Lecture 1 – Introduction to Software Engineering

Karl R. Wilcox
K.R.Wilcox@reading.ac.uk

Objectives Of The Course

- To understand the concepts and practice of software engineering, lifecycles, models and techniques
  - Strengths, weaknesses and how they are best applied
- To evaluate the maturity of software development as an engineering discipline
  - Art, craft or industry?

The Three Themes

- Software development as a process
  - What activities are part of software engineering?
  - What methods and techniques are used to develop software?
- The management of software development
  - How do we organise and control the various activities?
  - What happens when people get involved?
- Special Techniques of software development
  - What about real-time systems? Legacy integration?
  - What are formal methods and how are they used?

Methods

- Weekly Lecture
- Followed by Seminar (not every week)
- Short Assignment – October
- Longer Assignment
  - Set October
  - Due January
- Use ‘Blackboard’ and e-mail please!

Assessment

- Part 1 (practice)
  - Looking at the differences between a program and a "system"
  - List of bullet points
- Part 2 (30%)
  - Case Study of a real IT Project
  - Your own investigation & choice of project
  - Analysis of s/w eng. techniques

My Position on Software Engineering

Software engineering

- The economies of ALL developed nations are dependent on software
- More and more systems are software controlled
- Software engineering is concerned with theories, methods and tools for professional software development
- Software engineering expenditure represents a significant fraction of GNP in all developed countries

Software costs

- Software costs often dominate system costs. The costs of software on a PC are often greater than the hardware cost
- Software costs more to maintain than it does to develop. For systems with a long life, maintenance costs may be several times development costs
- Software engineering is concerned with cost-effective software development

What is software?

- Computer programs and associated documentation
- Software products may be developed for a particular customer or may be developed for a general market
- Software products may be
  - Generic - developed to be sold to a range of different customers
  - Bespoke (custom) - developed for a single customer according to their specification

What is software engineering?

- Software engineering is an engineering discipline which is concerned with all aspects of software production
- Software engineers should adopt a systematic and organised approach to their work and use appropriate tools and techniques depending on the problem to be solved, the development constraints and the resources available

What is the difference between software engineering and computer science?

- Computer science is concerned with theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software
- Computer science theories are currently insufficient to act as a complete underpinning for software engineering

What is the difference between software engineering and system engineering?

- System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this process
- System engineers are involved in system specification, architectural design, integration and deployment

What is a software process?

- A set of activities whose goal is the development or evolution of software

- Generic activities in all software processes are:
  - Specification - what the system should do and its development constraints
  - Development - production of the software system
  - Validation - checking that the software is what the customer wants
  - Evolution - changing the software in response to changing demands

What is a software process model?

- A simplified representation of a software process, presented from a specific perspective

- Examples of process perspectives are
  - Workflow perspective - sequence of activities
  - Data-flow perspective - information flow
  - Role/action perspective - who does what

- Generic process models
  - Waterfall
  - Evolutionary development
  - Formal transformation
  - Integration from reusable components

What are the costs of software engineering?

- Roughly 60% of costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs
- Costs vary depending on the type of system being developed and the requirements of system attributes such as performance and system reliability
- Distribution of costs depends on the development model that is used

What are software engineering methods?

- Structured approaches to software development which include system models, notations, rules, design advice and process guidance
- Model descriptions
  - Descriptions of graphical models which should be produced
- Rules
  - Constraints applied to system models
- Recommendations
  - Advice on good design practice
- Process guidance
  - What activities to follow

What is CASE (Computer-Aided Software Engineering)

- Software systems which are intended to provide automated support for software process activities. CASE systems are often used for method support

- Upper-CASE
  - Tools to support the early process activities of requirements and design

- Lower-CASE
  - Tools to support later activities such as programming, debugging and testing

What are the attributes of good software?

- The software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable
- Maintainability
  - Software must evolve to meet changing needs
- Dependability
  - Software must be trustworthy
- Efficiency
  - Software should not make wasteful use of system resources
- Usability
  - Software must be usable by the users for which it was designed

What are the key challenges facing software engineering?

- Legacy systems
  - Old, valuable systems must be maintained and updated
- Heterogeneity
  - Systems are distributed and include a mix of hardware and software
- Delivery
  - There is increasing pressure for faster delivery of software

Today's Seminar – Engineering

What are our expectations of a "traditional" engineering project?