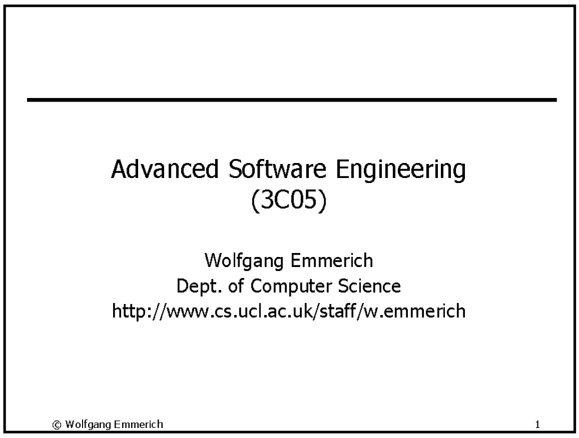
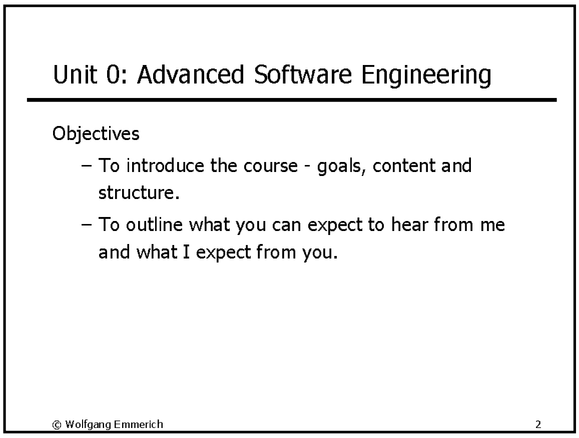


Advanced Software Engineering (3C05)

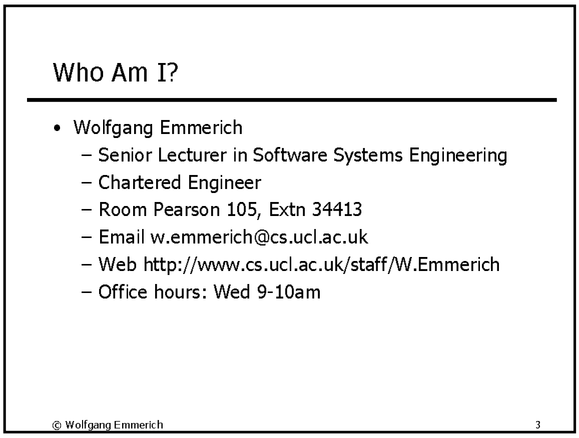
Wolfgang Emmerich
Dept. of Computer Science
http://www.cs.ucl.ac.uk/staff/w.emmerich



Unit 0: Advanced Software Engineering

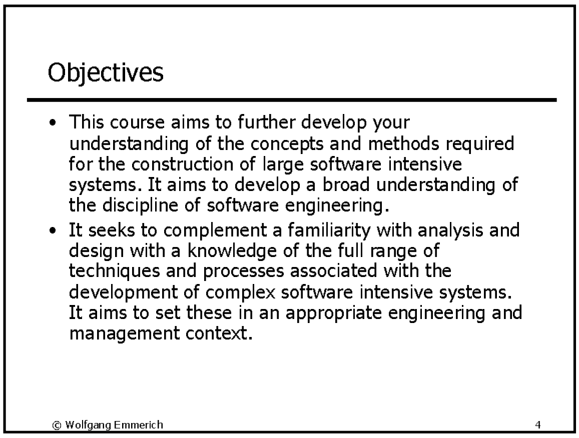
Objectives

- To introduce the course - goals, content and structure.
- To outline what you can expect to hear from me and what I expect from you.



