Concordia University
Department of Computer Science and Software Engineering
SOEN 341 — Software Process
Course Outline — Winter 2023 — Section CC

Instructor: Joumana Dargham	MoWe 18:30-21:00 FG C070 SGW
TA: Jimi Mehta: jimi.mehta@concordia.ca Hamed Jafarpour: hamed.jafarpour@concordia.ca Mina Masoumi: minamasoomi31@gmail.com	

Introduction

The purpose of the present offering of SOEN 6011 is to explore theoretical foundations and practical environments of the discipline of software engineering in preparation for an alignment with the CMMI Level 5 and to adopt a practice-based approach to learning about software processes.

Prerequisite Knowledge

You should have had some previous experience in programming, Web Programming, some knowledge of Software testing, and some knowledge of the principles of technical documentation.

Course Description

Introduction to software engineering concepts, modern management processes and software lifecycles. Tracking of software requirements and development through issue trackers. Version control practices include integration of feature branching. Validation practices including testing frameworks and code review. Continuous integration and build environments. A practical component on setting up a development environment similar to those used in professional environments. Case studies of successful management processes from open-source projects and industry. A project is required.

Textbooks

We will use the following book as the **recommended** textbook throughout the course:

- Ian Sommerville, *Software Engineering*. 10th edition. Pearson, 2015.

Other references:

- Mike Cohn, Agile Estimating and Planning. 1st edition. Prentice Hall, 2005.
- Dean Leffingwell, Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise. 1st edition, Addison Wesley Professional, 2011.

Course Moodle WebSite

Amendments to this syllabus, if any, as well as other important information, will be made available through the course's Moodle site. The Moodle site also provides additional reading material, the project, quizzes, etc.

Project

The major practical component of the course will be the development of software applications designed and implemented by a team. Each team will consist of approximately 5 and will implement an application whose description will be provided during the lectures. It is very important to understand that the role of this project is to demonstrate the importance and the application of a software process. It is not to be approached as a regular programming project. You may implement it using any language or scripting language, or any tools that may simplify or accelerate your work. All members of your team must have a good understanding of the tools used. Your documentation, both internal and external, must be clear enough to be easily understandable by a reader who may not be familiar with the language or tools used.

Evaluation

The evaluation scheme is as follows:

- Quizzes 40% (2 quizzes, 20% each)
- In-class exam 30%
- Team Project Sprint 1 5%
- Team Project Sprint 2 5%
- Team Project Sprint 3 5%
- Team Project Sprint 4 5%
- Team Project Presentation and final report (complete report integrating all sprints of the project, demonstration of the software, and quality of application) 5%
- Project contribution and individual project report 5%

There is no standard relationship between percentages and letter grades assigned for this course. In order to pass the course, you must **receive at least 50% of the overall possible marks and 50% of the quizzes and in-class exam**. There are no marked assignments in this course. Should you fail to write one of the quizzes and you have valid justification (e.g., doctor's note) then the weight of that quiz will be added to the other assessments. Note: It is your responsibility to adhere to the university’s code of conduct as detailed in the calendar.

Course Content and Schedule

We will cover the following topic(s) in each week (note that the weekly schedule may be subject to change):

Week 1: Introduction, Software processes
Week 1: Agile software process
Week 2: Software requirements
Week 2: Software engineering project planning
Week 3: Software design
Week 3: Software architecture and detailed design
Week 4: Software design and diagrams
Week 4: Design patterns
Week 5: Implementation
Week 5: Software quality assurance
Week 6: Software testing
Week 6: Release engineering
Week 6: Project presentations

There will not be time to cover all of these topics in depth. Other topics of interest to the class may also be included. More details for each topic will be made available on the Moodle website.

Graduate Attributes

As part of the Software Engineering program curriculum, the content of this course includes material and exercises related to the teaching and evaluation of graduate attributes. Graduate attributes are skills that have been identified by the Canadian Engineering Accreditation Board (CEAB) and the Canadian Information Processing Society (CIPS) as being central to the formation of Engineers, computer scientists and information technology professionals. As such, the accreditation criteria for the Software Engineering and Computer Science programs dictate that graduate attributes are taught and evaluated as part of the courses. The following is the list of graduate attributes covered in this course, along with a description of how these attributes are incorporated in the course.

Magnitude	Attributes coverage description
●●	Attribute 1: Knowledge-base: Knowledge of basic principles of software engineering. Introduction to software process models. Activities in each phase, including review activities. Knowledge related to team dynamics: organization; stages of formation; roles; conflict resolution. Notations used in software documentation. How to review, revise, and improve software documentation.
●●	Attribute 2: Problem analysis: Knowledge and use of various activities, techniques and tools to develop project analysis artifacts in software development.

●●	Attribute 4: Design: Knowledge and use of various activities, techniques and tools to

	develop various design artifacts in software development.
●●	Attribute 5: Use of Engineering tools: Use of appropriate tools for the preparation of various documentation artifacts, especially diagrams.
●●●	Attribute 6: Individual and team work: The course project is such that it needs to be tackled in teams. Students will cover a complete project development lifecycle, with project responsibilities rotating throughout the project. Project tasks will be assigned throughout the project to individual team members, which will require periodic integration of individual tasks into the common project context.
●●●	Attribute 7: Communication skills: Use of documentation templates, diagrams, standards, and software tools to develop various software development documents. Oral project deliveries.
●●	Attribute 11: Economics and project management: Project management and human resources management issues, including organization, roles, and conflict resolution.

Disclaimer

In the event of extraordinary circumstances beyond the University's control, the content and/or evaluation scheme in this course is subject to change.