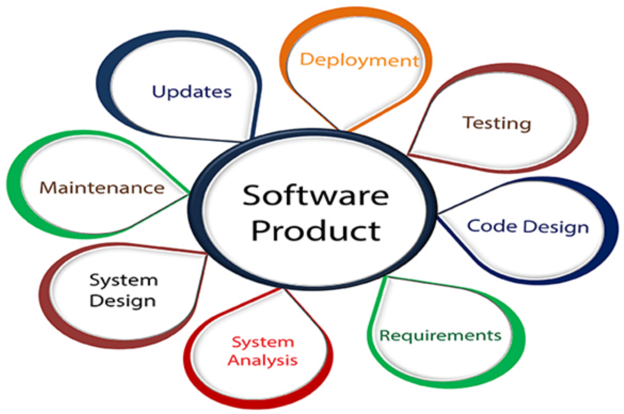
Introduction

Lecture 1

Book Reference: “SOFTWARE ENGINEERING”
Ian Sommerville

Last Class

- ✧ Software vs Program
- ✧ Software Engineering
  - **Software Engineering** is an engineering branch related to the evolution of software product using well-defined scientific principles, techniques, and procedures
- ✧ Software Product



- ✧ Definitions:
  - "Software engineering is the establishment and use of sound engineering principles in order to obtain **economically software** that is reliable and work efficiently on real machines."

Software engineering – Why?

- ✧ The economies of ALL developed nations are dependent on **software**.
- ✧ More and more **systems** are software controlled
- ✧ Software engineering is concerned with **theories**, **methods** and **tools** for professional software development.
- ✧ Expenditure on software represents a significant fraction of GNP in all developed countries.

Software costs

- Software costs often dominate computer 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 development costs.
- **Software engineering is concerned with cost-effective software development.**

Software products

- ✧ **Generic products / Ready Made Software**
  - 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.

Frequently asked questions about software engineering

✧ 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.

Frequently asked questions about software engineering

- ✧ 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

- ✧ 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 more general process.
- ✧ What are the key challenges facing software engineering?
  - Coping with increasing diversity, demands for reduced delivery times and developing trustworthy software.

Frequently asked questions about software engineering

- ✧ 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 analysable specification to be developed.
  - ▪ You can’t, therefore, say that one method is better than another.

Part -2

Software Engineering

✧ Engineering Approach to develop software

- principles, techniques, and procedures

  Or

- theories, methods and tools

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 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**.

Frequently asked questions about software engineering

- ✧ What differences has the web made to software engineering?

Software - Evolution

- The role of computer software has undergone significant change over a time span of little more than 75 years.
- Dramatic improvements in **hardware performance**, profound changes in **computing architectures**, **vast increases in memory** and **storage capacity**, and a wide variety of **exotic input and output** options have all precipitated more sophisticated and complex computer-based systems.
- The **lone programmer** of an earlier era has been replaced by a **team of software specialists**, each focusing on one part of the technology required to deliver a complex application.

Software Evolution

- ✧ EVOLUTION:
- ✧ 1945 - 65 : ERA of Origin - NASA, IBM: Build and Deliver approach
- ✧ 1965 – 85: CRISIS ERA – Failure of software- no proper approach followed (only 2% s/w successful) e.g.: Failure projects - OS/360, TORUS
- ✧ 1990 – 2000: Internet ERA: MS-Windows, browsers (Proprietary Software)
- ✧ 2000 – 2010: Light weight Software ERA
- ✧ 2010 – Till Date: AI, ML, DL: ERA of Freeware, Self Learning Software

Frequently asked questions about software engineering

- ✧ What differences has the web made to software engineering?
  - The web has led to the availability of software services and the possibility of developing highly distributed service-based systems.
  - Web-based systems development has led to important advances in programming languages and software reuse.

Software engineering and the web

- The **Web is now a platform** for running application and organizations are increasingly developing web-based systems rather than local systems.
- **Web services** allow application functionality to be accessed over the web.
- **Cloud computing** is an approach to the provision of computer services where applications run remotely on the ‘cloud’.
  - Users do not buy software but pay according to use.

Application Types

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.

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 Evolution

- The process of developing a software product using software engineering principles and methods is referred to as **software evolution.**
- This includes
  - the initial development of software
  - its maintenance and updates,

  till desired software product is developed, which satisfies the expected requirements.

