Course Outline

SENG3020
Software Engineering Workshops 3B
Teaching Period 2 2010

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	

Consultations

Staff Name	Day/Time	Location

Course Details

Course Code:	SENG3020
Course Title:	Software Engineering Workshops 3B
Units of Credit:	3
Course WebSite:	http://www.cse.unsw.edu.au/~se3020/
Handbook Entry:	http://www.handbook.unsw.edu.au/undergraduate/courses/2010/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 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. 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. An industry sponsor will be awarding a prize to the best project.

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 elicitations 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 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
- good knowledge of design concepts and techniques (equivalent to UML class diagrams and ER)
- good knowledge of database and Web technologies
- 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 guiding and providing continuous feedback to each group.

Teaching Strategies

In this course, 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. Teams can organize themselves and arrange meetings with the stakeholder as 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:

- Coding and demonstrations
  - First prototype (5%).
  - Second prototype (15%).

    - Final demonstration (45%).
- Reports
    - Initial Design and Project Management Report (5%)
    - Testing Documentation (10%)
    - Final Documentation (20%)

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: Lecture on orderbooks. Groups formation. Session 1 lab released.
- Week3: Teams mentoring. Session 2 lab released. Initial report due on Friday.
- Week4: Teams mentoring.
- Week5: Guest lecture. Session 3 lab released. First prototype due.
- Week6: Mentoring (by tutor).
- Week7: Mentoring (by tutor).
- Week8: Lecture on visualisation requirements. Second prototype due.
- Week9: Mentoring. Testing doc due on Friday.
- Week10: Mentoring.
- Week11: Mentoring and early demos.
- Week12: Final demonstrations.

Resources for Students

Students are encouraged to research appropriate sources of information depending on their needs. They are also expected to learn about the basic concepts in the application domain by using Web sources (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 courses, students noted ...

Based on their comments, we have ...