WIA2002: Software Modelling
Semester 1, Session 2016/17

Lecture 1: Introduction of Software Engineering

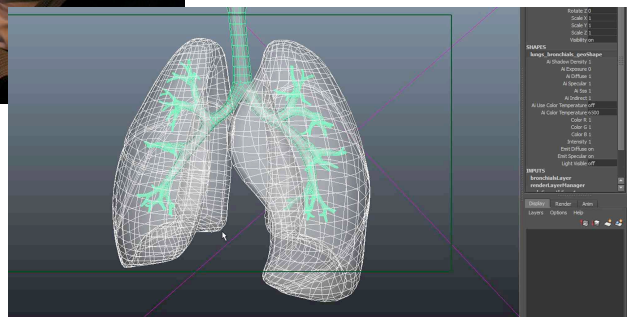
Learning objectives

- Understand what software engineering is and why it is important.
- Understand that the development of different types of software systems may require different software engineering techniques.
- Understand some ethical and professional issues that are important for software engineers.

The world we live in..

The economies of **ALL** developed nations are dependent on software.

More and more systems are **software controlled**.



The world we live in..

- *Software engineering* is concerned with
  - **theories**
  - **methods**
  - **tools** for **professional software development**.

Software Engineering - When it all started?

*The notion of 'software engineering' was **first proposed in 1968** at a conference held to discuss what was then called the 'software crisis' (Naur and Randell, 1969).*

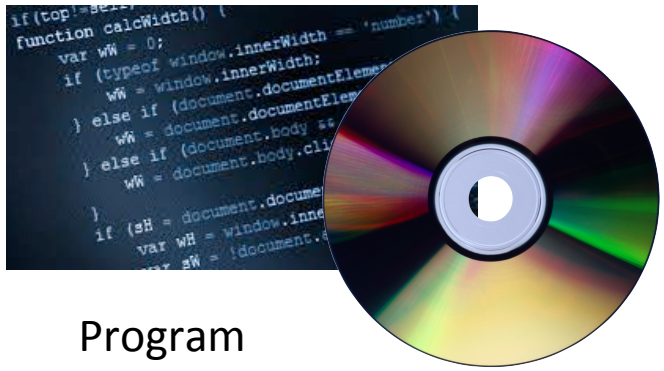
Why?

*It became clear that individual approaches to program development **did not scale up to large and complex software systems**. These were **unreliable, cost more than expected, and were delivered late**.*

*Throughout the 1970s and 1980s, a variety of new software engineering techniques and methods were developed, such as **structured programming, information hiding and object-oriented development**. Tools and standard notations were developed and are now extensively used.*

What is software?

- Computer programs and associated documentation.



Program

What is software?

- Software products may be developed
  - for a particular customer
  - for a general market.

What are the attributes of good software?

- Deliver the required **functionality** and **performance** to the user
- **Maintainable**
- **Dependable**
- **Usable**.

Essential attributes of good software

Product characteristic	Description
Maintainability	Software should be written in such a way so that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment.
Dependability and security	Software dependability includes a range of characteristics including reliability, security and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system.
Efficiency	Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilisation, etc.
Acceptability	Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable and compatible with other systems that they use.

What is software engineering (SE)?

- An engineering discipline that is concerned with **all aspects of software production**.
  - From beginning until system has been deployed.

- SE is an engineering discipline
  - using appropriate theories and methods
  - considering organizational and financial constraint

What is software engineering (SE)?

- SE concerned with **all** aspects of software production (anything that support software production):
  - technical
  - project management
  - tools development
  - methods development

Why we need SE?

- To **produce reliable** and **trustworthy** systems **economically** and **quickly**.

- Cheaper in the long run.
  - most of the cost goes to changing the software after deployment.

What are the 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 engineering vs. computer science

- **Computer science** focuses on theory and fundamentals.

- S**oftware engineering** is concerned with the practicalities of developing and delivering useful software.

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.

What are the fundamental software engineering activities?

- Software specification
  - customer and engineer define software features
  - operation constraint
- Software development (design and implementation)
  - software is designed and programmed
- Software validation
  - software is checked to ensure correctness
- Software evolution
  - software is modified to requirement change

What are the key challenges facing software engineering?

- Coping with increasing diversity
  - systems are required to operate as distributed systems across networks that include different types of computer and mobile devices.
    - no universal set of software techniques that is applicable to all of these.
    - methods and tools used depend on the type of application being developed, the requirements of the customer and the background of the development team.
- Demands for reduced delivery times
  - business and society are changing fast, so existing software has to be able to be updated quickly.
- Developing trustworthy software
  - dealing with lives.

