McGill University
School of Computer Science
www.cs.mcgill.ca

Course Outline

Course Name: Software Engineering Project
COMP-361D Fall 2012 & Winter 2013

Instructor: Joseph Vybihal

Contact Information: Joseph Vybihal
Office: ENGMC 323
Office Hours: TBD (in my office or by appointment)
TBD (In Trottier 3rd floor labs - look for me)
Email: Use the WebCT email tool
Emergencies or non-course related: jvybihal@cs.mcgill.ca

Course Objectives: This 1-year course will give students the opportunity to develop a considerable software system following a rigorous software development process. Over two semesters, students will develop a system that solves a set of requirements of practical relevance. The project will take students through all the phases of a software’s life-cycle, from inception to final deployment and release. Throughout the project, students will follow a defined process involving software engineering practices from a variety of disciplines, including: preliminary design, requirements gathering by interacting with a client, requirements modelling and specification, requirements analysis, software design, implementation, testing, configuration and change management, and project management. Project deliverables, documentation, presentations and demonstrations.

The students are to select one project from the following three domains: In-House Software Development Project, Contract Programming Project, Research & Development Project. The In-House Project is a game, an ideal project for students in the games stream. The Contract Programming Project will get you to work with a real-world customer in a business context. The Research & Development Project will give you the opportunity to work with a professor on one of their research interests (this could be one of the teachers of the course or some other professor in the university, as long as it meets all the requirements of the course).

Specific Objectives: At the end of the project, students will have learned how to conduct a software engineering project by following a defined process. They will know and understand the different activities pertinent to the different phases of the process. They will have acquired practical software development experience with a significant project and acquired or refined the following abilities:

- Effective team work and team management;
- Technical communication, both written and oral;
- Interacting with customers for requirement elicitation;
- Analyzing and designing software;
- Testing software for regression and acceptance;
- Working with software engineering tools;
- Managing software change;
- Assuring software quality;
- Task scheduling and effort estimation.

Course Description: Software development process in practise: requirement elicitation and analysis, software design, implementation, integration, test planning, and maintenance. Application of the core concepts and techniques through the realization of a large software system.

Recommended Readings:
(5) UML Specification (available for download from the OMG website)

Evaluation:
First Semester:
Preliminary Design Doc .................... 10%
Requirements Doc ............................ 10%
Design Documentation ...................... 10%
Class Introduction Presentation ........ 5%
Exam Term 1 ................................. 20%

Second Semester:
Acceptance Demo & Management .... 15%
Final Application Test ...................... 25%
Class Presentations ............................ 5%

Tutorials: Using various software tools.

You will be notified in advance of project hand-in due dates. All hand-ins are due on WebCT at the indicated time and date. Late hand-ins will lose 5% of its grade per day late. You may not submit hand-ins via e-mail without the permission of the instructor. Grades for the project are per team, grades for the two exams are individual. Exams will be a combination of all types of questions based on the material covered in class and on the experience gained during the project.

No make-up tests or make-up assignments are allowed in this course.

If you are not satisfied with the grading of a hand-in or exam, you may request a review within 7 days of return. Indicate in writing or during a meeting with the instructor where and why you feel the marks are unjustified and give it back to your instructor for re-grading. Note that the entire assignment or mid-term test will be re-graded and your grade can go up or down (or stay the same) accordingly.

Course Requirements: Prerequisites: COMP 206, COMP 250
Co-requisite: COMP 303

Calculators
Only non-programmable, no-tape, noiseless calculators are permitted. Calculators capable of storing text are not permitted in tests and examinations.

Dictionaries
Dictionaries are not permitted, but translation dictionaries are.
Handheld Devices
Handheld devices capable of storing text and having calculator functionality (e.g. Palm, etc.) are not permitted.

Additional Information: The course slides are not meant as a complete set of notes or a substitute for a textbook, but simply constitute the focus of the lecture. Important gaps are left in the slides that are filled in during class, thus lecture attendance should be considered essential.

The material covered in the classroom will be used to supplement textbook readings.

Academic Integrity: Code of Student Conduct
McGill University values academic integrity. Therefore all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures (see www.mcgill.ca/integrity for more information).

