Course Outline

SENG3020
Software Engineering Workshop 3B
Teaching Period 2 2011

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 Rabhi	Course Convenor	f.rabhi at unsw.edu.au	93854179
Adnan Rashid	Course Admininstrator	arashid at cse.unsw.edu.au	
Kader Lattab	Course Admininstrator	klattab at cse.unsw.edu.au	
Martin Bajalan	Course Admininstrator	mbaj471 at cse.unsw.edu.au	

Consultations

Staff Name	Day/Time	Location

Course Details

Course Code:	SENG3020
Course Title:	Software Engineering Workshop 3B
Units of Credit:	6
Course WebSite:	http://www.cse.unsw.edu.au/~se3020/
Handbook Entry:	http://www.handbook.unsw.edu.au/undergraduate/courses/2011/SENG3020.html

Course Summary

This course is the last installment of the software engineering workshops. It aims at allowing students to pursue a group- based large-scale software development project concerning a realistic application. in the business domain.

Course Aims

This workshop is similar to SENG3010 in that it focuses 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. In this workshop, teams will be working on separate software components that are somehow linked. Each team will be directly supervised by a mentor with significant programming experience. The mentor will advise on the technologies and methods to be used as they will depend on the task at hand. The application domain selected in this session will still be the finance area. Each mentor will be issue an initial requirements document outlining the essential features of a product.

Additional information will be given during the mentoring sessions. A guest lecture will be organised with an industry speaker in the area of algorithmic trading.

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:

- advanced knowledge of programming and algorithm design
- 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 core software engineering courses.

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 mentor. Weekly mentoring sessions are 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 their mentor. All teams will be publicly demonstrating their work few weeks before the end of the workshop to give them opportunities to improve their product based on the feedback received prior to the final demonstration in the last week of the course.

Assessment

The assessable components for the course are divided into two distinct categories:

- Coding and demonstrations
  - First prototype (10%).
  - Public Demonstration (15%).

- Final code inspection (10%).
- Final demonstration (35%).

- Reports
  - Initial Design and Project Management Report (5%)
  - Mentor Evaluation (10%)
  - Final Documentation (15%)

Details about these assessment components and the deadlines are given in the Assignments page.

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.
- Week2: Groups formation. Guest Lecture. First Mentoring Meeting.
- Week3: Mentoring Meeting.
- Week4: Mentoring Meeting.
- Week5: Mentoring Meeting.Initial report due.
- Week6: Mentoring Meeting. First prototype due.
- Week7: Mentoring Meeting.
- Week8: Public Demonstrations
- Week9: Mentoring Meeting.
- Week10: Mentoring Meeting.
- Week11: Mentoring Meeting and Early Demos.
- Week12: Final demonstrations.

Resources for Students

Students are encouraged to research appropriate sources of information depending on their topic. They are also expected to learn about the basic concepts in the application domain by using Web resources (such as the ASXÂ’s Web site www.asx.com.au).

Course Evaluation and Development

This course is evaluated each session using the CATEI system.

In the previous offering of this course, 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 and experienced mentors will be used to specify complex problems . There will be a public demonstration few weeks before completion of the workshops to enable students to get feedback early.