SOFTENG 306 Software Engineering Design 2

Course Outline. Semester 2, 2016

Draft as of 1st July 2016

Department of Electrical and Computer Engineering, The University of Auckland

Project 1 Contacts	**Project 2 Contacts**
Dr Oliver Sinnen (Lecturer)	Dr Rashina Hoda (Coordinator, Lecturer)
o.sinnen@auckland.ac.nz	r.hoda@auckland.ac.nz
Mostafa Mehrabi (TA)	Elliot Varoy (TA) evar872@aucklanduni.ac.nz
mmeh012@aucklanduni.ac.nz	Guyver Fu (TA) yfu959@aucklanduni.ac.nz

Team Project. Working in project teams to develop software to meet changing requirements for a large application. Project Planning. Requirements gathering. Estimating, costing and tracking. Acceptance and unit testing. Evolutionary design and development. Collaborative development tools.

The goals of the course include:

- Apply software process models and project management best practices to a large team project.
- Use a source code version control system to perform change management in large projects.
- Develop a software product of practical relevance in a team.
- Develop a test plan based on the specification of a program.
- Create documented and comprehensive open-source software releases.

These learning outcomes will be achieved by people working in teams to analyse, design, implement, test, document and present a solution to a specified problem, using version control and documentation tools. Lectures will introduce the project initially, then each team will work together to practically develop solutions, reporting regularly and documenting the regular project management process. Lectures will not present the solution methods, which should be based on courses undertaken prior to this semester, plus independent study of some additional material.

The project work follows on from your Part II and Part III semester 1 study, and relates to the semester 2 work. It is a stepping stone to Part IV project work where you will have more freedom in carrying out the project assigned only to your group. This paper continues the thread of professional design that runs throughout the Software Engineering program, and leads to professional work in industry, and to postgraduate study.

Approach. The approach is a "hands on" one; design skills and knowledge are acquired by practice. Design cannot well be taught as a passive academic subject. You will learn most effectively by actively engaging in the project team work.

You will document your ongoing project development on a group version control site, including code in the repository, documentation in the wiki, and issues in the ticket tracking. The first project will follow a more traditional project management approach to the groups' software development and the second project will use some key iterative software development.

Student time commitment. Over the course, the course work should take about 10 hours each week, for a total of about 120 hours over the semester. This includes a few hours receiving instructions (e.g. lectures) about the initial problem, and the rest of the time analysing, designing, implementing, testing, documenting, meeting your group regularly, and presenting your work to the staff. The process of managing and running your group work is an integral part of the coursework and should not be neglected.

Schedule and Assessment. (There is no final exam. Dates to be released in class.)

Week	Weighting	Description
1		Course introduction, Project 1 material, project management methods.
2		
3	5%	Project 1 plan due.
4	15%	Project 1 milestone demo/interview.
5		
6	20%	Project 1 final demo/interview.
	10%	Written report due.
		Mid-semester break
7		Introduction to iterative incremental development and Project 2.
8	5%	Project 2 Plan + demo due.
9	5%	Project 2 Design docs + demo due.
10	10%	Project 2 Prototype + demo due.
11		
12	20%	Project 2-Final Project + demo due
	10%	Written report due
2-12		TA session 12-2pm every Friday in UG4

Assessments will be held as per scheduling details above. The project plans for project 1 and 2 will be handed in, while some other assessments will be by a live demo and presentation given by all members of the group, in the lab or lecture, showing the software working, plus handed in documentation/report at the end of each project (details in class.) It is important to plan and practice your presentation and demo beforehand. It is important to ensure your code it working beforehand, to not change it at the last minute, and to keep a working version at each main step of development. Marks will be given for working code shown in the demo.

Each student is expected to have a full and complete knowledge of all aspects of every project, and this will be individually assessed in questions at interviews and in giving grades for each project. Each student is expected to contribute significantly to writing the code for each project. To assist marking, each student will be required to submit a confidential peer evaluation of their group members' performance.

