Course Outline

SENG3010
Software Engineering Workshops 3A/B
Teaching Period 1 2012

Contents

- Course Staff
- Course Details
- Course Summary
- Course Aims
- Student Learning Outcomes
- Assumed Knowledge
- Teaching Rationale
- Teaching Strategies
- Assessment
- Academic Honesty and Plagiarism
- Course Schedule
- Resources for Students
- Course Evaluation and Development

Course Staff

Contact Details

Staff Name	Role	Email	Phone
Fethi Rabbi	Course Convenor	f.rabbi at unsw.edu.au	93854179

Consultations

Staff Name	Day/Time	Location

Course Details

Course Code:	SENG3010
Course Title:	Software Engineering Workshops 3A/B
Units of Credit:	6
Course WebSite:	http://www.cse.unsw.edu.au/~se3010/
Handbook Entry:	http://www.handbook.unsw.edu.au/undergraduate/courses/2012/SENG3010.html

Course Summary

The purpose of the 3rd year software engineering workshops is to give students experience with a group-based large-scale software development project involving a realistic application in the business domain (finance). This session, both SENG3010/SENG3020 will be offered in an integrated way i.e. the same teams will be working in developing a complex software application. SENG3010's focus is to learn about a new application domain, study the requirements, manage the project, liaise with the stakeholder and deliver high quality working solutions. SENG3020's focus is on software design, testing, reporting and the use of support tools around these activities. These workshops are supported by several industry partners which provide guest lectures, realistic datasets and software tools.

Course Aims

Both third year workshops will focus on the issues of designing and implementing a quality software system that conforms to the requirements of a stakeholder. Besides sharpening their design and coding skills, students will have to get immersed in a complex application domain, learn the basic concepts to be able to understand the requirements and continuously communicate with the stakeholder to discuss issues arising from the project such as design trade-offs, additional functionalities, new interface features etc. In addition, students will develop interpersonal communication skills by preparing correctly formatted and structured reports, negotiating technical, management and interpersonal issues within their teams and resolving problems within their development teams using effective conflict resolution techniques.

Unlike previous projects with rigid requirements, groups are encouraged to be creative in their implementation

by focusing on delivering the best quality product in consultation with the stakeholder. Consequently, they have the complete freedom in choosing the implementation technologies of their choice e.g. (Java, C++, J2EE, .NET or hybrids) providing they meet the requirements of the stakeholder.

The application domain selected in this session is in the finance area. In this workshop, students will be issued with an initial requirements document outlining the essential features of a product. Additional information will be given on a needs basis throughout the workshop in a variety of ways (emails, regular meetings, class lecture). A guest lecture will be organised with an industry speaker and access to relevant data from the Australian Stock Exchange will be facilitated.

In addition, this session is piloting the use of an IBM commercial platform for managing software development projects called Rational Team Concert (aka JazzHub). Teams of students will be using this platform to collaborate with each other as well as manage the deliverables.

Student Learning Outcomes

After completing this course, students will:

- reinforce existing knowledge about the concepts and principles of software development associated with the implementation of quality software within an organisational context.
- learn about the processes of requirements elicitation and engineering in a realistic context
- acquire practical design skills, particularly in architectural design and software component integration
- experience the process of implementing a quality software system by choosing appropriate languages, libraries and frameworks.
- acquire additional skills involved in working as part of a project team implementing a quality software system within strict time constraints.
- learn the process of writing reports and documentation for specific needs.
- get introduced to a new business application domain (financial market operations in this case)

This course contributes to the development of the following graduate attributes:

Graduate Attribute	Where Acquired
the skills involved in scholarly enquiry	yes
an in-depth engagement with relevant disciplinary knowledge in its interdisciplinary context	yes
the capacity for analytical and critical thinking and for creative problem solving	yes
the ability to engage in independent and reflective learning	yes
the skills to locate, evaluate and use relevant information (Information Literacy)	yes
the capacity for enterprise, initiative and creativity	yes
an appreciation of and respect for, diversity	no
a capacity to contribute to, and work within, the international community	no
the skills required for collaborative and multidisciplinary work	yes
an appreciation of, and a responsiveness to, change	yes
a respect for ethical practice and social responsibility	no
the skills of effective communication	yes

Assumed Knowledge

Before commencing this course, students should have:

- The ability to develop complex programs from stated requirements
- The ability to design and implement algorithms for text and data processing
- Good knowledge of design concepts and techniques (equivalent to UML class diagrams and ER)
- Good knowledge of database and Web technologies
- Writing and communication skills

These are assumed to have been acquired in the previous software engineering courses and workshops

Teaching Rationale

This course adopts a project-based approach to Learning and Teaching where students learn through applying their knowledge in situations inspired from real-life. Software development in a group situation is encouraged with the lecturer and other external evaluators guiding and providing continuous feedback to each group.

