SENG1031 Software Engineering Workshop 1
Course Outline
Session 2, 2007
23 July 2007

## Contents


1 Course Information Summary
2 Course Philosophy
3 Course Aims
4 Learning Outcomes
5 Staff and Web Access
6 Meeting Times and Venues
7 Communication with students
8 Forum
9 Assessment
10 Conflict Resolution
11 Peer Assessment
12 References
13 Computing Resources
14 Academic Misconduct
15 Revision History


## 1 Course Information Summary

| | |
|---|---|
| Course Code | SENG1031 |
| Units of Credit | 6 |
| Hours per week | 5 |
| Prerequisite | none |
| Excluded | SENG1010, SENG1010 |
| Lectures | Monday 2-5pm |
| Location | MatSci G10 |
| Staff | Peter Ho & Ken Robinson |

## 2 Course Philosophy

Software Engineering is about designing and developing high quality software systems. People involved in software design and development must consider the full context of the proposed system including task requirements; user requirements and attributes; and technical possibilities and limitations. These considerations as well as the ethical and legal implications all form an integral part of the design process.

This workshop will offer students the opportunity to concentrate on software requirements analysis issues that address the above points. In addition, it aims to provide students with some of the skills, techniques and information that an engineer would use in the early stages of the development process.

## 3 Course Aims

This course will be project driven and students will have an opportunity to work on a significantly sized project that will tax your knowledge and skills acquired in this course.

In addition, the importance of people in technology, and the issues related to the quality of project management will also be emphasized. The role of managers, users, designers, programmers and analysts are critical throughout the system's development process. Students will have an opportunity to exercise their project management skills and to develop project plans that suggest "a good" solution to the problem in terms of cost, time and performance.

Aside from the primary task of building software systems, practitioners must also be able to critically review and evaluate such systems. In the latter part of the workshop, students will be required to critique the project report produced by other teams as well as your own. Our goal will be to improve your skills and techniques in being able to critically review software designs and to report on your findings.

## 4 Learning Outcomes

When students have successfully completed this course they will have a conceptual framework for:

- An appreciation of the role of Software Engineering in Software System development
- Experience with the development of project plans, design documents, status reports and results
- The recognition that production of quality software is a task demanding a disciplined approach to all stages of its development
- An understanding of the concepts and relevance of each stage of the software development life cycle
- An appreciation of the many and varied issues involved in the development of software systems

- The role and the importance that Software Engineering review processes play in producing quality systems
- The difference between a software bug and a requirements failure
- The ethical issues that software engineers have to be aware of

In particular, students will also learn and apply generic skills including interviewing techniques, the design review process, developing team skills, working creatively through group work and brainstorming and managing the varying levels of uncertainty involved in these processes. A primary goal of this course is to teach students the importance and process of group work.

Overall, students will learn that both behavioral (human based) and technical views (computer based) of design and implementation are equally important issues. In order to develop long-lasting and efficient software systems--in essence a quality product--Software Engineers need to be aware of and address both these factors in the development process.

Less tangible outcomes are for students to also see that specifications are not the panacea of correctness, specification can be wrong and more importantly they can be subtly wrong. In addition, students will see first hand how assumptions and bias can sometimes be embodied in a specification unintentionally.

## 5 Staff and Web Access

| Name | Location | Phone | Email |
|---|---|---|---|
| Peter Ho (LiC) | K17 - 212 | 9385 4052 | peterh@cse.unsw.edu.au |
| Ken Robinson | K17 - 208 | 9385 4045 | kenr@cse.unsw.edu.au |

The course website address is:

- http://www.cse.unsw.edu.au/~se1031

You must check this site regularly for updates and announcements. If you miss an important announcement because you did not check the class website, then you will have to wear any consequences that may result. No special dispensation will be granted as a result.

