

MODULE SPECIFICATION TEMPLATE

SECTION 1: MODULE SPECIFICATIONS

1. Title of the module:
CO510 Software Engineering
2. School which will be responsible for management of the module:
School of Computing
3. Start date of the module:
Academic year 2013/14
4. The number of students expected to take the module:
125
5. Modules to be withdrawn on the introduction of this proposed module and consultation with other relevant Schools and Faculties regarding the withdrawal:
CO531 and CO521
6. Level of the module;
I
7. The number of credits which the module represents:
30 (15 ECTS credits)
8. Which term(s) the module is to be taught in (or other teaching pattern):
Autumn and Spring
9. Prerequisite and co-requisite modules:
CO334, CO320 and CO520 (or equivalent) are pre-requisites.
10. The programme(s) of study to which the module contributes
 - Computer Science, CS(AI), CS(Networks), CS(Consultancy)
 - Applied Computing joint honours programmes,
 - Computing and Business Administration,
 - Web Computing
 - Plus year in industry variants of these programmes.
11. The intended subject specific learning outcomes and, as appropriate, their relationship to programme learning outcomes
 - 11.1. Understand the principles and practice of the development of software systems (broadly defined) – from requirements specification, design, validation, implementation, and evolution [A2, A4];
 - 11.2. Apply design principles and patterns while developing software systems [B5, C1];
 - 11.3. Create UML diagrams for modelling aspects of the domain and the software [A2, B1];
 - 11.4. Design and implement test plans, and apply a wide variety of testing techniques effectively and efficiently [B4, C2];
 - 11.5. Demonstrate the vital role of planning, documentation, estimation, quality, time, cost and risk evaluation in the business context [B1];

- 11.6. Show an understanding of system design, including, design simplicity, appropriateness, and styles of system thinking and focused problem solving [C1, B3, B4, B5];
- 11.7. Show an understanding of the professional and legal duties software engineers owe to their employers, employees, customers and the wider public [B6];
- 11.8. Use the appropriate tools and techniques when working in groups [C4, D1, D5].
12. The intended generic learning outcomes and, as appropriate, their relationship to programme learning outcomes
 - 12.1. Take a professional approach to software development [B6];
 - 12.2. Enhance their ability to work successfully in a group [A2, D1].
 - 12.3. Effectively use IT facilities [D3];
 - 12.4. Manage their own learning and time [D5];
 - 12.5. Be able to analyse a problem specification and to design and implement a solution [B3, B4, D3].
13. A synopsis of the curriculum
 - Introduction to basic design principles of systems;
 - Software process - concepts & implementation:
 - life cycle models (from Extreme Programming to CMM);
 - definition, model, measurement, analysis, improvement of software and team (organization) process;
 - Introduction to modelling principles (decomposition, abstraction, generalization, projection/views), and types of models (information, behavioural, structural, domain, and functional)
 - Basic UML: uses cases, classes, sequence and state diagrams, and OCL.
 - Introduction to design patterns;
 - Introduction to software testing: unit testing, coverage analysis, black box testing, integration testing, testing based on Use Cases, system and acceptance testing, and testing tools.
 - Understanding of a number of business techniques including estimation of time, costs and evaluation of technical alternatives in the business context.
 - Software management: project estimation and metrics, software and process quality assurance, documentation and revision control;
 - Quality assurance;
 - Software engineering tools: configuration control (e.g. SVN, GIT, etc.), project management (e.g. Trac), integrated development environments (e.g. Eclipse, NetBeans, etc.), and a UML tool (e.g. IBM Rational Rose).
 - Project management practice;
 - Risk & risk management in software:
 - risk management: identification, analysis and prioritization
 - software risks: project, process and product
 - development methods for reducing risk
 - Professional practice (reflective):
 - codes of ethics and professional conduct;
 - social, legal, historical, and professional issues and concerns;
 - Design and implement a simple system.
14. Indicative Reading List

UNIVERSITY OF KENT

K. Beck. Extreme Programming Explained: Embrace Change. Addison Wesley. Upper Saddle River, NJ, USA. 2000.

G. Booch, J. Rumbaugh, I. Jacobson. The Unified Modeling Language Users Guide. Addison Wesley. 1999

G. Booch, J. Rumbaugh, I. Jacobson. The Unified Software Development Process. Addison Wesley. 1999.

P. Coad, E. Lefebvre, J. De Luca. JAVA Modeling in Color with UML: Enterprise Components and Process. Prentice Hall. 1999.