Application types

- Stand-alone applications
  - application systems that run on a local computer, such as a PC.
    - includes 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.
    - includes web applications such as e-commerce applications.
- Embedded control systems
  - software control systems that control and manage hardware devices.
  - there are probably more embedded systems than any other type of system.

Application types

- Batch processing systems
  - business systems that are designed to process data in large batches.
  - process large numbers of individual inputs to create corresponding outputs.
- Entertainment systems
  - systems that are primarily for personal use and which are intended to entertain the user.
- Systems for modelling and simulation
  - 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
  - systems that collect data from their environment using a set of sensors and send that data to other systems for processing.
- Systems of systems
  - systems that are composed of a number of other software systems.

What are the best software engineering techniques and methods?

No one methods can be best applied to all system development!

BUT..

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

Software Engineering fundamentals

- Systems should be **developed using a managed** and **understood development process**.
- **Dependability** and **performance** are *important for all types of system*.
- **Understanding** and **managing** the **software specification and requirements** (what the software should do) are important.
- Where appropriate, **reuse software** that has already been developed rather than write new software.

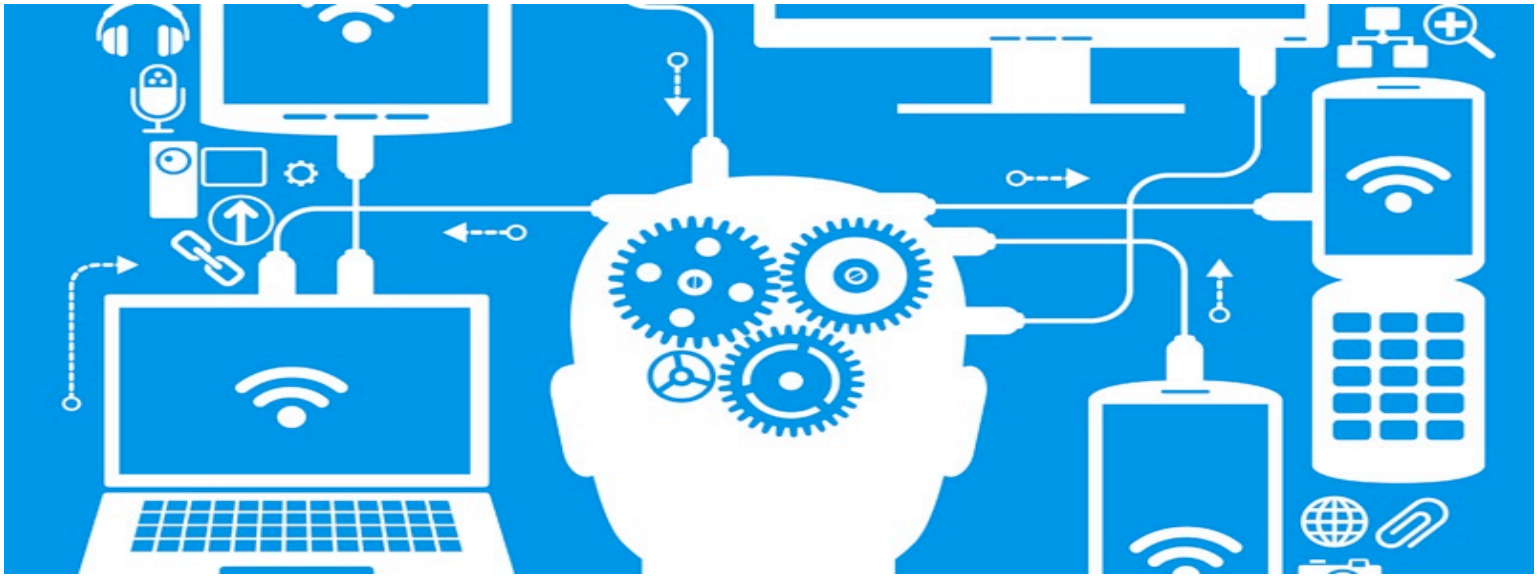
What differences has the web made to software engineering?

- **Software reuse** is the dominant approach for constructing web-based systems.

- The web has led to the availability of software services and the possibility of developing highly distributed service-based systems.
  - but the fundamental principles of software engineering discussed previously are as applicable to them as they are to any other types of system.