## 6 Meeting Times and Venues

SENG1031 is a workshop that can have up to 3 hours of contact per week. This will vary during the session as the need arises. Material from INFS1611, which no longer runs, is still being integrated into this course, as a result the lecture schedule listed below is tentative and will be subject to change as the need arises:

Location: Monday 2:00-5:00 PM, MatSci G10

| Weeks | Topic | Mode | References |
|---|---|---|---|
| 1 | Course Intro | Lec | |
| 2 | Software process | Lec | LM1 |
| 3 | Requirements Elicitation | Lec | KS3; LM3 |
| 4 | Requirement (FN & NF) | Lec | KS8.1, 8.2, 5 |
| 5 | Case study 1 | Tut | |
| 6 | Requirements Analysis | Lec | LM4.3; KS3.3, 3.4 |
| 7 | Case study 2 | Tut | |
| 8 | Spare | | |
| 9 | Presentations | | |
| | BREAK | | |
| 10 | Requirements v Bugs | Sem | |
| 11 | spare | | |
| 12 | Guess speaker (TBA) | Lec | |
| 13 | What's next | Lec | |
| 14 | Round up | Lec | |

The lectures are nominally 2 hours and may be shorter depending on the subject matter. During the presentation week, 3 hours are required and you are expected to make suitable arrangments to be present on that Monday between 2-5pm.

In addition to the lectures, you are expected to spend on average half an hour with your group mentor once a week. Where convenient, you may arrange to have your mentor meetings from 4-5pm Monday (ie after the lecture) so as to utilize the lecture room. However, if you wish to re-schedule your mentor meetings to a more convenient time, then as a team, you must negotiate a suitable compromise with your mentor. Everyone in the team must be present at each and every mentor meeting so it is essential that you find a time that is convenient for all members of the team.

## 7 Communication with students

In addition to the UNSW email address, the School of CSE provides students with a login account and email. You are hereby advised that:

- It is the policy of the School that all contact and announcements will be made via your official CSE email address only; and
- The course web page will be the other main source of announcements and information.

Students who choose to use other email accounts do so at their own risk as they may miss important announcements. You are adviced to redirect your CSE email to your preferred email account. You can redirect your CSE email by using

the mlalias command. Click here for more information on using this command.

If you do not want to redirect your CSE email then it is your responsibility to check your CSE email account and the course web page on a regular basis.

## 8 Forum

In addition to the above mentioned channels of communication, the course also supports a class forum which is open to students of this course. It provides a means of sharing problems and issues related to the project and coursework. A link to the forum will be made available from the class home page and an announcement will be made when it is configured. All students are encouraged to use the forum in lieu of direct emails to the lecturing staff.

## 9 Assessment

In order to pass this course, a satisfactory performance is required in each component of the assessment.

| Task | Mark | Due Date |
|---|---|---|
| Gantt Charts | 10 | weekly from Week 3 |
| Requirements Report | 25 | Week 7 |
| Presentation | 15 | Week 9 |
| Requirements Specification | 25 | Week 11 |
| Critique Report | 25 | Week 14 |

For the project, students will be placed in teams. All teams will have a uniform and fixed number of students, ideally 4 students per team but this will depend on the course population size. Each team will be responsible for producing and submitting the deliverables outlined in the Assessment table shown above. Lecture and Mentor meetings will provide specific guidance for preparing each of the required deliverables. Once students have been allocated to a team they will not be allowed to change to another team.

IMPORTANT If you are unsure about being enrolled in this course, do not put yourself into a team as this will cause a significant disruption to the other members of the team and the class as a whole when you do decide to discontinue. Put your name on a pending list, this will be handed out in the first lecture.

## 10 Conflict Resolution

Students working in a team must endeavor to behave in a cooperative and professional manner. However conflicts may still arise from time to time and it is in your interest to resolve the matter amicably while the matter is minor. Your mentor should be kept informed of such problems. If required your mentor will help in the mediation process. It is no point waiting till the end of session to vent your dissatisfaction and frustration.

Should the conflict escalate beyond being a minor disagreement, then as the lecturer in charge, I reserve the right to arbitrate on the matter and to make a decision that is final. Once again it is in your interest to resolve the matter before it escalates.

