Course Outline Public

Edit Resource (/SENG3011/20T1/resources/41287/edit)

Contents

- Course Details
- Course Summary
- Course Timetable
- 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 Details

Course Code	SENG3011
Course Title	Software Engineering Workshop 3
Units of Credit	6
Course Website	http://cse.unsw.edu.au/~se3011 (http://cse.unsw.edu.au/~se3011)
Handbook Entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/SENG3011.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/SENG3011.html)

Course Summary

The purpose of the 3rd year software engineering workshop is to give students experience with a group-based large-scale software development project involving a realistic application in a real-life domain (data analytics).

In this session, teams will be developing a complex software application and the 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. Another aspect of the workshop is to reinforce skills in software design, testing, reporting and the use of support tools around these activities.

Course Timetable

The course timetable is available here (/SENG3011/20x1/timetable) .

Course Aims

This third year workshop 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.

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 data analytics 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). Many guest lectures will be organised with industry speakers and access to relevant data for the project will be facilitated.

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, 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. Assistance can also given over email by the lecturer in charge.

Assessment

There are four assessable components for the course with respective deliverables. Details of the deliverables are described in GitHub repository specifications.

- D1(10%) - Initial setup of the GitHub repository (5%), Project management plan and initial API design (5%)
- D2(20%) - API implementation and specification in swagger (15%), API design and test documentation (5%)
- D3(20%) - Platform prototype demonstration (20%)
- D4(50%) - Final demonstration (30%), Final GitHub repository (5%), Final report (15%)

Academic Honesty and Plagiarism

Plagiarism is defined as (https://student.unsw.edu.au/what-plagiarism) *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 (http://www.lc.unsw.edu.au/plagiarism/index.html)
- MyUNSW: Plagiarism (https://student.unsw.edu.au/plagiarism) and Academic Misconduct (https://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf)
- CSE: Addendum to UNSW Plagiarism Guidelines (http://www.cse.unsw.edu.au/~chak/plagiarism/plagiarism.html)
- CSE: Yellow Form (http://www.cse.unsw.edu.au/people/studentoffice/policies/yellowform.html) (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

- Week 1 (Tuesday 11-13. Colombo C): Introductory lecture. Group Formation.
- Week 1 (Friday 14-16 Colombo C): Guest lectures. Finalising Groups.
- Week 2: Mentoring (Ainsworth 101).
- Week 3: Mentoring (Ainsworth 101). D1 due.
- Week 4: Mentoring (Ainsworth 101).
- Week 5 (Tuesday 11-13. Colombo C): Extra lectures.
- Week 5 (Friday 14-16 Colombo C): Extra lectures.
- Week 5: Mentoring (Ainsworth 101) D2 API specification URL due.
- Week 6: Mentoring (Ainsworth 101). D2 due.
- Week 7: Mentoring (Ainsworth 101).
- Week 8: First presentations D3 (Ainsworth 101+extra rooms).
- Week 9: No mentoring (Easter Monday)
- Week 10: Mentoring (Ainsworth 101).
- Week 11: Final demonstrations (Ainsworth 101+extra rooms). D4 due.

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

Resources for Students

For domain knowledge, students are encouraged to research appropriate sources of information depending on their needs. Domain experts will also be made available on-demand. Students are also expected to learn about the basic concepts using Web sources (more information will be given).

Course Evaluation and Development

This course is evaluated each session using the CATEI system.

In the previous offering of this course, feedback was generally positive mainly because of the industry nature of the project and the involvement of industry partners. The course sponsor (optiver) gave at the end of the workshop a very clear message that this course supports software development skills being put into practice and provides a great foundation for working with industry.

Most suggestions for improvements relate to how deliverables were assessed. In response, every effort will be made at explaining the criteria used for marking in the documentation and during mentoring sessions with students.

Having said that, students must also appreciate that workshop projects are not programming assignments. They deliberately have some ambiguities in the requirements and whenever unsure, students should seek advice during mentoring sessions. In industry, it is rare that all requirements are clearly written down in advance for every project and workshops are designed to teach students to cope with uncertainty in business environments.

Here is the message that Optiver (the course sponsor) has sent us after the last offering. Optiver is the proud sponsor of SENG3011. Software Developers at Optiver are fully responsible for the entire development process, working closely with Traders as end users, designing and testing their own systems and making appropriate technology choices. This course supports these skills being put into practice and provides a great foundation for working with us. We also believe an interest and passion for the domain of Finance makes people successful at Optiver and helps them enjoy their work, so the relevant project this year would help students determine if it's something they enjoy and provided them with the opportunity to consider whether

they would like to start their career with us. Having been heavily involved in the course this year we were extremely impressed with the calibre of projects presented to us. The amount of work that went into this project is evident in the high quality of work they produced, and is a credit to the students involved. Due to the outstanding effort, and quality of projects we decided to offer an additional prize to the runners up. We wish to congratulate the winning group and the runners up and commend the other groups for getting involved.

Resource created 2 months ago (Tuesday 21 January 2020, 01:41:32 PM), last modified 20 days ago (Thursday 20 February 2020, 02:48:50 PM).

Comments

(/SENG3011/20T1/forums/search?forum_choice=resource/41287) (/SENG3011/20T1/forums/resource/41287)

There are no comments yet.