Introduction to Software Engineering

Week 1: Overview

Objectives

☐ understand what software engineering is and why it is important;

☐ understand that the development of different types of software system may require different software engineering techniques;

☐ understand ethical and professional issues that are important for software engineers;

Topics covered

1. Software
2. Software engineering
3. Software process
4. Software engineering ethics

Topics covered

1. Software
2. Software engineering
3. Software process
4. Software engineering ethics

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.

Role of software [1]

❑ Is software important? Why?

❑ Give ten examples of software

Role of software [2]

☐ 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 products

- ☐ Generic products
  - ☐ Stand-alone systems that are marketed and sold to any customer who wishes to buy them.
  - ☐ The specification of what the software should do is owned by the software developer and decisions on software change are made by the developer.
- ☐ Customized products
  - ☐ Software that is commissioned by a specific customer to meet their own needs.
  - ☐ The specification of what the software should do is owned by the customer for the software and they make decisions on software changes that are required.

Essential attributes of good software

☐ Maintainability
  ☐ Software should evolve to meet the changing needs of customers.

☐ Dependability and security
  ☐ Software dependability includes a range of characteristics including reliability, security and safety.

☐ Efficiency
  ☐ Software should not make wasteful use of system resources.

☐ Acceptability
  ☐ Software must be acceptable to the type of users for which it is designed.

Topics covered

1. Software
2. Software engineering
3. Software process
4. Software engineering ethics

Building real things ...

How about producing a software?

FAQs about Software Engineering

1. What is software engineering?
2. What are the fundamental software engineering activities?
3. What is the difference between software engineering and computer science?
4. What is the difference between software engineering and system engineering?
5. What are the key challenges facing software engineering?
6. What are the costs of software engineering?
7. What are the best software engineering techniques and methods?
8. What is software process?

History of software engineering

☐ Software crisis

☐ The term 'software engineering' was suggested at conferences organized by NATO in 1968 to discuss the 'software crisis'.

☐ The adoption of an engineering approach to software development would

  ☐ reduce the costs of software development

  ☐ lead to more reliable software.

What is software engineering?

❑ Software engineering is an <u>engineering discipline</u> that is concerned with <u>all aspects of software production</u> from the early stages of system specification through to maintaining the system after it has gone into use.

❑ Engineering discipline

 ❑ Using appropriate theories and methods to solve problems bearing in mind organizational and financial constraints.

❑ All aspects of software production

 ❑ Not just technical process of development. Also project management and the development of tools, methods etc. to support software production.

