 10—13 [3] Chapter 1—2	Model 1
		Creating flexible code	[1] Chapter 2—6 [2] Chapter 6—7, 14—17 [3] Chapter 6, 10, 13	
Week 11	Mar 18	Activity 3 scrum/check-in: Construction/coding progress check-in with project work time		Model 2
Week 12	Mar 25	Continuous software improvement; Pragmatic Paranoia	[1] Chapter 3, 6, 8 [2] Chapter 20—22, 28	N/A
		Documentation; Other System Considerations	[2] Chapter 27—28 [3] Chapter 4	
Week 13	Apr 1	End-point assessment		N/A
Week 14	Apr 8	Activity 4: Final presentation & project delivery		N/A
Week 15+	Apr 22	*There is no "final" exam*		