


**Advanced Software Engineering  
(3C05/D22)**

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

 **UCL** Computer Science 1

---

---

---

---

---

---


---

---

**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.

 **UCL** Computer Science 2

---

---

---

---

---


---

---

---

**Who Am I?**

- Wolfgang Emmerich
  - Professor of Distributed Computing
  - Head of Software Systems Engineering Research Group
  - Chartered Engineer
  - Room: Malet Place Engineering Building 7.20, Extn 34413
  - Email: [w.emmerich@cs.ucl.ac.uk](mailto:w.emmerich@cs.ucl.ac.uk)
  - Web: <http://www.cs.ucl.ac.uk/staff/W.Emmerich>

 **UCL** Computer Science 3

---

---

---

---

---

---

---

---

## 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: make sure you are registered on 3c05



---

---

---

---

---


---

---

---

**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


**UCL Computer Science**
7

---

---

---

---

---


---

---

---

**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>


**UCL Computer Science**
8

---

---

---

---

---


---

---

---

**Books**

<p>Software Engineering (International Computer Science Series)          by Ian Sommerville          Hardcover - 742 pages 5th edition (November 1995)          Addison-Wesley Pub Co; ISBN: 0201427656</p> <p>Software Engineering : A Practitioner's Approach          by Roger S. Pressman          Hardcover - 852 pages 4th edition (August 1996)          McGraw Hill College Div; ISBN: 0070521824</p> <p>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</p> <p>Further Books for specific subjects</p>	<p>} You are          advised to          purchase one          of these for          reference</p>
--	---


**UCL Computer Science**
9

---

---

---

---

---

---


---

---

**Course Structure**

- Unit 0 – This overview (today)
- **The Wider Software Engineering Context**
  - Unit 1 – Systems Engineering (this afternoon)
  - Unit 2 – Project Planning & Scheduling (16/1)
    - Set coursework 1
  - Unit 3 – Risk Management (21/1)
  - Unit 4 – Standards (27/1)

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


**UCL** ComputerScience 10

---

---

---

---

---


---

---

---

**Course Structure**

- **Advanced Software Engineering Process Lectures**
  - Unit 5 – Unified Software Development Process (27/1+30/1)
  - Unit 6 – eXtreme Programming (3/2)
  - Unit 7 – Software Development Team Structures (6/2)
  - Unit 8 – Software Quality (10/2)
  - Unit 9 – Software Process Improvement (13/2)
    - Set coursework 2
  - Unit 10 – Requirements Engineering (27/2)
  - Unit 11 – Testing & Inspections (2/3)
  - Unit 12 – Component-based Software Engineering (2/3)
    - Submit coursework 1


**UCL** ComputerScience 11

---

---

---

---

---


---

---

---

**Course Structure (Cont'd)**

- **Advanced Software Engineering Process Seminars**
  - Unit 14 - Pattern-oriented Software Architecture (Sarvinderjitt Singh, Ibrahim Rahman):
  - Unit 15 – UML Extension Mechanisms (Xianzhe Jin, Oscar Kozlowski):
  - Unit 16 – Object Constraint Language (Tim Schooley):
  - Unit 17 – Model Checking (Marios Kallis, Boris Feigin):
  - Unit 18 – Program Slicing (Nick Cameron,Umut Atabek):
  - Unit 19 – Distributed SW Architectures using Middleware (Miitul Patel, David Nguyen):
  - Unit 20 – Distributed Objects and Components (Loic Chabardes):
  - Unit 21 – Model Driven Architecture (Michael Le, Sunil Laungani, Sukhwinder Cheema, Jeanesh Patel):
  - Unit 22 – Enterprise Application Integration Techniques (Florence Lin, Jennifer Li):
  - Unit 23 – Mobile Computing (Boris Vilidnitsky, John Tang):
  - Submit coursework 2


**UCL** ComputerScience 12

---

---

---

---

---


---

---

---

**Notes**

- Lecture notes available from JJ Giwa from next week
- Seminar notes
  - Will be distributed before each lecture
  - This assumes that you submit your presentation to me a week before the scheduled seminar!

 **UCL** ComputerScience 13

---

---

---

---

---


---

---

---

**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.

 **UCL** ComputerScience 14

---

---

---

---

---


---

---

---

**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.

 **UCL** ComputerScience 15

---

---

---

---

---

---


---

---

**Other Courses**

- Software Engineering (2B15/D101)
- Group Project (3C02)

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

 **UCL** ComputerScience 16

---

---

---

---

---

---


---

---

**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.

 **UCL** ComputerScience 17

---

---

---

---

---


---

---

---

**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!

 **UCL** ComputerScience 18

---

---

---

---

---

---

---

---

## 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.



---

---

---

---

---

---

---

---