Layered Technology

tools

methods

process model

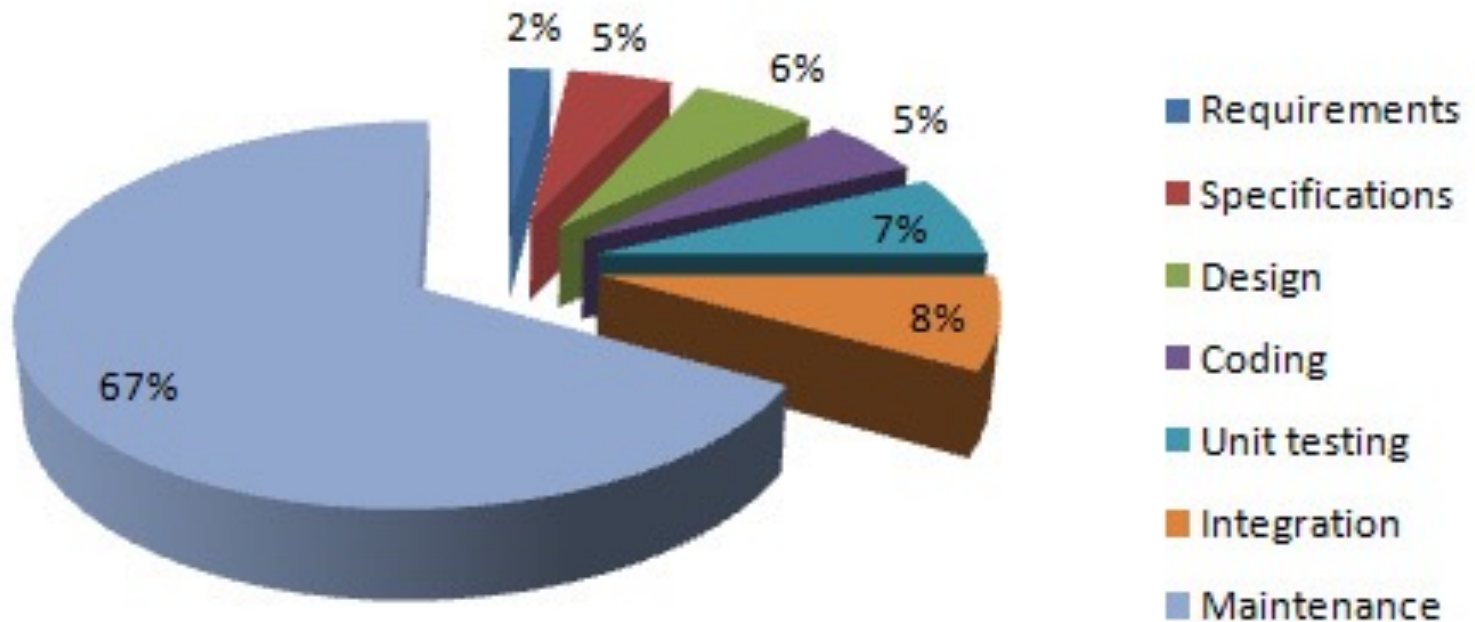**a "quality" focus**
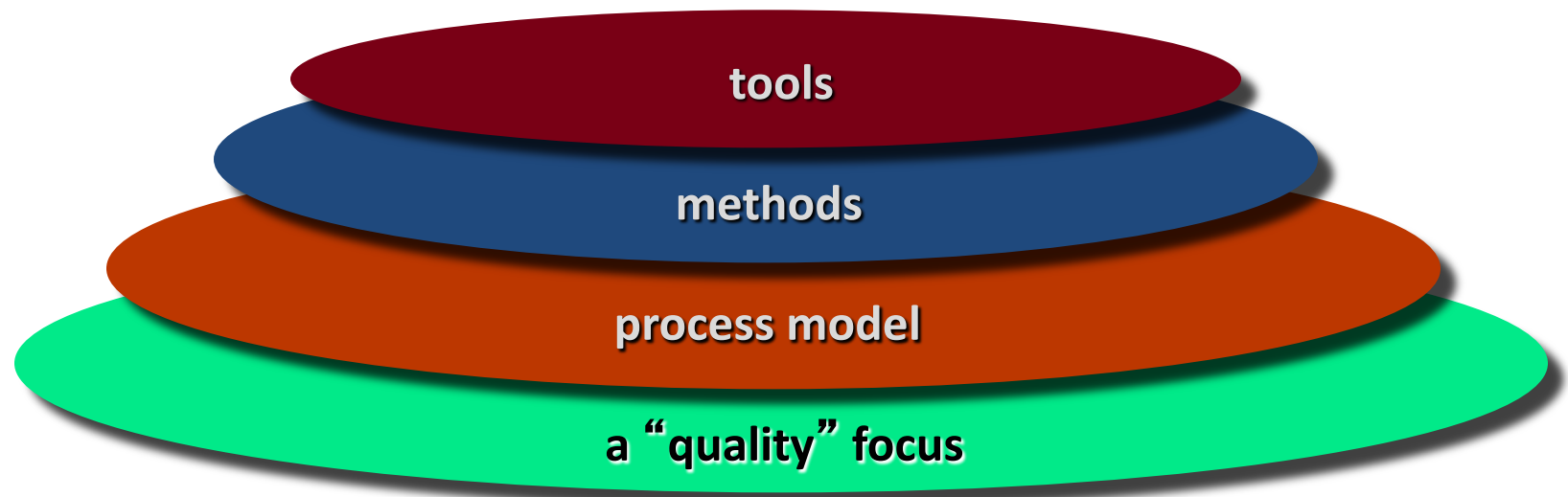
Software Engineering

Software costs

☐ Software costs often dominate computer 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.

