Course Outline

SENG2021
Software Engineering Workshop 2B
Semester 2 2014

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
Lawrence Yao	Lecturer	lawry at cse.unsw.edu.au	
Matthew Saxby	Lecturer	mcsa293 at cse.unsw.edu.au	

Consultations

Staff Name	Day/Time	Location

Course Details

Course Code:	SENG2021
Course Title:	Software Engineering Workshop 2B
Units of Credit:	6
Course WebSite:	http://www.cse.unsw.edu.au/~se2021/
Handbook Entry:	http://www.handbook.unsw.edu.au/undergraduate/courses/2014/SENG2021.html

Course Summary

This course represents the third of a series of software engineering workshops designed to teach students to work in teams and apply their knowledge to solve real-life problems. In this workshop, students will be getting experience on different aspects of designing a Web application with a major focus on the front-end. Activities in this course include developing user requirements, producing design documents, designing user interfaces and producing a prototype. The requirements for this project will be given by an industry partner and will relate to designing a realistic infographic application. Most of the teaching will be conducted via mentoring of the teams. At the beginning of the course, some lectures will give background on some key technologies and on how to document designs in general. The course has a number of industry sponsors that incude Powerhouse Museum, Fairfax Media and Macquarie Bank.

Course Aims

To develop:

- a practical appreciation of the software design process;
- an understanding of the relation between specification concepts and implementation considerations;
- an understanding of Web systems software design approaches;
- design documents describing how a specified system will be implemented;
- an executable prototype using Web development tools.

- a report summarising all the activities and experiences of the workshop.

Student Learning Outcomes

After completing this course, students will:

- reinforce existing knowledge about the concepts and principles of designing quality software within an organisational context.
- learn about the processes of converting requirements to design in a realistic context
- acquire practical design skills, particularly in architectural design and software component integration
- experience the process of implementing a prototype Web system by choosing appropriate languages, libraries and frameworks.
- acquire additional skills involved in working as part of a project team working within strict time constraints.
- learn the process of writing reports and documentation for specific needs.
- get introduced to a new business application domain (infographics 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 requirements documents
- The ability to design and implement general algorithms
- Basic knowledge of essential design concepts and techniques (equivalent to UML class diagrams and ER)
- Basic knowledge of scripting and Web technologies
- Writing and communication skills

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

Teaching Rationale

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.

Teaching Strategies

Early weeks will consist of lectures, Afterwards, all teams will meet weekly with their mentors.The Schedule specifies the activities for each week. Teams are offered the possibility to hold additional mentoring sessions if the need arises. Students can also ask for lectures on particular topics.

The Macquarie Second Year Software Engineering prize is awarded to one team from SENG2021 in a particular year. A number of teams usually three are chosen on the basis of their final demonstration and are asked to prepare a 20 to 30 minute presentation explaining their design and prototype implementation of the current project. The presentation is to be made to members of Macquarie Bank.

Assessment

The assessable components for the course are:

- Three design reports: Week 5 (10%) + Week 7 (10%) +Week 12 (25%)=45%.

- Two software prototypes Week 9 (15%) +Week 12 (30%)=45%.
- Mentoring sessions participation: 10%.

For more information on these deliverables, see the Course Web page.

It is assumed that all students will have read and understood the regulations outlined in the myUNSW Assessment Policy If you have not familiarised yourself with these it is strongly suggested that you do so. This course will be administered using those regulations.

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 (30 July): Introductory lecture. Guest lectures:project idea and some implementation considerations. Overview of Web development architectures and frameoworks (Fairfax).
- Week2 (6 August): Guest lectures: key Web technologies. Finalising groups.
- Week3 (13 August): Mentoring Meeting.
- Week4 (20 August): Mentoring Meeting.
- Week5 (27 August): Guest lectures: systems analysis, more on the project (Powerhouse museum). Design report 1 due.
- Week6 (3 September): Mentoring Meeting.
- Week7 (10 September): Mentoring Meeting. Design Report 2 due.
- Week8 (17 September): Mentoring Meeting.
- Week9 (24 September): Public Demonstrations of Prototype 1
- Week10 (8 October): Mentoring Meeting.
- Week11: (15 October): Mentoring Meeting.
- Week12 (22 October): Final deliverables (Demonstrations of Prototype 2, Design Report 3)
- Week13 (29 October): Macquarie Second Year Software Engineering Prize Presentations

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

Resources for Students

For domain knowledge (i.e. infographics), 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 data sources.

Course Evaluation and Development

This course is evaluated each session using the CATEI system.

During the last CATEI evaluation, students have raised many issues related to the clarity of the specifications given. Although every effort is made to produce good specifications, students must appreciate that most workshops projects are open ended and leave room for innovation by students. Therefore, it is important that they seek information about the project requirements stakeholder (this role being played by LIC) on a continuous basis during mentoring sessions. There were many issues with the marking scheme which will be resolved in the next offering.