Submission Your work will be submitted by checking in your code and documentation to the code repository. Markers will examine the version that was available at the deadline given, as well as the history of development of the code, documentation, tickets/user-stories, etc.

Marking in the demonstration will be by checking out the code from the repository, compiling and running it. Make sure your system builds correctly in a new directory.

Grade expectations Numerical marks will be assigned to each student for each component, and at the end of the paper a total mark will be calculated by weighting and adding the components. A letter grade will be assigned to the total mark. In a design paper with 100% coursework such as this one, the raw pass mark is typically well above 50%. For a pass, you should aim for 60% or higher. Your aim should be to make a fair contribution to your group in completing a working project and report. In general each student can expect a mark of: 9-10 for excellent or better work, 8 for good work that fulfils the requirements well, 6-7 for a satisfactory project or exercise that works and meets the requirements, 3-4 for a significant effort but that does not meet the requirements, 2-3 for a minimal effort. Numerical marks in Design papers are often high since many students put in considerable good work to produce good projects. High marks are needed to achieve an A.

Teamwork Effective teamwork is important in your future software engineering career (e.g. employers commonly ask academic referees about graduates' ability to work in a team). Here are some tips for helping your group work together:

- Focus on people's abilities and talents (not on weaknesses) and put these to good use.
- Agree on a group structure and meeting schedule. For Project 1, meet at least twice a week. For Project 2, groups are expected to work together for most of the time. Record meeting results on your group wiki.
- For Project 1 and 2, appoint a group leader, inform the staff (that is a requirement) and agree on the leader's role and each group member's role.
- If there is a disagreement, arrange to meet asap; it usually helps.
- If the group dynamics are not working after you attempt to resolve issues, then report to the staff about the difficulties. We will try to help. Let us know if a group member is not participating.

We will look closely at your code repository, online documentation, and ticket records (or user-stories and tasks) to see what contributions group members have been making, and how the group has consistently worked on the project over the project. Make sure you are doing things each week! If you are programming as a pair; take turns to check in the code so that each person has a track record of contributions, and also include the names of both of the pair in your comments for each check in. And contribute always with the same account (don't use multiple accounts to submit your work to the code repository).

In other words: it is your responsibility to both do the work and also communicate what you have done clearly. If there is no clear record of your work, we cannot give any marks for it.

Labs and software tools The teaching assistant will be available to help you on a regular basis, at set times each week.

For Project 1, the programming language will be Java and the application needs to be executable (and will be tested) on Linux platform. A well designed solution should equally work under Windows.

For Project 2, the implementation involves using a suitable game development kit such as the Android development kit (based on Java) or Unity. Games should be executable on Windows platform. Link to tutorial resources and help in labs will be provided.

Academic misconduct The University of Auckland will not tolerate cheating, or assisting others to cheat, and views cheating in coursework as a serious academic offence. The work that a student submits for grading must be the student's own work, properly acknowledged and referenced. This requirement also applies to sources on the world-wide web. A student's assessed work may be reviewed against electronic source material using computerised detection mechanisms. Upon reasonable request, students may be required to provide an electronic version of their work for computerised review.

See "Academic Honesty," accessible from the Current Students' Website under Academic Information. The Student Learning Centre and the Kate Edgar Information Centre can also provide students with assistance in this area.

Since the course work is carried out in groups, we will take some care to evaluate each individual student's work. We do not wish to see students move on to the Part IV project unless they are actively taking part and effectively contributing to the group work. Do not expect to hide behind your colleagues work; every student will be expected to make a contribution, to show it working, to report on it and to document it. It will not be enough to contribute to one part of the project; each student must make a contribution to all phases of the project (analysis, design, implementation, testing, project management, documentation, meetings, group dynamics, etc.)

We will use Canvas Discussions as the discussion forum for this course.

Problems? . . . Please come and see us or email if you have problems with the paper. We will use the Canvas forum for discussions about the course and people should submit their questions to the forum rather than emailing, so that everyone in the course can benefit from the discussion.