L'université McGill attache une haute importance à l'honnêteté académique. Il incombe par conséquent à tous les étudiants de comprendre ce que l'on entend par tricherie, plagiat et autres infractions académiques, ainsi que les conséquences que peuvent avoir de telles actions, selon le Code de conduite de l'étudiant et des procédures disciplinaires (pour de plus amples reseignements, veuillez consulter le site www.mcgill.ca/integrity).

Right to submit in English or French written work that is to be graded
In accord with McGill University’s Charter of Students’ Rights, students in this course have the right to submit in English or in French any written work that is to be graded.

Final Exam Policy: Regulations
Students should not make other commitments during the final exam period. Vacation plans do not constitute valid grounds for the deferral or the rescheduling of examinations. See the Centre Calendar for the regulations governing Examinations: http://www.mcgill.ca/student-records/exams/regulations/

Students are required to present their I.D. Card (with photo) for entrance to their examination.

Conflicts
If you are unable to write your final examination due to scheduling conflicts, you must submit a Final Exam Conflict Form with supporting documentation at least one month before the start of the final examination period. Late submissions will not be accepted. For details, see http://www.mcgill.ca/student-records/exams/conflicts/

Exam Timetable
Examination schedules are posted at the Centre and on the following page approximately 6-8 weeks before the examination period commences http://www.mcgill.ca/student-records/exams/

The Centre cannot provide examination dates over the telephone.

Classroom Rules:
All electronic devices (cell phones and beepers) must be turned off or left on silent mode during class time.
Assignments Pickup: All hand-ins are submitted to and picked-up from WebCT.

Email Policy: E-mail is one of the official means of communication between McGill University and its students. As with all official University communications, it is the student's responsibility to ensure that time-critical e-mail is accessed, read, and acted upon in a timely fashion. If a student chooses to forward University e-mail to another e-mail mailbox, it is that student's responsibility to ensure that the alternate account is viable. Please note that to protect the privacy of the students, the University will only reply to the students on their McGill e-mail account.

Computing Resources: Trottier 3rd floor.

Students Rights and Responsibilities: Regulations and policies governing students at McGill University can be downloaded from the website: http://www.mcgill.ca/deanofstudents/rights/

Students Services and Resources: Various services and resources, such as email access, walksafe, library access, etc., are available to students: http://www.mcgill.ca/student-records

Minerva for Students: http://www.mcgill.ca/minerva-students/

Note: In the event of extraordinary circumstances beyond the University’s control, the content and/or evaluation scheme in this course is subject to change.
Detailed Course Outline: Tentative Schedule

September

- **Goal:**
  - Introduction to the course & The Preliminary Design Document

- **Due:**
  - Team and Project Selection (2nd week) – Teams will be listed in MyCourses for all to see.
  - Preliminary Design, Base Case, and Fundamental Requirements (October 1st) - Technology selection will be added to Teams list in MyCourses for all to see.

- **Lectures:**
  - **Week 1**
    - Day 1: Why Software Engineering?, About this course, our Course Project, My Research
    - Day 2: Software Life Cycle, About software projects, Project Life Cycle, Project Types
  - **Week 2**
    - Day 1: RUP and Scrum, Extreme and Open Source
    - Day 2: Team environments and Team Dynamics
    - Day 3: (overflow time)
  - **Week 3**
    - Preliminary Design Document
      - Day 1: Basic Format and Stakeholder identification and fundamental satisfaction list
      - Day 2: Technology Selection
      - Day 3: Baseline
  - **Week 4**
    - Time, Cost and Risk Estimation
      - Day 1: Time – Phase breakdown, duration activity graph, dated activity graph (calendar spaces)
      - Day 2: Cost – Step 1: Duration activity graph, Step 2: Theoretical estimates, Step 3: Conclusion
      - Day 3: Risk Identification and Estimation

- **Labs:**
  - Time, Cost, Technology Selection and Stakeholder's fundamental satisfaction list
  - Programming environments: JMonkey, Android, etc.