THE EVOLVING ROLE OF SOFTWARE

THE EVOLVING ROLE OF SOFTWARE

✧Software takes on a dual role:

- a product and,
- at the same time, the vehicle for delivering a product.

Software as a Product

- ✧ As a product, it delivers the **computing potential** embodied by computer hardware or, more broadly, a network of computers that are accessible by local hardware.
- ✧ Whether it resides within a cellular phone or operates inside a mainframe computer,
  - Software is an **information transformer**—producing, managing, acquiring, modifying, displaying, or transmitting information that can be as simple as a single bit or as complex as a multimedia presentation.

Software as a Vehicle

- As the vehicle used to deliver the product,
  - software acts as the basis for the control of the computer (operating systems),
  - the communication of information (networks),
  - and the creation and control of other programs (software tools and environments).
- Software delivers the most important product of our time—
  - **Information**.

Software - Role

- ✧ Software transforms personal data (e.g., an individual’s financial transactions) so that the data can be more useful in a local context;
- ✧ it manages business information to enhance competitiveness;
- ✧ it provides a gateway to worldwide information networks (e.g., Internet) and provides the means for acquiring information in all of its forms.

Case studies

✧ A personal insulin pump

- An embedded system in an insulin pump used by diabetics to maintain blood glucose control.

✧ A mental health case patient management system

- A system used to maintain records of people receiving care for mental health problems.

