Course Outline

COMP9323
e-Enterprise Project
Teaching Period 2 2010

Contents

- Course Details
- Course Aims
- Course Staff
- Assumed Knowledge
- Student Learning Outcomes
- Course Schedule
- Assessment
- Academic Honesty and Plagiarism

Course Details

Course Code:	COMP9323
Course Title:	e-Enterprise Project
Units of Credit:	6
Course WebSite:	http://www.cse.unsw.edu.au/~cs9323/
Handbook Entry:	http://www.handbook.unsw.edu.au/postgraduate/courses/2010/COMP9323.html

Course Aims

This course is the third installment of the three-part series of the services engineering stream. It aims at allowing students to pursue a group-based and practical project related to the topics covered in COMP9321 and COMP9322 in an identified area of interests such as e-research, e-learning, e-lifestyle, etc. There are no lectures but the groups will be supervised by a mentor throughout the project design/implementation. The project demonstration will be exhibited to all members of the class at the end of the session.

In this course, we will follow a product-based framework to the project-based learning. The main features of the course are:

- Group-based project related to Web and service-oriented systems
- Learning to develop complex systems through projects
- Projects may focus on an identified area of interests (e.g., e-research, e-learning, e-lifestyle portals)

Course Staff

Contact Details

Staff Name	Role	Email	Phone
Boualem Benatallah	Lecturer in charge	boualem at cse.unsw.edu.au	54767
Adnene Guabtni	Lecturer	aguabtni at cse.unsw.edu.au	56931

Consultations

Staff Name	Day/Time	Location

Assumed Knowledge

Pre-requisites of the Course are:

- Good knowledge of programming and basic knowledge of web technologies
- Good knowledge of design concepts and techniques such as UML class diagrams, ER diagrams, and service-oriented design

These are assumed to have been acquired in the courses COMP9321, COMP9321 and COMP9322

Student Learning Outcomes

After completing this course, students will:

- Learn the process of building a large enterprise system in collaboration with other project members
- Learn the process of developing software in culture of participation
- Learn the process of developing software using agile software engioneering methodologies
- Integrate a number of separate components to build an integrated system
- Develop appropriate testing strategy and methodologies for given projects
- Develop software documentation for various stakeholders

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	yes
a capacity to contribute to, and work within, the international community	yes
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	yes

Course Schedule

There are no lectures for the course. However, the lecturer will hold a class meeting to first introduce the main components of the project and also some specific topics and then to mentor the project groups. This first class meeting will be held on Thursday, 22nd of July 2010 from 6pm to 7pm. Location: K17, Seminar room 113..

This is not a weekly meeting. There are no lectures for COMP9323. There is only one class meeting to first introduce the main components of the project.

Assessment

There will be no mid-term or final exam in this subject. There will be a large project which consists of the following phases:

- Phase one: Brainstorming - Week 3. Brainstorming process evaluated (10% of marking)
- Phase two: Requirement analysis & design - Week 4, 5. Requirement and design documents (20% of marking) are due on week 5.
- Phase three: Implementation - Prototype version 1 - Week 6, 7, 8, 9. A first prototype version is due on week 9. Feedback will be provided by the mentors. This evaluation does not have Impact on the marking of the project. Student's contribution to evaluate and comment on each others projects represents 5% of marking.
- Phase four: Implementation of final version, testing and documenting - Week 10, 11 and 12. A final prototype version is due on week 12. The implementation represents 50% of marking. The user and technical documentation represents 10% of marking.

Finally, student's contribution to the course wiki sharing tools they use etc. represents 5% of marking.

Marking scheme

Marking scheme for the brainstorming document

The objective of this phase is to do a brainstorming assisted by web technologies provided on the project management system.
In order to evaluate each group's brainstorming, you are required to produce a document summarizing how the brainstorming process was performed in your group and what are the outcomes.
The marking scheme for the brainstorming document is the following:

- Initial problem statement (10%): re-formulate using your own words the project's aim.
- Participant selection (20%): describing the process of selecting the core members of the project, factors considered, etc.
- Description of the brainstorming processes (20%)
  - Electronic: the use of wikis, forums, and any other electronic tools to do the brainstorming.
  - Directed: including or not the mentor.
  - Individual: the use of brainstorming on a solitary basis
  - ...
- Resulting questions and ideas (30%): quantity breeds quality. The assumption is that the greater the number of ideas generated, the greater the chance of producing a radical and effective solution.