- **Suggestions:**
  - The listing of team members and technology on MyCourses will be useful to you if there is another team that is using a similar technology as you when you may want some help.
  - When selecting technology for the Preliminary Design Document it is a good idea to experiment by installing software and trying it out before committing yourself to a particular technology. This is called Exploratory Research.

October

- **Goal:**
  - The Requirements Document

- **Due:**
  - Requirements Document due October 31st
  - Project Introduction Presentations, week 4
  - Meeting Prof or TA to review your documentation, week 4

- **Lectures:**
  - **Week 5**
    - Day 1: Requirements Document Format, Meetings & Speaking to client in their language
    - Day 2: Screen Flowchart and Screen Layouts
    - Day 3: Use Case Diagrams
  - **Week 6**
    - Day 1: Use Case Templates
    - Day 2: The Environment Diagram (not Environment Model)
    - Day 3: The Domain Model
Week 7
- Day 1: The Installation Model
- Day 2: Crafting the Requirements Document & Good design: completeness, correctness, and look
- Day 3: (overflow time)

Week 8
- Schedule meeting with Prof / TA to review Requirements Document
- Day 1, 2 & 3: Project Introduction presentations (10 minutes – email presentation to me)

- Labs
  - Use Case diagrams and templates, Environment Diagram and Domain Model

- Suggestions
  - It is important to meet me or the TA to review the quality of your document and diagrams
  - To facilitate the presentations email your presentation to me. You will play it from my laptop.

November
- Goal:
  - The Design Document
- Due:
  - Nothing
- Lectures:
  - Week 9
    - Day 1: Design document format & Good design: completeness, correctness, and look
    - Day 2: From Environment Diagram to Domain model to Design Patterns
    - Day 3: Encapsulation, Protocols, and Sequence diagrams
  - Week 10
    - Day 1: Essential Skills for the Agile Developer
    - Day 2: (overflow time)
    - Day 3: (overflow time)
  - Week 11
    - Day 1: Team software practises: directories, SVN, Peer Review
    - Day 2: Risk planning: Encapsulation with data validation, testing with assertions
    - Day 3: Risk planning: White box vs Black box testing, path identification
  - Week 12
    - Day 1: The Test Form, risk mitigation: multi-test, assume, pass, ...
    - Day 2: Putting it all together
    - Day 3: (overflow day)

- Labs:
  - Identifying design patterns
- Suggestions:
  - Make a meeting with either the TA or Prof to review your designs

December
- Goal:
  - Finishing your design document
- Due:
  - Design Document due 1st lecture in January
  - Final Term Exam, during exam period
- Lectures:
  - None.
- Lab:
  - None.
- Suggestions:
  - It would be a good idea to set a meeting with the Prof or TA to review your design before submission.
January

- **Goal:**
  - Software development in a team environment & Visiting speakers (seeing real-world engineering)
- **Due:**
  - None.
- **Lectures:**
  - Week 1
    - Day 1: Selection of - Team organization, work flow, problems
    - Day 2: Example scrum work flow with measurements and peer review
    - Day 3: Example classical work flow with measurements and supervisor review
  - Week 2
    - Possible visiting speakers
  - Week 3
    - Day 1: Managing Software Quality and Acceptance Demos
    - Day 2: Gold plating and ramp down
    - Day 3: Software Maintenance Form and Procedures
  - Week 4
    - Possible overflow time
- **Labs:**
  - As suggested by students or as needed
- **Suggestions:**
  - Start programming early
  - Use exploratory programming to work out issues now before they become a problem later!!

February

- **Goal:**
  - Programming!
- **Suggestions:**
  - No classes but make appointment for acceptance demo for March, first come first serve.
  - Do not fall behind in your programming.

March

- **Goal:**
  - Acceptance Demo & Evaluation: This is an important course milestone – is your project on track?
- **Due:**
  - Acceptance Demo & Project Management Evaluation (2nd or 3rd week of March)
  - Make appointment for classroom demo presentation for next month, first come first serve.

April

- **Goal:**
  - Final program Due.
- **Due:**
  - In class presentation and demo, last week(s) of class.
  - Make appointment for final program demonstration, in May (first come first serve).
- **Lectures:**
  - Possible visiting speakers