SENG1031 Software Engineering Workshop 1
Course Outline
Session 2, 2008
20 July 2008

Contents

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

1 Course Information Summary

Course Code	SENG1031
Units of Credit	6
Hours per week	5
Prerequisite	none
Excluded	SENG1010, SENG1010
Lectures	Tuesday 3-6pm
Location	Chemical Sc M10 (F10-M10)
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 all 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
- An awareness of some of the ethical issues that software engineers face

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.

In addition, students will also develop an awareness of the community of engineering professions and the importance of keeping current through life-long learning and through interacting with that community. Students will also be encouraged to develop their research skills as one of the means of acquiring the necessary knowledge and skills to solve engineering problems.

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

Important: 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 Lecture Times and Venues

Three consecutive hours per week has been set aside for the course. The lectures are nominally 2 hours but may be longer or shorter depending on the subject matter. Students should avoid scheduling tutorials or laboratory classes from other courses in this time slot.

Time:Tuesday 1500 - 1800
Location:Chemical Science M10

Important: Lectures starts in Week 0 and all students are expected to attend.

During Presentation Week, 3 hours are required and you are expected to make suitable arrangements to be present on that Tuesday between 3-6pm.

7 Syllabus and Teaching Strategy

Unlike the majority of the courses in your degree, all the SENG workshops are project driven courses focusing on the practical application of software engineering skills acquired from other courses in the degree program.

The workshops are not designed to be lecture-based courses. Lectures in this course are there to provide students with support and guidance in developing their practical skills. Students are expected to engage themselves in research and self-study of the materials required for this course.

Being a project driven course, the lecture schedule for this course may vary during the the semester as the need arises. Changes to the lecture schedule will be announced on the course website.

Weeks	Topic	References
0	Course Intro & Admin	
1	Intro to Requirements	SL1; KS5
	Software Process	SL7; LM1.4
	Project Management	LM1.4.3.1
2	Requirements	
	- Functional	SL3; KS5; LM2.1
	- Non-functional	SL6; KS8.1; KS8.2; LM2.1
3	Requirements Elicitation	SL8; KS3; LM2.2
4	Case study 1	
5	Requirements Analysis	SL2; SL4; KS3.3, KS3.4; LM4
6	Spare	

7	Case study 2
8	Spare
9	Presentations Break
10	Quality: Checking & ValidationSL9
11	Guess speaker (TBA)
12	Round up

The topics covered in this course will be based on the prescribed text written by Soren Lauesen. There is an expectation that you would have read the material before the lecture. Secondary references in the table above are there to provide alternate views and approaches to the topic.

References
SL:Soren Lauesen [Prescribed Text]
KS:Gerald Kotonya & Ian Sommerville
LM:Leszek A. Maciaszek

8 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 advised 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.

9 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.

10 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 2
Interim Requirements Report	10	Week 4
Group Presentation	15	Week 9
Requirements Specification	40	Week 9
Critique Report	25	Week 12
Total	100	

Table 1: Summary of Assessments

Important: Individual student marks for each deliverable (except for the Gantt Charts) will be calculated based on a peer assessment formula (see Peer Assessment below for more information).

Late Penalty: A late penalty of 10% per day applies to late submissions. The penalty applies to the assessed value of the deliverable; individual marks will then be calculated using the peer assessment formula after the penalty has been applied.

11 Group Work

For the project, students will be placed in teams. You are expected to spend on average half an hour with your group mentor once a week. If convenient, you may arrange to have your mentor meetings from 5-6pm Tuesday (ie after the lecture) as the lecture room is booked until 6pm. 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.

Important: 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. Missing your mentor meeting without a valid reason may have an effect on your final mark.

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 a team they will not be allowed to change to another team. Team

formation will take place during week 0's lecture.

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 and inform the lecturer of your situation. The pending list will be handed out in week 0's lecture.

12 Peer Assessment

For each deliverable tabled above (except for the Gantt Charts), peer assessments will be used to rate your overall performance in the team. Peer assessment allows your team members to rate the way you worked during that particular deliverable. Some (but not exhaustive) factors that may influence your peer rating might include:

- how you discharged your share of the tasks;
- how well you worked to project management time lines;
- how you actively participated in group activities and discussions;
- how you dealt with and communicated with other members of the group.

Peer assessments are confidential: All peer assessments must be an honest appraisal of your team member's performance for that deliverable.

All peer assessment submissions will be collected electronically. More details of how to use the electronic submission will be announced later during the session.

Note: The mark for individuals within a group after the peer assessment process can result in a calculated mark that is higher or lower than the assessed mark given to that deliverable. This is the group's direct recognition of the good or poor performance of individuals within the team.

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.

Peer assessment is compulsory: All students must submit their peer assessment by the deadline specified. Notification of the deadlines will be made via the class web page. Failure to lodge your peer assessments by the due date will result in a 5% late penalty being applied to your final mark for every peer assessment submitted late.

Individual marks for a deliverable will be calculated using the following formula:

$$mark = projectMark + scaleFactor$$

where

$$scaleFactor = 1 - e^{(10 - (Share * 10))}$$

Share=peer assessment value between 0.7 - 1.3

Example

Assuming a team size of 4 and a project mark of 80. A peer assessment value of 1.0 indicates that the individual should be awarded a 100% share of the deliverable's mark. The following table shows the effect on an individual's mark when the peer assessment value deviates above or below the 1.0 norm (the values were calculated using the formula shown above):

Share	Mark
0.80	73.6
0.85	76.5
0.90	78.3
0.95	79.4
1.00	80.0
1.05	80.4
1.10	80.6
1.15	80.8
1.20	80.9

In essence, the penalty for performing poorly in your group activity can be significant. Conversely, the reward for putting in a greater amount of work only provides a small return. The goal of using a non-linear scaling formula is to motivate poor performers and to encourage better collaboration between team members.

13 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. There is no point waiting till the end of session to vent your dissatisfaction and frustration, the damage will have already been done. It is best to solve the problem when it is small and manageable.

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.

14 References

Prescribed text:

- Soren Lauesen, Software Requirement, Styles and Techniques, Addison-Wesley, UK, ISBN 0201745704, 2002.

Recommeded text:

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

15 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

16 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, designs 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 (past or presently enrolled or 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.

17 Revision History

15 July 2008
: Initial draft.

18 July 2008
: Updated syllabus to suit new textbook

19 July 2008
: Updated assessment and peer assessment method

20 July 2008
: Course outline published

30 July 2008

Fixed broken mlalias link

Last Updated: 11/19/2021 14:25:47