

CSE 333 – Software Engineering

Pranta Sarker
Lecturer
Dept. of CSE, NEUB

Lecture 01

Reference book

SOFTWARE ENGINEERING

Ian Sommerville

Ninth Edition

Software Engineering: Theme

- Software
- Science and Engineering
- Why Software with Engineering not Science?
- What does it mean?
- How software engineering contribute on the software market industry?
- How software engineering teach a student to make a professional software?

Software

- Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.

Types of Software Product

- Generic product
 - Stand-alone systems
 - Produced by a development organization
 - Sold on the open market
 - Ex. PC software such as graphics programs, project management tools, CAD software.
- Customized product
 - Commissioned by a particular customer
 - Software developer/contractor develops the software only for that customer
 - Ex. Embedded control system, air traffic control system, traffic monitoring system.

Software Engineering
focuses to make a
customer/customized
product

Software Engineering

- Software engineering is concerned with theories, methods and tools for professional software development.
- The economies of ALL developed nations are dependent on software.
- More and more systems are software controlled.

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.

Engineers are all in all. They make decision, apply any theory to develop a software.

Not just development of a software. It includes software management, development tools, methods, theories and others

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.

Take a tour: Software Engineering

- Software process
- Requirement Engineering
- System modeling
- Architectural design
- Design and implementation
- Software testing
- Software evolution

Take a tour: Software Engineering

- Security
- Project management
- Project planning
- Quality management
- Process improvement

Essential attributes of good software

- Maintainability

- Software should be written in such a way so that it can **evolve to meet the changing** needs of customers.

- 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.

Essential attributes of good software

- 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 process activities

- Software specification
 - **customers and engineers** define the software that is to be produced and the constraints on its operation.
- Software development
 - the software is **designed and programmed**.
- Software validation
 - the software is **checked** to ensure that it is what the customer requires.
- Software evolution
 - the software is **modified to reflect changing** customer and market requirements.

General issues that affect most 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**.
- Security and trust
 - As software is intertwined with all aspects of our lives, it is essential that **we can trust that software**.

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 modelling 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 (what the software should do) are important.
 - Where appropriate, you should **reuse software** that has already been developed **rather than write new software**.