CO2015 Software Engineering Project

Credits: *20* **Convenor:** *Dr R. Craggs* **Semester:** *2nd*

Prerequisites:	*Essential: CO2006, CO2012, CO1003, CO1005, CO1019* *Desirable: CO1012*		
Lectures:	*5 hours*		
Surgeries:	*2 hours*	**Independent Study:**	*135 hours*
Laboratories:	*8 hours*		
Assessment:	*Coursework: 100%*		

Subject Knowledge

Aims According to a recent report of British Computer Society, only above 16% of IT projects can be considered truly successful and over 60% of projects experience sever problems in. The main reason is that software is still developed in an ad hoc way. The purpose of the module is to: provide opportunities for students to develop skills in the analysis, design, specification, implementation, testing and documentation of computer software systems; develop skills that will enhance employment prospects, especially in the IT industry or other numerate disciplines.

Learning Outcomes At the end of this course, successful students will be able to demonstrate participation, according to a role description, in: elicitation and specification of functional and non-functional requirements for a medium-size software system; design, validation and verification of an object-oriented system; construction and testing of specific modules of the system; documentation and demonstration of the system. Moreover, successful students will be able to: recognise important dependencies between the activities mentioned above; critically assess the software life cycle in terms of general quality attributes and viable trade-offs presented within the given problem; employ a configuration management system effectively; schedule and manage a variant of the Unified Process for software development; identify and address risks in the software life cycle; apply appropriate practices within a professional, legal and ethical framework.

Methods Lectures, lab session, workshop, meetings with supervisor, independent research and study.

Assessment Project deliverables, presentation, and demonstration of the software system.

Skills

Aims To develop written and oral communication skills; to provide students with experience of team-based project work; to develop scientific problem solving abilities, along with an appreciation for mathematical and scientific methods.

Learning Outcomes At the end of this course, successful students will be able to: work as a member of a development team, recognising the different roles within a team; conduct significant background research; retrieve information from different sources and manage it effectively; work with uncertain, limited and possibly contradictory information; solve complex problems with other members of the team; communicate in electronic as well as written and oral form; apply management techniques to allocate resources to projects; undertake a risk assessment for a medium-scale team-based software project, especially for risks arising from the use of the resulting software, and to specify appropriate security requirements; formulate and apply suitable tests to assess the security of their software in relation to its requirements; manage their own learning and development including time management and organisational skills as the foundation of on-going professional development.

Methods Lectures, lab session, workshop, meetings with supervisor, textbooks, and web support.

Assessment Project deliverables, presentation, and demonstration of the software system.

Explanation of Prerequisites In order to implement their system students need to be familiar with the basic techniques of programming as taught in CO1003, CO1005, and web/database development as in CO1019.

They will specify and design systems using the software engineering methods taught in CO2006, and also in CO2012.

Module Description

In this course, students will apply software engineering methods that have been studied in the degree programme. Students will work in groups of 5 or 6. Groups will follow the Scrum process to deliver the software project.

Syllabus The practice of software engineering methods; software life-cycle management; risk assessment; definition and prioritization of project goals; requirements elicitation and analysis; system design, construction and testing; quality assurance; requirements verification and validation; configuration management.

Reading List

[B] B. Bruegge and A.H. Dutoit, *Object-Oriented Software Engineering: Using UML, Patterns, and Java; 3rd edition,* Pearson, 2009.

[B] Ken Schwaber and Jeff Sutherland, *The Scrum Guide,*

[C] G. Booch, *Object-Oriented Analysis and Design with Applications, 3rd edition,* Addison-Wesley, 2007.

[C] T. Gilb, *Principles of Software Engineering Management,* Addison-Wesley, 1988.

[C] R. Pressman, *Software Engineering — A Practitioner's Approach, European Adaptation 7th edition,* McGraw Hill, 2010.

[C] I. Sommerville, *Software Engineering; 9th edition,* Addison Wesley, 2010.

Resources Various course notes, web page, books, study guide, handouts.

Module Evaluation Course questionnaires, course review.