Costs of software engineering

☐ Roughly 60% of software costs are development costs, 40% are testing costs.

☐ For custom software, evolution costs often exceed development costs.

Software costs

Software Life-Cycle Costs

- 2% Requirements
- 5% Specifications
- 6% Design
- 5% Coding
- 7% Unit testing
- 8% Integration
- 67% Maintenance

Source : Digital Aggregates

Software engineering vs. Computer science?

☐ Computer science focuses on theory and fundamentals.

☐ Software engineering is concerned with the practicalities of developing and delivering useful software.

CS   SE
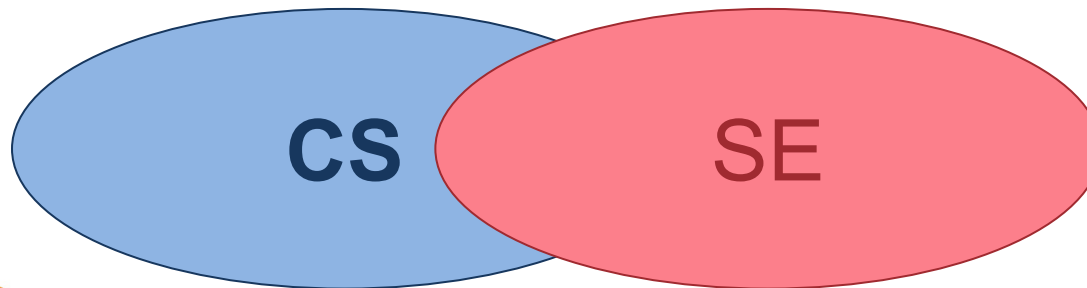
Software engineering vs. System engineering?

☐ System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering.

☐ Software engineering is part of this more general process.

System engineering **Software engineering**
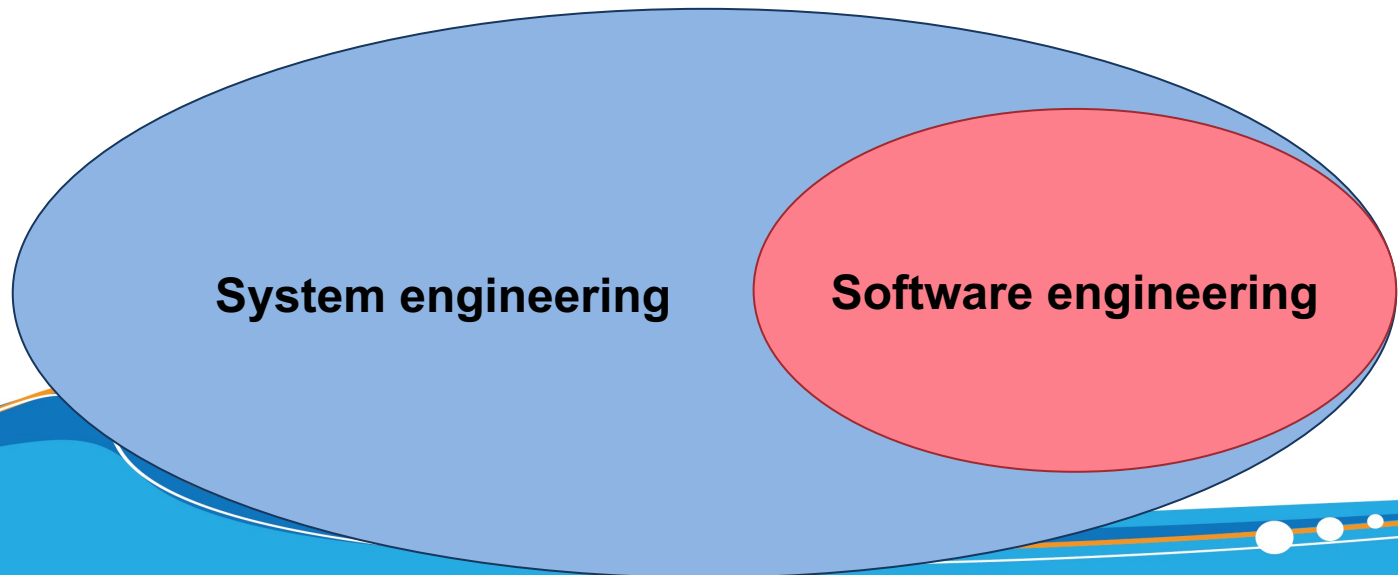
Best software engineering techniques and methods?

