SOEN 6011 Software Engineering Processes

Course Outline

Some Logistics

- **Instructor:** Joumana Dargham joumana.dargham@concordia.ca (H-961-25)
- **TAs**:
  Jimi Mehta: jimi.mehta@concordia.ca
  Hamed Jafarpour: hamed.jafarpour@concordia.ca
  Mina Masoumi: minamasoomi31@gmail.com
- **Lectures: MoWe** — 18:30-21:00 (FG C070 SGW)
- **Office Hours :** joumana.dargham@concordia.ca.

Why study Software Engineering?

Design, code, and test software products

Software engineering concerns

Study	Study	Create	Learn
Study the life cycle of software products from specification through analysis and design, to testing maintenance and evaluation	Study the range of paradigms practised by software developers	Create professional-quality software systems with professional techniques and tools	Learn to balance large-scale product development, with safety, reliability, cost and scheduling

What do I do to survive?

SOEN6011: Software processes

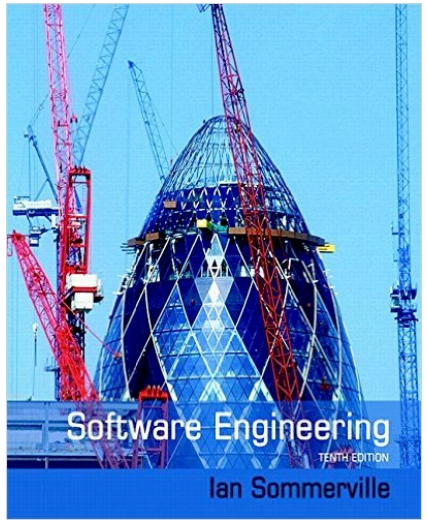
- This is **NOT** a programming course!
- Learn and follow the software development process.
- Manage and track your project: **GitHub** for EVERYTHING.
- Team work.

What are we doing here?

Learning about the process of design, development, and testing of software systems:

(1) Requirements analysis
(2) Software architecture and design
(3) Implementation
(4) Integration
(5) Test planning
(6) Software maintenance

Reference Book



The textbook is recommended but NOT REQUIRED. If you have an older version, it should also be fine.

Part 1 Introduction to Software Engineering

1. Chapter 1 Introduction
2. Chapter 2 Software processes
3. Chapter 3 Agile software development
4. Chapter 4 Requirements engineering
5. Chapter 5 System modeling
6. Chapter 6 Architectural design
7. Chapter 7 Design and implementation
8. Chapter 8 Software testing
9. Chapter 9 Software evolution

How will I be evaluated?

Overall evaluation scheme

- 2 Quizzes (40%, 20% each)
- 1 final exam (30%)
- Project (25%)
- Project contribution and individual project report (5%)

Evaluations tentative dates?

- Quizzes (20%*2), Exam (30%)

- Tentative dates:
  - Jul. 12th (Quiz 1)
  - Jul. 26th (Quiz 2)
  - August. 7th (Exam)

- You MUST receive (35/60) on the quizzes and exam combined to pass the course

How will I be evaluated?

- Team project (a total of 25%):
  - Term-long SE project (20%)
  - Project presentation, Final project deliverable 5%
- Project contribution and individual project report (5%)
  - You will document your contributions and what you have learned in the project

Sprints	Due date	Weight
1st sprint	Jul. 10th	5%
2nd sprint	Jul. 19th	5%
3rd sprint	Jul. 31st	5%
4th (final) sprint	Aug. 9th	5%

What is the project about?

Project topic

- You will implement a simplified version of a **Career service platform**
- Core features required:
    1. Browsing for available postings
    2. Adding and manage postings to the System by the Employers
    3. Candidates can apply to Employers job postings and get informed by the Employer if candidate gets selected
    4. Creating and Managing Student Profile
- More features are needed.
- Your TA is your customer.
- Propose your features to your TA
- You can use any programming language and framework

Project Schedule and Team

- Form the team yourself
- Join groups on moodle: groups should be of 6 members.
- Once the groups are finalized, all members should get together for an introduction.
- Give a name to your group

Managing your project

- **Manage everything on GitHub.**
- Continuous delivery is required (more to come later in the course).
- Team lead needs to be confirmed in the second tutorial.
- TA will give a short lecture and hand-on session on Git and GitHub during POD sessions.

Using GitHub

- All the discussion, all the documentation need to be found on GitHub.
    - Discussion offline or on other platform needs to be documented on GitHub.
- Code review is needed.
    - Someone else in your team need to read your code before committing.
- Everything needs to be referenced.

Project Schedule and Team

- The entire project is due in 4 sprints.
- Each sprint needs concrete tasks finished, features implemented or bugs fixed.

Focus of each sprint

- Sprint 1: Setting up the running environment.
- Sprint 2: Core features implementation. Setting up testing infrastructure and continuous integration infrastructure.
- Sprint 3: More features with TA feedback.
- Sprint 4: Bug fixing, testing and wrapping up.

Final delivery

- Documentation:
  - Based on the material on GitHub.
  - Mismatching information between GitHub and the final documentation will not be considered and final delivery mark will be punished.

- Source code:
  - The source code snapshot of the release will be evaluated.

Final project presentation

- Presenting what you have done in your project
- Tentative presentation date: final week of the lectures
    - More details later