

# Module Information

Module Identifier	<b>CS22120</b>
Module Title	<b>The Software Development Life Cycle</b>
Academic Year	<b>2014/2015</b>
Co-ordinator	<b><u>Mr Nigel William Hardy</u></b> <b><u>(mailto:nwh@aber.ac.uk?subject=CS22120)</u></b>
Semester	<b>Semester 2 (Taught over 2 semesters)</b>
Mutually Exclusive	<b><u>CC21100 (?m=CC21100)</u></b>
Mutually Exclusive	<b><u>CC21120 (?m=CC21120)</u></b>
Other Staff	<b><u>Professor Christopher John Price</u></b> <b><u>(mailto:cjp@aber.ac.uk?subject=CS22120)</u></b> <b><u>Mr Nigel William Hardy</u></b> <b><u>(mailto:nwh@aber.ac.uk?subject=CS22120)</u></b> <b><u>Mr Neil Scott Taylor (mailto:nst@aber.ac.uk?subject=CS22120)</u></b> <b><u>Mr Christopher William Loftus</u></b> <b><u>(mailto:cwl@aber.ac.uk?subject=CS22120)</u></b> <b><u>Dr Lynda Ann Thomas (mailto:ltt@aber.ac.uk?subject=CS22120)</u></b> <b><u>Dr Wayne Aubrey (mailto:waa2@aber.ac.uk?subject=CS22120)</u></b> <b><u>Dr Bernard Paul Tiddeman</u></b> <b><u>(mailto:bpt@aber.ac.uk?subject=CS22120)</u></b> <b><u>Dr Jun He (mailto:jqh@aber.ac.uk?subject=CS22120)</u></b> <b><u>Dr Neal Snooke (mailto:nns@aber.ac.uk?subject=CS22120)</u></b>

## Course Delivery

<b>Delivery Type</b>	<b>Delivery length / details</b>
Lecture	18 hours

## Assessment

<b>Assessment Type</b>	<b>Assessment length / details</b>	<b>Proportion</b>
Semester Assessment	Assessed coursework. Essay and demonstration about a comparison of different image processing techniques	100%

## Learning Outcomes

The major learning outcome of this module is that the student should:

1. be able to participate in an industrial scale project.

In addition, on successful completion of this module, students should be able to:

2. apply the elements of the software life cycles, contrast a range of life cycle models and choose appropriate models for a range of typical projects;
3. be able to apply software quality procedures and convince others of their value;
4. be able to use version and configuration management and convince others of their value;
5. produce the key deliverables in software life cycles.
6. evaluate ethical situations and make professional judgements on them.

## Brief description

The objectives of the lecture course are first to introduce students to the best traditional practices for the specification, design, implementation, testing and operation of large software systems; and second to provide a framework for the more detailed material on design which is taught in other courses. The practical work is a group project. Students on different degree schemes may be given different projects.

Long thin module. No lectures in Semester 2

The general tutorial system for Year 2 honours students is administered through this module.

## Aims

This module aims to introduce students to the basic principles of software engineering and to give them experience of developing a software system in a team. Specifically, it aims to:

- expose students to best practices in the engineering activities of project management, quality assurance and standards compliance;
- enable students to identify and employ appropriate practices for the specification, design, testing and operation of large software systems;
- provide a framework for the discipline of software engineering, including the more detailed material on design and implementation that is taught in other courses;
- involve students in the development of a piece of software which approximates as closely as possible in the university environment the software development conditions found in industry.

## Content

### 1. Introduction- 1 Lecture

The approach and the obligations of the professional engineer. Software as an engineering artifact. Analogies between software and other branches of engineering.

### 2. The Software Life Cycle - 3 Lectures

Description of the phases of a range of software life cycles (including the Waterfall, Prototyping, Extreme Programming and Spiral models) and the major deliverables and activities associated with each phase. Software process improvement.

### 3. Project Management - 2 Lectures

Planning and cost estimation. Progress monitoring. Team structure and team management.

**4. Quality Management - 2 Lectures**

Validation, verification and testing. Quality plans. Walkthroughs, code inspections and other types of review. Role of the quality assurance group. Standards (international, national and local).

**5. Configuration Management - 2 Lectures**

Baselines. Change control procedures. Version control. Software tools to support configuration management.

**6. Requirements Engineering and HCI - 2 Lectures**

The IEEE standard for requirements specifications. Validation of requirement by e.g., prototyping. Deficiencies in the traditional approach to requirements. Introduction to UML Use cases. An introduction to HCI.

**7. Design - 2 Lectures**

Outline (architectural) design and detailed design. Use of abstraction, information hiding, functional and hierarchical decomposition at levels higher than the individual program. Contents of design documentation. State diagrams. Relevant UML notations: packages, sequence and activity diagrams, active objects.

**8. Implementation and maintenance - 2 Lectures**

Choice of language. Cutover. Types of maintenance. Maintenance process. Refactoring.

**9. Testing - 2 Lectures**

Testing strategies. Testing tools: static and dynamic analysers, test harnesses and test data generators, simulators. Performance testing. Regression testing. User documentation and training.

**10. Tutorials**

A weekly tutorial will be associated with this course. The tutorial will be used to organise group project activities and to discuss software engineering issues.

**Module Skills**

<b>Skills Type</b>	<b>Skills details</b>
Application of Number	No.
Communication	Written skills will be needed to complete supporting documents to accompany coursework.
Improving own Learning and Performance	See 2 above.
Information Technology	The whole module concerns this area.
Personal Development and Career planning	Careful time management will be needed as so to enable students to complete coursework etc.
Problem solving	This is inherent in both the group project and examined material.
Research skills	The students will need to search for and use relevant technical information while completing coursework.
Subject Specific Skills	Yes. See module title and content.
Team work	Yes. Fundamental to module.

**Reading List****Recommended Text**

Ian Sommerville (2010) *Software Engineering*. <http://voyager.aber.ac.uk/cgi-bin/Pwebrecon.cgi?DB=local&CNT=50+records+per+page&FT=0321210263> (<http://voyager.aber.ac.uk/cgi-bin/Pwebrecon.cgi?DB=local&CNT=50+records+per+page&FT=0321210263>) 9th Ed. Pearson Education

Roger S. Pressman (2004) *Software Engineering: A practitioner's approach*. <http://voyager.aber.ac.uk/cgi-bin/Pwebrecon.cgi?DB=local&CNT=50+records+per+page&FT=0071238409> (<http://voyager.aber.ac.uk/cgi-bin/Pwebrecon.cgi?DB=local&CNT=50+records+per+page&FT=0071238409>) 7th McGraw-Hill

## Notes

This module is at [CQFW](http://wales.gov.uk/topics/educationandskills/qualificationsinwales/creditqualificationsframework/?lang=en) (<http://wales.gov.uk/topics/educationandskills/qualificationsinwales/creditqualificationsframework/?lang=en>) Level 5