□ While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of system.

□ You can't say that one method is better than another.

Software failure: The Ariane launcher accident

Importance of software engineering

☐ More and more, individuals and society rely on advanced software systems. We need to be able to produce reliable and trustworthy systems economically and quickly.

☐ It is usually cheaper, in the long run, to use software engineering methods and techniques for software systems rather than just write the programs as if it was a personal programming project.

  ☐ For most types of system, the majority of costs are the costs of changing the software after it has gone into use.

Topics covered

1. Software
2. Software engineering
3. Software process
4. Software engineering ethics

What is software process?

☐ A sequence of activities that leads to the production of a software product.

☐ There are four fundamental activities that are common to all software processes.

Software process activities

☐ Software specification

    ☐ customers and engineers define the software that is to be produced and the constraints on its operation.

☐ Software development

    ☐ software is designed and programmed.

☐ Software validation

    ☐ software is checked to ensure that it is what the customer requires.

☐ Software evolution

    ☐ software is modified to reflect changing customer and market requirements.

General issues that affect software [1]

- ☐ Heterogeneity
  - ☐ Increasingly, systems are required to operate as distributed systems across networks that include different types of computer and mobile devices.

- ☐ Business and social change
  - ☐ Business and society are changing incredibly quickly as emerging economies develop and new technologies become available. They need to be able to change their existing software and to rapidly develop new software.

General issues that affect software [2]

☐ Security and trust

   ☐ As software is intertwined with all aspects of our lives, it is essential that we can trust that software.

☐ Scale

   ☐ Software has to be developed across a very wide range of scales, from very small embedded systems in portable or wearable devices through to Internet-scale, cloud-based systems that serve a global community.

Software engineering diversity

☐ There are many different types of software system and there is no universal set of software techniques that is applicable to all of these.

☐ The software engineering methods and tools used depend on

  ☐ the type of application being developed,

  ☐ the requirements of the customer and

  ☐ the background of the development team.

Application types

- ☐ Stand-alone applications
  - ☐ These are application systems that run on a local computer, such as a PC. They include all necessary functionality and do not need to be connected to a network.

- ☐ Interactive transaction-based applications
  - ☐ Applications that execute on a remote computer and are accessed by users from their own PCs or terminals. These include web applications such as e-commerce applications.

- ☐ Embedded control systems
  - ☐ These are software control systems that control and manage hardware devices. Numerically, there are probably more embedded systems than any other type of system.

Application types

☐ Batch processing systems

- ☐ These are business systems that are designed to process data in large batches. They process large numbers of individual inputs to create corresponding outputs.

☐ Entertainment systems

- ☐ These are systems that are primarily for personal use and which are intended to entertain the user.

☐ Systems for modeling and simulation

- ☐ These are systems that are developed by scientists and engineers to model physical processes or situations, which include many, separate, interacting objects.

Application types

- Data collection systems
  - These are systems that collect data from their environment using a set of sensors and send that data to other systems for processing.

- Systems of systems
  - These are systems that are composed of a number of other software systems.

Software engineering fundamentals

☐ Some fundamental principles apply to all types of software system, irrespective of the development techniques used:

  ☐ Systems should be developed using a managed and understood development process. Of course, different processes are used for different types of software.

  ☐ Dependability and performance are important for all types of system.

  ☐ Understanding and managing the software specification and requirements are important.

  ☐ Where appropriate, you should reuse software that has already been developed rather than write new software.

Internet software engineering

☐ The Web is now a platform for running application and organizations are increasingly developing web-based systems rather than local systems.

☐ Web services allow application functionality to be accessed over the web.

☐ Cloud computing is an approach to the provision of computer services where applications run remotely on the 'cloud'.

☐ Users do not buy software buy pay according to use.

