UNIVERSITY OF LONDON | INTERNATIONAL PROGRAMMES

Course information 2018–19

IS3139 Software engineering: theory and application

This course covers the methods, values, attitudes and techniques in software systems. It provides an understanding of the need for rigour, and enables students to select and apply a relevant methodological approach to the development of well designed and documented systems.

Prerequisite

If taken as part of a BSc degree, courses which must be passed before this course may be attempted:

IS2062 Information systems development and management **and**
IS2138 Information and communication technologies: principles and perspectives.

Aims and objectives

The main aims of this course are to:

- introduce students to the overarching concerns of software engineering practice
- provide an understanding of the various processes software engineers may employ in developing software
- develop an understanding of the tools and techniques employed in contemporary software engineering
- provide students with an experience of the way techniques are applied in practice (this is achieved through the individual project work)
- develop the capacity to identify relevant approaches to software engineering.

Assessment

This course is assessed by a three-hour unseen written examination and coursework. See ‘Coursework and assessment’ overleaf.

Learning outcomes

At the end of this course and having completed the essential reading and activities students should be able to:

☑ describe the expectations, pressures and problems faced in developing software and the need for processes, tools, techniques and approaches

☑ outline the underlying processes of software engineering and critically assess relevant approaches

☑ analyse, design, test and maintain software systems and document these actions correctly.

Essential reading

For full details, please refer to the reading list

Pressman, R.S. *Software Engineering: A Practitioner’s Approach.* Adapted by D. Ince. (London: McGraw Hill)

Sommerville, I. *Software Engineering*. (Wokingham: Addison Wesley)

Students should consult the appropriate *EMFSS Programme Regulations*, which are reviewed on an annual basis. The *Regulations* provide information on the availability of a course, where it can be placed on your programme’s structure, and details of co-requisites and prerequisites.

Syllabus

This is a description of the material to be examined. On registration, students will receive a detailed subject guide which provides a framework for covering the topics in the syllabus and directions to the essential reading

This syllabus covers the methods, attitudes and values which underlie professional contemporary software systems development. The emphasis is on how to undertake formal software development through requirements specification, design and implementation, but within a broader understanding of software engineering practices.

Section 1: Software Engineering Process

- The changing pressures on software engineering practices: History of the field, definition of software, the software crisis.
- The Process for Developing Software and its importance
- The Capability Maturity Model
- The traditional software engineering process: The lifecycle model, evolutionary software development, incremental software development, spiral model. Prototyping
- Rapid software development
- Internet speed web based application development
- End-user development.
- Agile methods
- Extreme programming
- Refactoring

Section 2: The Practices of software engineering:

Introduction to Structured vs. object oriented paradigms.
Acquiring requirements
Specifying requirements and design (both structured and object oriented)

- Structured approaches: ER diagram, Data flow diagrams, Data dictionary
- OO approach (Using UML): Use-case diagrams, Class diagrams, Object sequence diagrams, State-chart diagrams

Features of good design
Coding and configuration management
Implementation and testing (both structured and object oriented)

- Choice of programming languages and techniques
- Test planning
- White-box and black-box testing
- Testing automation
- Implementation

Maintenance and software evolution

- Systems re-engineering for Legacy systems

Reuse

- Reasons for reuse.
- Concept reuse – patterns, configurable systems products and program generators.
- Component-based software engineering.

Computer Aided Software Engineering (CASE) Tools.
Documentation and Help Systems.
Project management in software engineering.
Documentation and help systems.
Managing Software Engineering Projects.

Coursework and assessment

The examination will be three hours and consist of seven questions of which students must answer four. This will contribute 60% to the overall mark.

Students will undertake a project for the remaining 40% of marks. This will require them to review a software engineering approach or technique, apply this to a real problem and finally reflect on the experience. The project will require them to write a short essay describing the approach they are intending to apply and its relevance to the problem chosen. They will then present an account of how the approach was applied in practice. This should include the relevant documentary material required for the chosen approach (e.g. if the Rational Unified Process is chosen UML diagrams and various documents should be included).
Finally they will need to complete a two page pro forma in which they will reflect on the practice of developing the system from the approach chosen. This should include lessons learnt and critical reflections on the process. A bibliography must also be provided demonstrating reading beyond the core textbooks.

Students are not required to produce programming code, but rather are assessed on their attempt to apply software engineering techniques and principles in practice. The focus is not on the produced system but on the quality of the process undertaken, the coherence of the documents presented and how successful the documents would be in developing a software system. Clearly however, for some students programming will be an integral part of this exercise.