A. Cockburn. Writing Effective Use Cases. Addison-Wesley. Boston, Ma, USA. 2001.

E. M. Hall. Managing Risk: Methods for Software Systems Development. Addison-Wesley. Reading, MA, USA. 1998.

D. G. Johnson, H. Nissenbaum. Computers, Ethics and Social Values. Prentice-Hall. 1995

E. A. Kallman, J. P. Grillo. Ethical Decision Making and Information Technology: An Introduction with Cases. 3rd Edition. McGraw-Hill. 1999

D. Kulak, E. Guiney. Use Cases: Requirements in Context. Addison-Wesley. Boston, Ma, USA. 2000.

J. Newkirk, R. C. Martin. Extreme Programming in Practice. Addison Wesley. Upper Saddle River, NJ, USA. 2001.

Mauro Pezze, Michal Young. Software Testing and Analysis: Process, Principles and Techniques. John Wiley & Sons. 2007.

R. Pooley, P. Stevens. Using UML Software Engineering with Objects and Components. Addison-Wesley. 2001.

G. Schneider, J. P. Winters. Applying Use Cases: A Practical Guide. Addison-Wesley. 2001.

I. Sommerville. Software Engineering.9th Edition. Harlow, England, UK. 2010.

15. Learning and Teaching Methods, including the nature and number of contact hours and the total study hours which will be expected of students, and how these relate to achievement of the intended learning outcomes

The module will comprise 300 hours of study over two teaching terms and the exam term. There will be 44 lectures and 22 terminal sessions. In addition, students will need to spend 64 hours on private study and materials, 90 hours on exercises and assessments, and 80 hours on pre-examination revision.

The overall structure of the module is that a number of theoretical aspects of software engineering are introduced in the first term, and then a substantial part of the second term is used for the students to work on a group based software engineering project.

The lectures initially serve to introduce the relevant issues and terminology, often on the basis of interactive discussion of illustrative examples. Lectures will be accompanied by course notes or textbooks, with some topics supplemented by directed reading and exercise sheets. The students will acquire the listed skills by participating in exercises and assessments, in many cases using software tools. Feedback on their work and further lectures will then concentrate on directing the students' reflection on their own practical work, leading to insights and skills at a more abstract level.

16. Assessment methods and how these relate to testing achievement of the intended learning outcomes

Assessment is through a combination of unseen written examination (50%) [11.1, 11.2, 11.3, 11.7], and assessed coursework (50%). The main piece of the coursework will be

UNIVERSITY OF KENT

used to assess the students' practical skills in developing a medium size software system as a group project (40%) [11.1, 11.2, 11.3, 11.4, 11.5, 11.6, 11.8, 12.1, 12.2, 12.3, 12.4, 12.5]. In addition to this main piece, there will be several small formative assessments (10%) [11.3, 11.4, 11.8].

The written examination will test the students' skills and understanding at a deeper level, concentrating on:

- the skills that have not been assessed in coursework already;
- the students' reflective understanding of the techniques and their applicability on how to develop software systems;

17. Implications for learning resources, including staff, library, IT and space

Several members of the School of Computing will be able to teach this module or parts of this module. The library holds several copies of some of the listed books. A number of relevant tools are available for free, on different platforms. There are no particular implications for space.

18. The School recognises and has embedded the expectations of current disability equality legislation, and supports students with a declared disability or special educational need in its teaching. Within this module we will make reasonable adjustments wherever necessary, including additional or substitute materials, teaching modes or assessment methods for students who have declared and discussed their learning support needs. Arrangements for students with declared disabilities will be made on an individual basis, in consultation with the University's disability/dyslexia support service, and specialist support will be provided where needed.

19. Campus(es) where module will be delivered¹

Canterbury

If the module is part of a programme in a Partner College or Validated Institution, please complete the following:

20. Partner College/Validated Institution

21. University School (for cognate programmes) or Faculty (for non-cognate programmes) responsible for the programme

SECTION 2: MODULE IS PART OF A PROGRAMME OF STUDY IN A UNIVERSITY SCHOOL
Statement by the School Director of Learning and Teaching/School Director of Graduate Studies (as appropriate): "I confirm I have been consulted on the above module proposal and have given advice on the correct procedures and required content of module proposals"

.....

Director of Learning and Teaching/Director of
Graduate Studies (delete as applicable)

.....

Date

.....

Print Name

Statement by the Head of School: "I confirm that the School has approved the introduction of the module and, where the module is proposed by School staff, will be responsible for its resourcing"

¹ Required for information purposes only. Changes of campus will not require re-approval of the module specification.

UNIVERSITY OF KENT

.....
Head of School

.....
Date

.....
Print Name

**SECTION 3: MODULE IS PART OF A PROGRAMME IN A PARTNER COLLEGE OR
VALIDATED INSTITUTION**

(Where the module is proposed by a Partner College/Validated Institution)

Statement by the Nominated Officer of the College/Validated Institution (*delete as applicable*): "I confirm that the College/Validated Institution (*delete as applicable*) has approved the introduction of the module and will be responsible for its resourcing"

.....
Nominated Responsible Officer of Partner
College/Validated Institution

.....
Date

.....
Print Name

.....
Post

.....
Partner College/Validated Institution