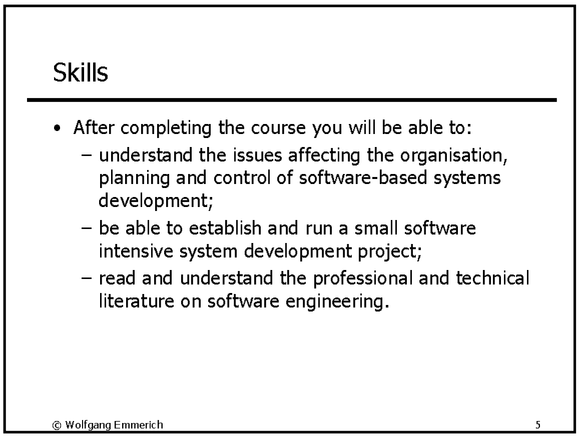
Who Am I?

- Wolfgang Emmerich
  - Senior Lecturer in Software Systems Engineering
  - Chartered Engineer
  - Room Pearson 105, Extn 34413
  - Email w.emmerich@cs.ucl.ac.uk
  - Web http://www.cs.ucl.ac.uk/staff/W.Emmerich
  - Office hours: Wed 9-10am



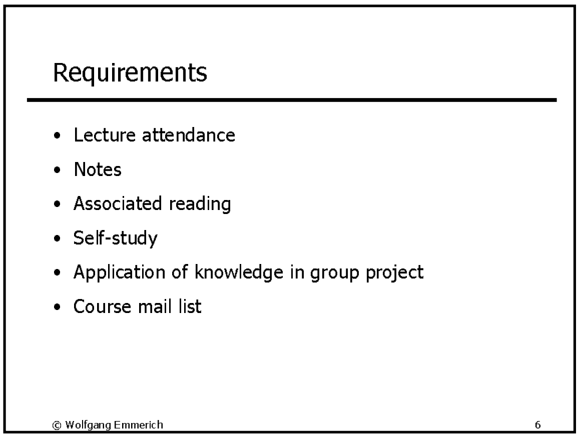
Objectives

- This course aims to further develop your understanding of the concepts and methods required for the construction of large software intensive systems. It aims to develop a broad understanding of the discipline of software engineering.
- It seeks to complement a familiarity with analysis and design with a knowledge of the full range of techniques and processes associated with the development of complex software intensive systems. It aims to set these in an appropriate engineering and management context.



Skills

- After completing the course you will be able to:
  - understand the issues affecting the organisation, planning and control of software-based systems development;
  - be able to establish and run a small software intensive system development project;
  - read and understand the professional and technical literature on software engineering.



Requirements

- Lecture attendance
- Notes
- Associated reading
- Self-study
- Application of knowledge in group project
- Course mail list

Assessment

- 10% coursework, 90% examination
- 4 examination questions
  - 1 compulsory (Part I) - 34%
  - 2 from 3 (Part II) - 66%
- 2 courseworks
  - Each worth 50%
- plus link to project and other work

Books

The Future of Software Engineering
edited by Anthony Finkelstein
ACM Press, 386 pages. July 2000. ISBN 01-58113-253-0.
Can be ordered from http://computer.org or http://www.acm.org
Papers are available at http://www.softwaresystems.org

I will make reference to this book!

Books

Software Engineering (International Computer Science Series)
by Ian Sommerville
Hardcover - 742 pages 5th edition (November 1995)
Addison-Wesley Pub Co; ISBN: 0201427656

Software Engineering : A Practitioner's Approach
by Roger S. Pressman
Hardcover - 852 pages 4th edition (August 1996)
McGraw Hill College Div; ISBN: 0070521824

You are advised to purchase one of these for reference

The Mythical Man-Month : Essays on Software Engineering
by Frederick P., Jr. Brooks, Frederick P. Brooks Jr
Paperback - 322 pages anniversary edition (July 1995)
Addison-Wesley Pub Co; ISBN: 0201835959

Read this book

Further Books for specific subjects

Course Structure

- Unit 0 – This overview (today)
- The Wider Software Engineering Context
  - Unit 1 – Systems Engineering (3/10)
  - Unit 2 – Project Planning & Scheduling (4/10)
    Set coursework 1
  - Unit 3 – Risk Management (8/10)
  - Unit 4 – Standards (10/10)

