

Introduction to Software Engineering

1

Fattane Zarrinkalam

Personal software development

- Writing a program for **yourself**, no one else will use it
- Example:
 - People in business write spreadsheet programs to simplify their jobs;
 - Scientists and engineers write programs to process their experimental data;
 - Hobbyists write programs for their own interest and enjoyment.
- You don't have to worry about writing program guides, documenting the program design, and so on.

Professional software development

- A professional activity in which software is developed for **business purposes**
 - for inclusion in other devices, or as software products such as information systems and computer-aided design systems.
- Professional software
 - It is intended for use by the people other than its developer.
 - It is usually developed by teams rather than individuals.
 - It is maintained and changed throughout its life.

Software engineering

- Software engineering is concerned with theories, methods and tools for **professional** software development rather than personal software development.
- It includes techniques that support program specification, design, and evolution, none of which are normally relevant for personal software development.



Professional software development

1

What is software?

What is software?

- Computer programs and
 - all associated documentation,
 - libraries,
 - support websites,
 - and configuration data that are needed to make these programs useful.
- Two kinds of software products:
 - Generic products
 - Customized products

Two kinds of software products

- Generic products
 - Stand-alone systems that are produced by a development organization and sold on the **open market** to **any customer** who is able to buy them.
 - Examples – PC software such as project management tools; software for specific markets such as appointments systems for dentists.
 - The specification of what the software should do is owned by the software developer and decisions on software change are made by the developer.

Two kinds of software products

- Customized products
 - Software that is developed for a **specific customer** to meet their own needs.
 - Examples – air traffic control software, traffic monitoring systems.
 - 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.

2

What are the attributes of good software?

Attributes of good software

- Good software should deliver the required **functionality** and performance to the user.
- The quality of professional software also includes the software's behavior while it is executing and the structure and organization of the system programs and associated documentation.
 - This is reflected in the software's **non-functional** attributes.
- Examples
 - an aircraft control system must be **safe**,
 - an interactive game must be **responsive**,
 - a telephone switching system must be **reliable**.

Attributes of good software

Product characteristic	Description
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.
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.
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.

3

What are software engineering?

Software engineering

- Software engineering is an **engineering discipline** that is concerned with **all aspects of software production**.
- Engineering discipline
 - Engineers make things work.
 - They apply appropriate theories, methods, and tools to solve problems within organizational and financial constraints.
- All aspects of software production
 - Not just technical process of development. from the early stages of system specification through to maintaining the system after it has gone into use

4

What are the fundamental software engineering activities?

Fundamental software engineering 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.

5

What is the difference between software engineering and computer science?

Software engineering VS. Computer science

- Computer science:
 - The theories and methods that underlie computers and software systems.
- Software engineering:
 - The practical problems of producing software

Some knowledge of computer science is essential for software engineers

6

What is the difference between software engineering and system engineering?

Software engineering VS. System engineering

- System engineering:
 - All aspects of computer-based systems development including hardware, software and process engineering.
- Software engineering:
 - Part of system engineering as a more general process.

7

What are the key challenges facing software engineering?

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.
 - Create most appropriate software for each device and computer
- Demands for reduced delivery times
 - Many traditional software engineering techniques need to evolve so that the time required for software to deliver value to its customers is reduced.
- Developing trustworthy software
 - As software is intertwined with all aspects of our lives, it is essential that we can trust that software.

8

What are the costs of software engineering?

The cost of software engineering

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

9

What are the best software engineering techniques and methods?

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.
- For example, games should always be developed using a series of prototypes whereas safety-critical control systems require a complete and analysable specification to be developed.
- 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.

Software engineering fundamentals

- 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.

10

Why software engineering?

The importance of software engineering

- Our infrastructure, power, telecommunication, transport, the financial systems are all dependent to software.
- More and more, individuals and society rely on advanced software systems to cope with different challenges for humanity.
 - climate change, fewer natural resources, and expanding world population.
- Software engineering is essential for the functioning of government, society, and national and international businesses and institutions.
- We need software engineering to be able to produce **reliable** and **trustworthy** systems **economically** and **quickly**.

A grayscale photograph of a large crowd of people, likely at a sporting event, with the word 'Summary' overlaid in the center.

Summary

Summary

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