Teaching Strategies

In this workshop, we will follow a product-based framework to the project-based learning. A set of intermediate deliverables leading to a product are specified by the stakeholder, a role assumed by the lecturer in charge. Some weekly lecture slots will be used to elaborate on the deliverables and answer general questions. During these meetings, teams are encouraged to discuss their progress and demonstrate work-in-progress. Teams can

also arrange additional meetings with the stakeholder if required. A tutor will be available to assist with technical matters and answer queries related to the case study. On-line assistance will also be provided in the form of a discussion forum.

Assessment

The assessable components for the course are divided into two distinct categories (SENG3010 and SENG3020)

- SENG3010 (Understanding of application domain, project management and quality of product)
  - Functionality of the four prototypes (5%+10%+15%+30%=60%).
  - Management Reports (5%+15%=20%).
  - Mentor Evaluation (10%)
  - On-Line Project Management (10%).
- SENG3020: (Quality of design, coding and testing)
  - Design Reports (5%+15%=20%).
  - Testing Report (10%)
  - On-Line Code Integration (10%)
  - Technical quality of the four prototypes (5%+10%+15%+30%=60%).

Details about these assessment components and the deadlines will be given via the course's on-line system.

Academic Honesty and Plagiarism

Plagiarism is defined as "using the words or ideas of others and presenting them as your own". UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Learning Centre: Plagiarism and Academic Integrity
- MyUNSW: Plagiarism and Academic Misconduct
- CSE: Addendum to UNSW Plagiarism Guidelines
- CSE: Yellow Form (whose terms you have agreed to)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism.

Course Schedule

Weekly Schedule

- Week1: Introductory lecture. Guest Lecture: ASX Trading Process.
- Week2: Groups formation. Guest Lecture: Agile Software Development.
- Week3: Finalising groups. APESMA Presentation. Guest Lecture: Rational Team Concert (aka JazzHub) by IBM.
- Week4: Mentoring Meeting. Management report 1 due.
- Week5: Mentoring Meeting.
- Week6: Mentoring Meeting. Assignment 1 Due (Prototype 1 and Design Report 1)
- Week7: Mentoring Meeting.
- Week8: Mentoring Meeting. Assignment 2 Due (Prototype 2 and Testing Report). GUI requirements lecture.
- Week9: Mentoring Meeting.
- Week10: Public Demonstrations. Assignment 3 Due (Prototype 3 and Presentation Slides)
- Week11: Mentoring Meeting.
- Week12: Final demonstrations. Assignment 4 Due (Prototype 4, Design Report 2, Management Report 2)

Additional details and changes will be posted on the course's noticeboard.

Resources for Students

For domain knowledge (i.e. Finance), students are encouraged to research appropriate sources of information depending on their needs. They are also expected to learn about the basic concepts using Web sources (such as the ASXÂ's Web site www.asx.com.au).

Support for most questions, issues, concerns with Rational Team Concert and JazzHub will come from the jazz.net community itself. There are many Forums and Discussion groups that will provide very good support. Additional details will be given during IBM's guest lecture.

Course Evaluation and Development

This course is evaluated each session using the CATEI system.

In the previous offerings of this course (both SENG3010 and SENG3020), students noted that more structure needed to be imposed on the requirements and what was expected. Also, students wanted to demonstrate their work as early as possible. Finally, the best students wanted more challenging requirements.

Based on these comments, the specifications will be released as early as possible. A table of requirements, some of which will be highly complex, will be compiled and progress will be judged against this table. In addition, each mentoring session will include a mandatory demonstration. In addition, an On-Line Project Management System provided by IBM has been introduced to support the agile development process