Chapter 1. Objectives

- To introduce software engineering and to explain its importance
- To set out the answers to 11 key questions and answers about software engineering
- To introduce ethical and professional issues and to explain why they are of concern to software engineers

Topics covered

- 11 FAQs about software engineering
- Professional and ethical responsibility

Why needs Software engineering?

- **Producing SW in a cost-effective way is important for all developed national and international economies**
- **More and more systems are software controlled**
- **New techniques and methods were used in SW development (SW Crisis)**
- **Software engineering is concerned with theories, methods and tools for professional software development, deliver on time, under budget**
- **Software engineering cost represents a significant fraction of GNP in all developed countries**

11 FAQs about software engineering

- What is software?
  - **Computer programs and associated documentation**
- What is software engineering?
  - **An engineering discipline concerned with all aspects of SW production**
- What is the difference between software engineering and computer science?
  - **Computer science is concerned with theory and fundamentals; SE is concerned with the practicalities of developing and delivering useful SW**
- What is the difference between software engineering and system engineering?
  - **System engineering is concerned with all aspects of computer-based system development, including HW, SW and Process engineering**
- What is a software process?
  - **A set of activities whose goal is the development or evolution of SW**
- What is a software process model?
  - **A simplified representation of a SW process, presented from a specific perspective**

11 FAQs about software engineering

- What are the costs of software engineering?
  - **60% is development cost, 40% is testing cost; evolution cost often exceed development cost**
- What are software engineering methods?
  - **Structured approaches to SW development which include system models, notations, rules, design advise and process guidance Ex. SDG 2.0, UML**
- What is CASE (Computer-Aided Software Engineering)
  - **SW system are intended to provide automated support for SW process activities**
- What are the attributes of good software?
  - **SW should deliver the required functionality and performance to the user, and should be maintainable, dependable and usable**
- What are the key challenges facing software engineering?
  - **Coping with legacy system, increasing diversity and demands for reduced delivery time**

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
    specs are kept by the developer
  - Bespoke (customised) - developed for a single customer according to their specification, specs are kept by the buyer

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
- Some knowledge of CS is essential for SE
- 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 products
- 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 → evolve to meet changing customer's needs

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 of the development costs
- Software engineering is concerned with cost-effective software development

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
- Evolution SW cost often exceed the development cost(Fig. 1.4)
- Distribution of costs depends on the development model that is used – such as waterfall, evolutionary development (Fig. 1.2 v.s. Fig. 1.3)

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 in the processes
  - Data-flow perspective - information flow
  - Role/action perspective - who does what
- General process models of SW development
  - Waterfall – specification, design, development, integration & testing
  - Evolutionary development – these processes are carried out in parallel
    An initial system is rapidly developed from very abstract specs(Fig. 1.3)
  - Formal transformation – producing a formal mathematical system spec and transform this spec to program by mathematical methods
  - Integration from reusable components(SW IC chap. 14)

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 – object model(UML), data-flow model(DFD), state machine model(DFA)
- Rules
  - **Constraints** applied to system models – unique name of an entity in a E-R model
- Recommendations
  - **Advice on good design practice** – No object should have 7 sub-objects associated with it
- Process guidance
  - What activities to follow – object attribute should be **documented** before defining the operations associated with the object

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 many tool support
- Upper-CASE(Front-end CASE)
  - Tools to support the early process activities of **requirements analysis** and **design**
- Lower-CASE(Back-end CASE)
  - Tools to support later activities such as **program editor, programming**, **debuggin**g and **testing**

What are the attributes of good software?

- Functional attributes v.s. Non-functional attributes
- 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(reliable,security, and safety)
- Efficiency
  - Software should not make wasteful use of system resources(memory or CPU cycle)
- Usability
  - Software must be usable by the users for which it was designed

What are the key challenges facing software engineering?

- Coping with **legacy systems**, coping with **increasing diversity** and coping with **demands for reduced delivery times**
- Legacy systems
  - Old, valuable systems must be maintained and updated
- Heterogeneity -- diversity
  - Systems are distributed and include a mix of hardware and software(distributed system across network)
- Delivery – delivery time
  - There is increasing pressure for faster delivery of software(Traditional SE techniques are time-consuming)

Professional and ethical responsibility

- 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**
- Software engineers are bounded by **local**, **national** and **international law**.

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 accept work which is out with 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(ACM/IEEE) in the US have cooperated to produce a code of ethical practice.
- **Members of ACM/IEEE** or **British Computer Society** sign up to the code of conduct when they join.
- The Code contains **eight Principles** related to the behaviour of decisions made by professional software engineers, including practitioners, educators, managers, supervisors and policy makers, as well as trainees and students of the profession.

Code of ethics – ACM/IEEE's preamble

- **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**:

Code of ethics – 8 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.

Code of ethics - principles

- 4. JUDGMENT
  - Software engineers shall maintain integrity and independence in their professional judgment.
- 5. MANAGEMENT
  - Software engineering managers and leaders shall agree 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.

Code of ethics - principles

- 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 in the company?
- Your employer acts in an unethical way and releases a safety-critical system without finishing the testing of the system?➔back-door system
- Participation in the development of military weapons systems or nuclear systems

Key points

- **Software engineering** is an engineering discipline which is concerned with all aspects of **software production**.
- Software products consist of **developed programs** and **associated documentation**. Essential product attributes are maintainability, dependability, efficiency and usability.
- The software process consists of activities which are involved in developing software products. **Basic activities** are software **specification, development, validation and evolution**.
- Methods are organised ways of producing software. They include suggestions for the process to be followed, the notations to be used, rules governing the system descriptions which are produced and design guidelines.

Key points

- CASE tools are software systems which are designed to support routine activities in the software process such as editing design diagrams, checking diagram consistency and keeping track of program tests which have been run.
- 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.

HomeWork#1

- 1.2
- 1.3
- 1.5