- **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.

Any question so far..?



What is ethics?

Ethics (also moral philosophy) is the branch of philosophy that involves systematizing, defending, and recommending concepts of right and wrong conduct.

- Internet Encyclopedia of Philosophy on "Ethics"

Ethics in Software Engineering

- 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).

The 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

Software Engineering Code of Ethics and Professional Practice

ACM/IEEE-CS Joint Task Force on Software Engineering Ethics and Professional Practices

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:

The ACM/IEEE Code of Ethics - Ethical principle

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.

Ethical dilemmas

- Disagreement in principle with the policies of senior management.
- Your employer acts in an unethical way and releases a safety-critical system without finishing the testing of the system.
- Participation in the development of military weapons systems or nuclear systems.

Lets reflect..

- **Software engineering** is an engineering discipline that is concerned with all aspects of software production.
- **Essential software product attributes** are maintainability, dependability and security, efficiency and acceptability.
- The **high-level activities** of specification, development, validation and evolution are part of all software processes.
- There are many **different types of system** and each requires appropriate software engineering tools and techniques for their development.

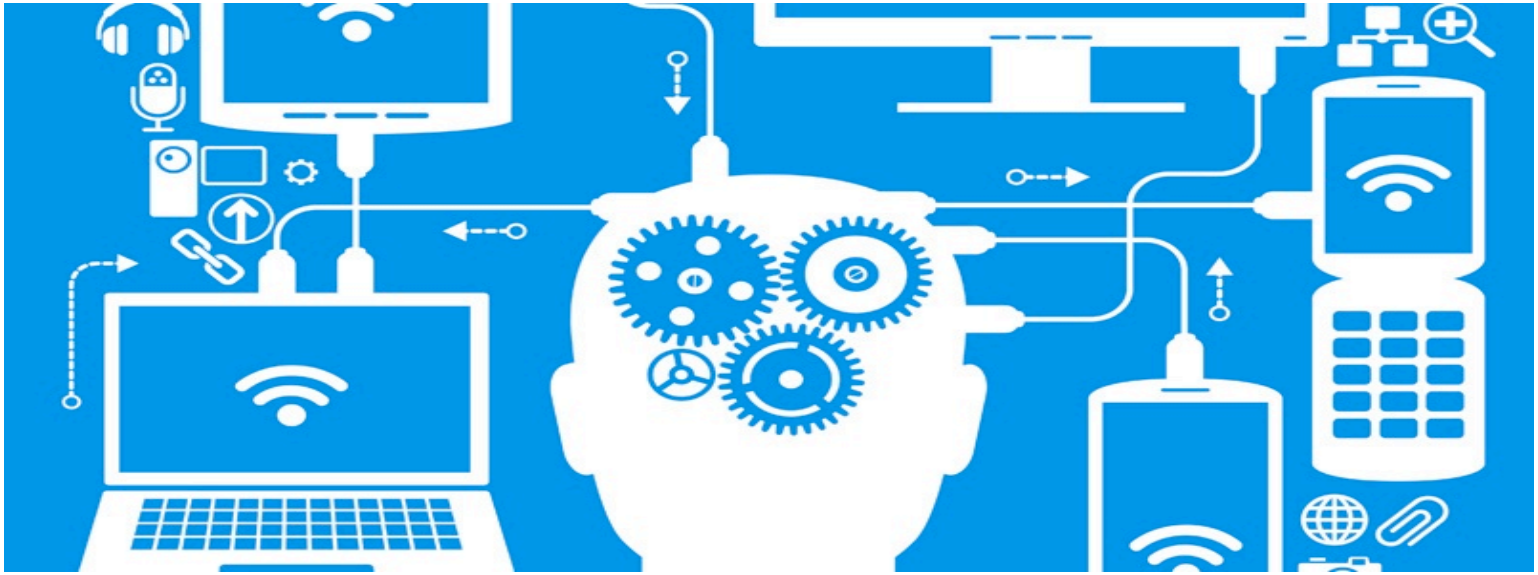
Lets reflect..

- The **fundamental ideas** of software engineering are applicable to all types of software system.

- **Software engineers have responsibilities** to the engineering profession and society. They should not simply be concerned with technical issues.

- Professional societies publish **codes of conduct** which set out the standards of behaviour expected of their members.

In the next lecture..



Lecture 2: Overview of Software Development Lifecycle models