
CO2006 Software Engineering and System Development

Credits: 20 **Convenor:** Dr A. Boronat, Dr J.O. Ringert **Semester:** 1st

Prerequisites: *Essential:* CO1003, CO1005, CO1019
 Desirable: CO1001, CO1012

Lectures: 6 hours

Laboratories: 58 hours

Class Test Hours: 2 hours

Independent Study: 84 hours

Assessment: *Coursework:* 100%

Subject Knowledge

Aims According to a report of the British Computer Society, only about 16% of IT projects can be considered truly successful and over 60% of them experience severe problems. The difficulties of software development led to the coining of the phrase “the software crisis” and the birth of software engineering as a discipline. However, in many companies, software is still developed in an ad-hoc way. The purpose of this module is to teach object-oriented methods for analysis, specification, design, implementation, and testing of software systems.

Learning Outcomes At the end of this course, successful students will be able to: explain the main phases in a software development process; analyse customer requirements following an agile methodology; produce object-oriented system designs, by applying design patterns and architectural styles; use UML for consistent specification of software systems and business processes; incorporate security into specifications and designs by following a flexible security specification process; and use appropriate techniques for software development and testing, including mechanisms for software reuse.

Methods Lectures, lecture notes, surgeries, recommended textbooks, worksheets, online videos, supervised laboratories, VLE discussion board, GitHub, formative feedback and web resources.

Assessment Formative coursework, online quizzes, assessed class tests and mini project.

Skills

Aims To teach students a range problem-solving skills tailored to SE, including knowledge acquisition and software modelling.

Learning Outcomes At the end of this course, successful students will be able to use short, clear summaries of technical knowledge; solve abstract and concrete problems (both routine seen, and simple unseen).

Methods Lectures, lecture notes, surgeries, recommended textbooks, worksheets, online videos, supervised laboratories, VLE discussion board, GitHub, formative feedback and web resources.

Assessment Formative coursework, online quizzes, assessed class tests and mini project.

Explanation of Prerequisites A sound knowledge of basic algorithms, data structures and programming is required. Some knowledge of database systems of basic web application development (HTML, CSS, Javascript) is desirable.

Module Description This module introduces students to principles and methods used to specify, design, implement and test software systems. In particular, the object-oriented paradigm will be followed, and techniques therein.

Syllabus

Introduction: Introduction to software engineering; the inherent complexity of software; examples of complex systems; basic notions and techniques for modelling software systems, build automation with Gradle, Groovy, agile principles; tasks; dependencies; repositories.