Professional Software Development

Instructor: Cong Pu

Email: cong.pu@ttu.edu

Topics

✧ Professional software development

  ▪ What is meant by software engineering.

✧ Software engineering ethics

  ▪ A brief introduction to ethical issues that affect software engineering.

Introduction

✧ We cant run the modern world without software:

- National infrastructures and utilities
- Industrial manufacturing and distribution
- Entertainment

✧ Software system are abstract and intangible:

- Properties of materials
- Governed by physical laws
- Manufacturing process
- No natural limits to the potential of software

✧ Software system can quickly become extremely complex, difficult to understand, and expensive to change.

Introduction

✧ Different types of systems:

 ▪ Simple embedded system

 ▪ Complex worldwide information systems

✧ No universal notations, methods, or techniques for software engineering for different types of systems.

✧ "Software failure":

 ▪ Increasing demands

 ▪ Low expectations

Professional Software Development

Professional Software Development

♢ Lots of people write program:

- For particular job
- For fun

♢ Software development is a professional activity:

- Software is developed for specific business purpose

♢ Professional software:

- Intended for use by someone rather than its developer
- Maintained and changed throughout its life

Professional Software Development

 ✧ Software engineering is intended to support professional software development, rather than individual programming:

 ▪ Including techniques that support program specification, design, and evolution.

 ▪ Not relevant for personal software development.

Frequently asked questions about software engineering

Question	Answer
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.
What are the attributes of good software?	Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable.
What is software engineering?	Software engineering is an engineering discipline that is concerned with all aspects of software production.
What are the fundamental software engineering activities?	Software specification, software development, software validation and software evolution.
What is the difference between software engineering and computer science?	Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.

Frequently asked questions about software engineering

Question	Answer
What are the key challenges facing software engineering?	Coping with increasing diversity, demands for reduced delivery times and developing trustworthy software.
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.
What are the 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. For example, games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analyzable specification to be developed. You can't, therefore, say that one method is better than another.

Software Products

✧ Generic products

- Stand-alone systems that are marketed and sold to any customer who wishes to buy them.

- Examples – PC software such as graphics programs, project management tools; CAD software; software for specific markets such as appointments systems for dentists.

✧ Customized products

- Software that is commissioned by a specific customer to meet their own needs.

- Examples – embedded control systems, air traffic control software, traffic monitoring systems.

Product Specification

✧ Generic products

- 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

- 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

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.

Software Engineering

✧ Software engineering is an engineering discipline that is concerned with all aspects of software production 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.

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.

Software Process Activities

✧ Software specification, where customers and engineers define the software that is to be produced and the constraints on its operation.

✧ Software development, where the software is designed and programmed.

✧ Software validation, where the software is checked to ensure that it is what the customer requires.

✧ Software evolution, where the software is modified to reflect changing customer and market requirements.

General Issues Affecting Software

✧ 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 Affecting Software

✧ 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.

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 (what the software should do) are important.

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

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.

✧ You should not use your skills and abilities to behave in a dishonest way or in a way that will bring disrepute to the software engineering profession.

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 outside 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.

Rationale for Code of Ethics

- *Computers have a central and growing role in commerce, industry, government, medicine, education, entertainment and society at large. Software engineers are those who contribute by direct participation or by teaching, to the analysis, specification, design, development, certification, maintenance and testing of software systems.*

- *Because of their roles in developing software systems, software engineers have significant opportunities to do good or cause harm, to enable others to do good or cause harm, or to influence others to do good or cause harm. To ensure, as much as possible, that their efforts will be used for good, software engineers must commit themselves to making software engineering a beneficial and respected 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:

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.

Code of Ethics

- *https://www.youtube.com/watch?v=d91MTV9AlNg*

Key Points

✧ 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.

✧ The fundamental notions of software engineering are universally applicable to all types of system development.

Key Points

◇ There are many different types of system and each requires appropriate software engineering tools and techniques for their development.

◇ 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.