## 11 Peer Assessment

Your overall performance in the team project will be affected by how you perform in your team. This will be determined by how you are rated by your team members and vice versa. Academic Staff reserves the right to moderate marks according to the information provided by your peers and mentor. A failure to actively participate or where participation is considered marginal will result in a fail for the course.

For each deliverable, every member of your team must declare their percentage of contribution to that particular deliverable. The sum total of everyone’s contribution must add up to 100%. The declaration must be signed by each member of the team to signify that they agree with the declared individual contributions for that deliverable. Marks will be withheld until there is a signed agreement for that deliverable.

## 12 References

Prescribed text:


- Leszek A Maciaszek, Requirements Analysis and System Design, 2e, Addison Wesley, UK, ISBN 0321204646, 2005.


Recommeded text:


- Robert Glass, Software Runaways, Prentice-Hall, ISBN 013673443X, 1997.
- Gerald Kotonya and Ian Sommerville, Requirements Engineering, processes and techniques, Wiley, NY, ISBN 0471972088, 1998.


Reference text:


- Shari Pfleeger, Software Engineering Theory and Practice, Second Edition, Prentice-Hall, 2001.
- Ian Sommerville, Software Engineering, 6th ed, Addison-Wesley Publishing Company, 2001.
- Watts S. Humphrey, Introduction to the Personal Software Process , SEI Series in Software Engineering, Addison-Wesley Longman, Inc., 1997.
- Watts S. Humphrey, Introduction to the Team Software ProcessTM, Addison-Wesley Longman, Inc., 2000.

## 13 Computing Resources

Students will be able to make use of computing resources in CSE labs. CSE labs are located in the Electrical Engineering, Mechanical Engineering and the K-17 building, a map is available at:

- http://www.cse.unsw.edu.au/ helpdesk/webpages/csemaps.html

## 14 Academic Misconduct

Copying assignments is unacceptable. Assignments will be checked. The penalties for copying range from receiving no marks for the assignment, through receiving a mark of 00 FL for the course, to expulsion from UNSW (for repeat offenders). Allowing someone to copy your work counts as plagiarism, even if you can prove that it is your work.

Further details you are directed to the CSE School Plagiarism Policy. You acknowledged receipt of these rules when you obtained your CSE computer account, and the link above is for your convenience so that you can review the rules now.

We are aware that a lot of learning takes place in student conversations, and don't wish to discourage those. However, it is important, for both those helping others and those being helped, not to provide/accept any programming language code or other written material in writing or electronically, as this is apt to be used exactly as is, and lead to plagiarism penalties for both the supplier and the copier of the codes. Write something on a piece of paper, by all means, but tear it up/take it away when the discussion is over.

If you are new to studying in Australia, be aware that attitudes to plagiarism at UNSW may be different from those in your home country. Make sure you are clear about the rules here at UNSW. In brief, and for the purposes of SENG1031, plagiarism includes copying or obtaining all, or a substantial part, of the material for your project, whether written or graphical report material, without written acknowledgment in your assignment from:

- a location on the internet;
- a book, article or other written document (whether published or unpublished) whether electronic or on paper or other medium;
- another student, whether in your class or another class;
- a non-student (e.g. from someone who writes assignments for money).

Note that if you copy code or other material from another student or non-student with acknowledgement, you will not be penalised for plagiarism, but you are unlikely to get any marks for the copied code/material. If you use code or material found in a publication (on the internet or otherwise) then the marks you get for this will be at the marker's discretion, and will reflect the marker's perception of the amount of work you put into finding and/or adapting the code/material, and the degree to which you understand the code/material.

Note also that there is a big difference between being able to understand someone else's code/material, and writing that code/material yourself from scratch. A software engineer must be able to derived their own work from scratch.

## 15 Revision History

22 July 2007
: Initial draft.

23 July 2007
: Published via class website.

---

Last Updated: 11/19/2021 14:48:42