Web-based software engineering

☐ Web-based systems are complex distributed systems but the fundamental principles of software engineering discussed previously are as applicable to them as they are to any other types of system.

☐ The fundamental ideas of software engineering apply to web-based software in the same way that they apply to other types of software system.

Web software engineering

☐ Software reuse

☐ Incremental and agile development

☐ Service-oriented systems

☐ Rich interfaces

Topics covered

1. Software
2. Software engineering
3. Software process
4. Software engineering ethics

Software engineering ethics

☐ Software engineering involves wider responsibilities than simply the application of technical skills.

☐ Software engineers must behave in an honest and ethically responsible way if they are to be respected as professionals.

☐ Ethical behaviour is more than simply upholding the law but involves following a set of principles that are morally correct.

Issues of professional responsibility

☐ Confidentiality
  ☐ Engineers should normally respect the confidentiality of their employers or clients irrespective of whether or not a formal confidentiality agreement has been signed.

☐ Competence
  ☐ Engineers should not misrepresent their level of competence. They should not knowingly accept work which is outwith their competence.

Issues of professional responsibility

☐ Intellectual property rights

☐ Engineers should be aware of local laws governing the use of intellectual property such as patents, copyright, etc. They should be careful to ensure that the intellectual property of employers and clients is protected.

☐ Computer misuse

☐ Software engineers should not use their technical skills to misuse other people's computers. Computer misuse ranges from relatively trivial (game playing on an employer's machine, say) to extremely serious (dissemination of viruses).

ACM/IEEE Code of Ethics

- The professional societies in the US have cooperated to produce a code of ethical practice.

- Members of these organisations sign up to the code of practice when they join.