We reserve the right to change structure at any time and without notice

Course Structure

- Advanced Software Engineering Process Topics
  - Unit 5 – Unified Software Development Process (12/10)
  - Unit 6 – Software Development Team Structures (15/10)
  - Unit 7 – Software Quality (22/10)
  - Unit 8 – Software Process Improvement (24/10)
  - Unit 9 – Requirements Engineering (22/10)
  - Unit 10 – Configuration Management (24/10)
    Set coursework 2
  - Unit 11 – Software Architecture (12/11)
  - Unit 12 – Design Pattern (14/11)
  - Unit 13 – Component-based Software Engineering (15/11)
    Submit coursework 1

Course Structure (Cont'd)

- Seminars
  - Unit 14 - Pattern-oriented Software Architecture (26/11): Menand, Jig, Roman, Ravin
  - Unit 15 – UML Extension Mechanisms (28/11): Jahed, Jasmin, Shahzad, Martin, Aradna
  - Unit 16 – Object Constraint Language (29/11): Ruben, Riaz, Steven, Sujay, Arif, Alice
  - Unit 17 – Model Checking (3/12): Sam, Martin, Wilfred, Benjit, Dragana
  - Unit 18 – Distributed SW Architectures using Middleware (5/12): Mark, Robert, Peter, Leon
  - Unit 19 – Distributed Objects and Components (6/12): Chris, Kevin
  - Unit 20 – Model Driven Architecture (10/12): Raj, Jag, Zul, Anish
  - Unit 21 – Enterprise Application Integration Techniques (12/12): Neil, Jim, Mark, Bungy, Christoph
  - Unit 22 – Mobile Computing (13/12): Alissa, Toan, Ian, Alex, Wissam
  - Submit coursework 2

Notes

Course notes will be handed out before each lecture

for revision!

Definition

Software engineering is the branch of systems engineering concerned with the development of large and complex software intensive systems. It focuses on: the real-world goals for, services provided by, and constraints on such systems; the precise specification of system structure and behaviour, and the implementation of these specifications; the activities required in order to develop an assurance that the specifications and real-world goals have been met; the evolution of such systems over time and across system families. It is also concerned with the processes, methods and tools for the development of software intensive systems in an economic and timely manner.

Software Engineering

- Is not a static discipline, there are unresolved debates and controversies. Many of the topics we will cover are the subject of considerable ongoing research.
- Do not expect cut and dried answers to your questions. Expect working solutions, approximations, rules of thumb and indications of best practice.

Other Courses

- Software Engineering I
- Group Project

WARNING
I will work on the assumption that you have a familiarity with these, but will try to ensure material is synchronised.

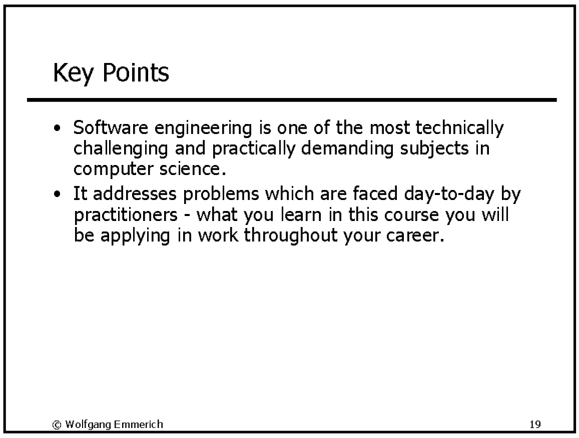
The Software Engineering Agenda …

- Scaling-up does not work
  - not easily understood by one person
  - communication overhead
  - effect of changes not obvious
  - need for discipline, documentation and management

Note:
It is very important that you keep the problems of scale and complexity firmly in mind throughout the course.

The "Trust Issue"

- I cannot give you large examples, if we do the "clerical work" would exceed the time you have available for the course.
- The examples we give you could probably be handled without the techniques we are showing you.
- You lack experience of large systems therefore you will have to take some of what we are saying on trust.
- We will try and give examples, if in doubt - ask!



Key Points

- Software engineering is one of the most technically challenging and practically demanding subjects in computer science.
- It addresses problems which are faced day-to-day by practitioners - what you learn in this course you will be applying in work throughout your career.