Engineering Tripos Part IIB, 4M21: Software Engineering and Design, 2018-19

Module Leader
Dr E Punskaya [1]

Lecturer
Dr E Punskaya

Timing and Structure
Lent term. 16 lectures (including integrated examples classes). Assessment: 100% exam.

Objectives
As specific objectives, by the end of the course students should be able to:

- Understand the benefits of object-oriented analysis and design, its concepts and processes.
- Be familiar with formal design tools for object oriented design and analysis.
- Recognise and understand some frequently used design patterns.
- Be aware of the process involved in user interface design.
- Understand software development methodologies.
- Understand the main issues and processes necessary to achieve effective software product development.

Content
Software forms an important part of many modern engineering products, from telecommunications to automotive to internet-based systems. This course will provide an understanding of the technical and management processes involved in the design of software systems. Software engineering concepts are considered at a range of scales, from the manipulation of object-orientated concepts, through architectural design components, to the building of large complex software systems.

Software Design (10L)
- Concepts Behind Software Design: managing complexity of the software systems and minimizing risks.
- Object-Oriented Software Design Principles: abstraction, problem partitioning, how to identify components.
- Object-Orientated Design and Analysis: classes and objects, encapsulation and data hiding, abstraction and inheritance, polymorphism.
- Formal Tools: introduction to UML.
- Design Patterns: frequently occurring design techniques and their role in building systems.
- Principles of Good Design and User Interface Design: designing experiences, designing for the user, use cases, process and main elements, usability.

Software Systems and Engineering (6L)
- Quality Assurance and Risk Management: testing, automated testing, tools.
- Software Development Methodologies: from waterfall to agile programming.
- Software Management: project lifecycle, source code control, code reviews, testing, the nightly build, release management, maintenance and refactoring, organising software teams.
Software Innovation and Entrepreneurship.

Booklists

Please see the [Booklist for Group M](https://www.vle.cam.ac.uk/mod/book/view.php?id=364101&chapterid=56281) Courses for references for this module.

Examination Guidelines

Please refer to [Form & conduct of the examinations](http://teaching.eng.cam.ac.uk/content/form-conduct-examinations).

Last modified: 17/05/2018 15:02