- Evaluation and selection of the ideas (20%): evaluate the ideas and select the ones that better address the solution to the problem. What metrics did you choose to evaluate ideas? What criteria did you choose to select ideas?

Marking scheme for Requirement and Design documents

The objective of this phase is to produce two documents:

1. The Software Requirements Specifications
2. The Software Design Descriptions

To guide you during this phase, we require these documents to follow the two following standards:

1. IEEE Std 830-1998 : IEEE Recommended Practice for Software Requirements Specifications (available here)
2. IEEE Std 1016-1998 : IEEE Recommended Practice for Software Design Descriptions (available here)

The evaluation of each document will be based on the recommendations of the above two standards.
The marking scheme for each document is the following:

The Software Requirements Specifications (10 % of marking)

- Introduction
  - Purpose (10%)
  - Scope (10%)
  - Definitions, acronyms, and abbreviations (5%)
  - References (5%)
  - Overview (5%)
- Overall description
  - Product perspective (5%)
  - Product functions (15%)
  - User characteristics (5%)
  - Constraints (5%)
  - Assumptions and dependencies (5%)
- Specific requirements (20%) (See 5.3.1 through 5.3.8 in IEEE Std 830-1998 for explanations of possible specific requirements. See also Annex A for several different ways of organizing this section of the SRS.)
- Appendixes (5%)
- Index (5%)

The Software Design Descriptions (10 % of marking)

1. Decomposition description (20%)
2. Dependency description (20%)
3. Interface description (20%)
4. Detailed design description (40%)

The following example of a table of contents shows only one of many possible ways to organize and format the design views and associated information presented in Clause 6 of the IEEE Std 1016-1998 standard.

- Introduction
  - Purpose
  - Scope
  - Definitions, acronyms, and abbreviations
  - References
- Decomposition description
  - Module decomposition
  - Concurrent process decomposition
  - Data decomposition
- Dependency description
  - Intermodule dependencies
  - Interprocess dependencies
  - Data dependencies
- Interface description
  - Module interface
  - Process interface
- Detailed design
  - Module detailed design
    - Module 1 detail
    - Module 2 detail
  - Data detailed design
    - Data entity 1 detail

- Data entity 2 detail

Marking scheme for the user documentation, developer documentation and final implementation

User documentation marking scheme (5 % of marking).
- (80%) Tutorial style for the end-user. This describes each feature of the application and assists the user in realizing these features by providing commented screen-shots. It guides through each step of accomplishing particular tasks. User documentaion quality factors will be understandability (30%), completeness (30%), and conciseness (20%).
- (20%) Features index. User documents need not be organized in any particular way, but it is very important for them to have an index.

Developer (Technical) documentation marking scheme (5 % of marking)
- (40%) Documentation of code (can be auto-generated from comments in the code).
- (40%) Web Service Interfaces and execution scenarios to realize the application features proposed to the end-user.
- (20%) Code compilation and execution environment requirements

Final implementation marking shceme (50 % of marking): this will be based on the quality of the application and its source code according to the following quality factors:
- Completeness (30%): all features planned in the requirement and design documents need to be available.
- Web Services (20%): features proposed need to be available as web services (SOAP or REST).
- Portability (5%): the application needs to run well on any recent web browser (mozilla, IE, Safari)
- Maintainability (5%): the application needs to be easy to update to satisfy new requirements.
- Testability (5%): all features proposed by the application should be fully testable.
- Usability (10%): the application must be easy to use.
- Reliability (10%): the application must not crash during normal use.
- Efficiency (10%): the application does not waste resources
- Security (5%): data must be protected against unauthorized access.

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.