- The Code contains eight Principles related to the behaviour of and decisions made by professional software engineers, including practitioners, educators, managers, supervisors and policy makers, as well as trainees and students of the profession.
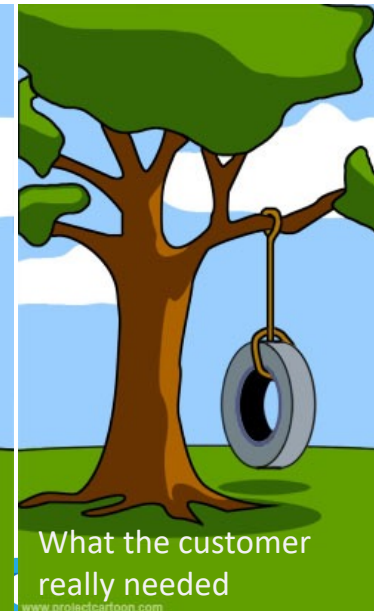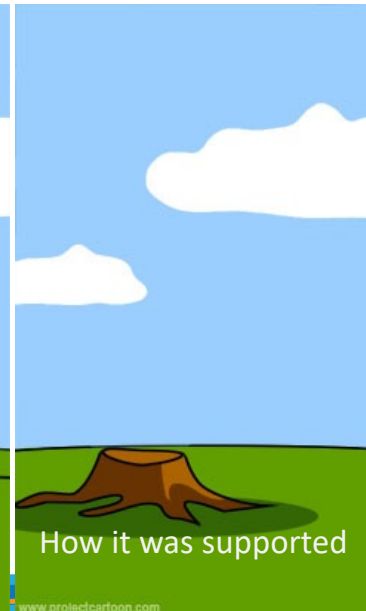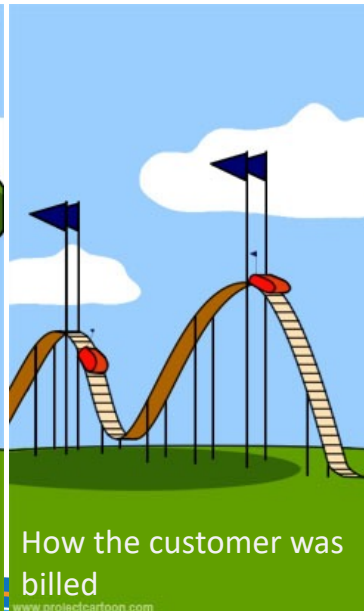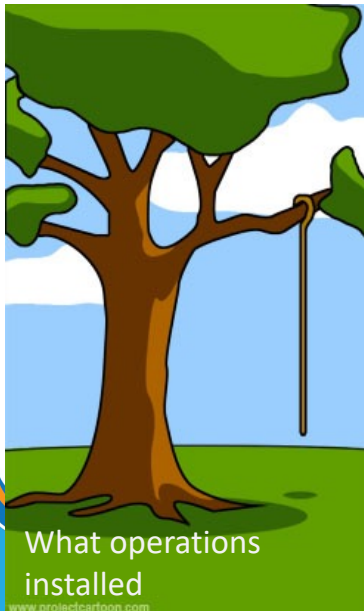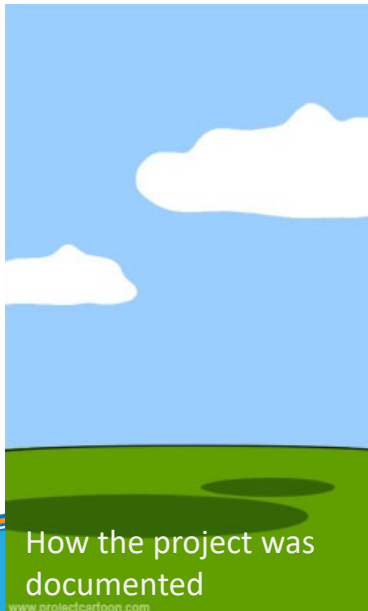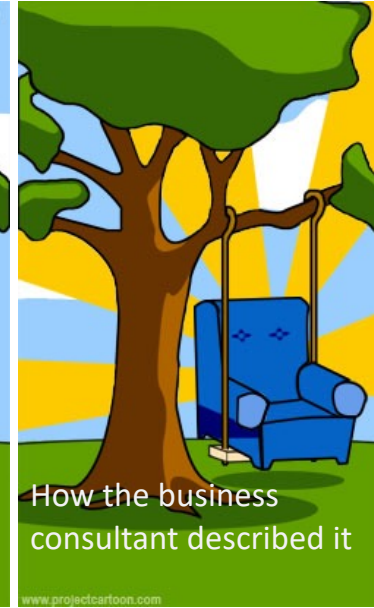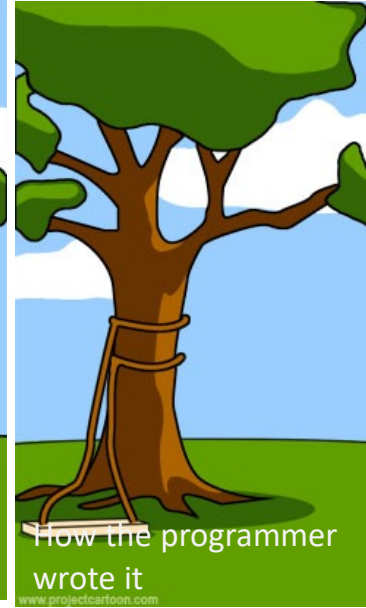
The ACM/IEEE Code of Ethics: PREAMBLE

☐ The short version of the code summarizes aspirations at a high level of the abstraction; the clauses that are included in the full version give examples and details of how these aspirations change the way we act as software engineering professionals. Without the aspirations, the details can become legalistic and tedious; without the details, the aspirations can become high sounding but empty; together, the aspirations and the details form a cohesive code.

☐ Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession. In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to the following Eight Principles:

Ethical principles

1. PUBLIC - Software engineers shall act consistently with the public interest.

2. CLIENT AND EMPLOYER - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.

3. PRODUCT - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.

4. JUDGMENT - Software engineers shall maintain integrity and independence in their professional judgment.

5. MANAGEMENT - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.

6. PROFESSION - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.

7. COLLEAGUES - Software engineers shall be fair to and supportive of their colleagues.

8. SELF - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

How projects really work ?

How the customer explained it

How the project leader understood it

How the analyst designed it

How the programmer wrote it

How the business consultant described it

How the project was documented

What operations installed

How the customer was billed

How it was supported

What the customer really needed

Questions?