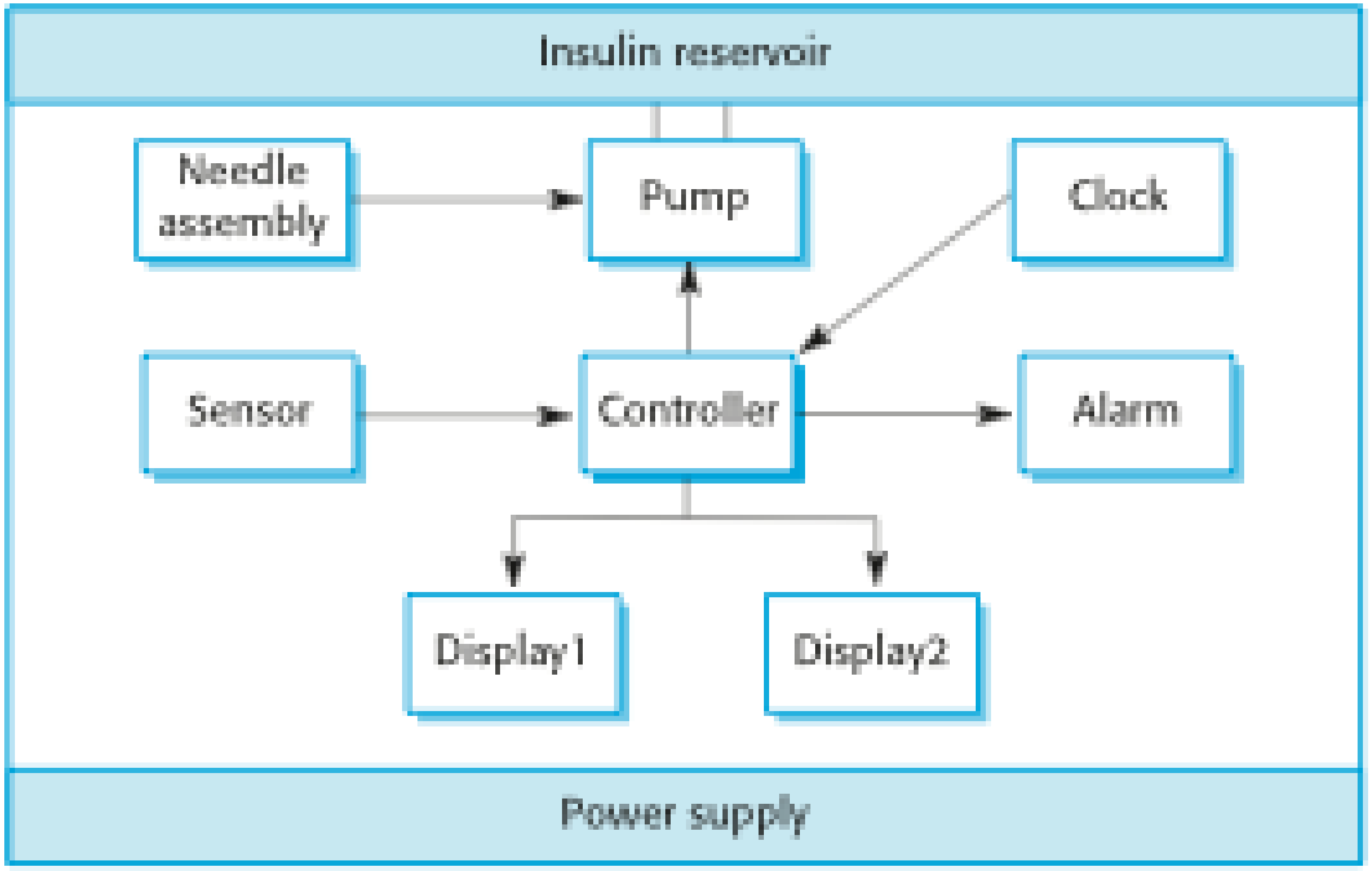
✧ A wilderness weather station

- A data collection system that collects data about weather conditions in remote areas.

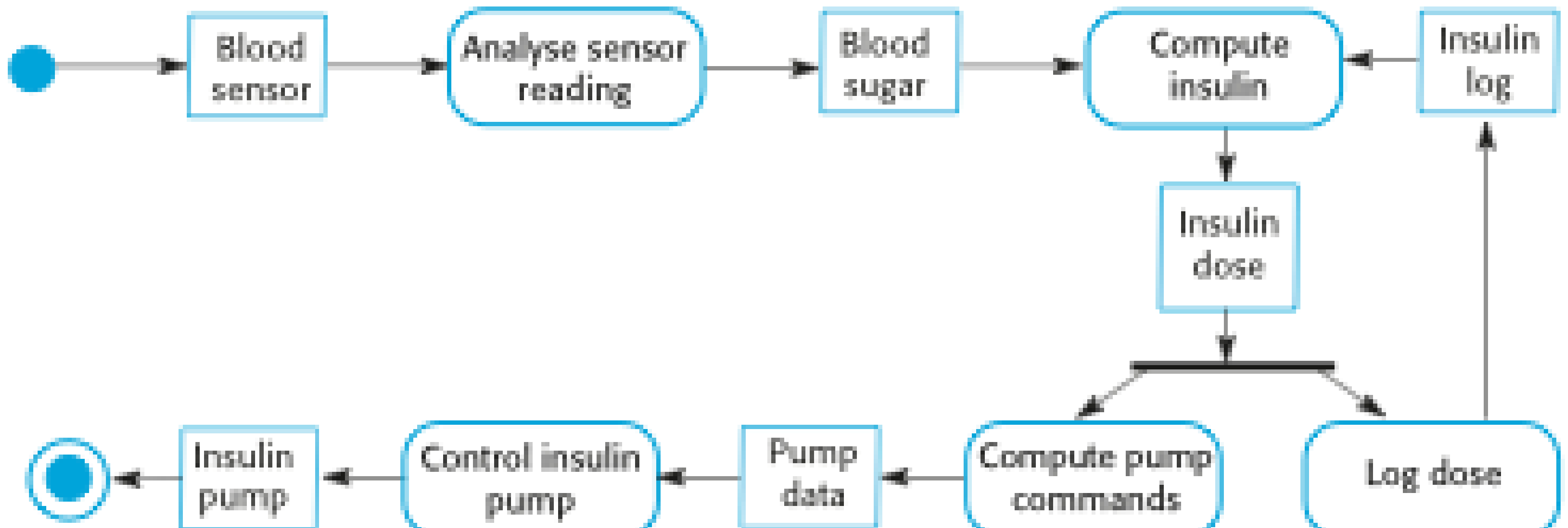
Insulin pump control system

- ✧ Collects data from a blood sugar sensor and calculates the amount of insulin required to be injected.
- ✧ Calculation based on the rate of change of blood sugar levels.
- ✧ Sends signals to a micro-pump to deliver the correct dose of insulin.
- ✧ Safety-critical system as low blood sugars can lead to brain malfunctioning, coma and death; high-blood sugar levels have long-term consequences such as eye and kidney damage.

Insulin pump hardware architecture



Activity model of the insulin pump



Essential high-level requirements

- ✧ The system shall be available to deliver insulin when required.
- ✧ The system shall perform reliably and deliver the correct amount of insulin to counteract the current level of blood sugar.
- ✧ The system must therefore be designed and implemented to ensure that the system always meets these requirements.

A patient information system for mental health care

- ✧ A patient information system to support mental health care is a medical information system that maintains information about patients suffering from mental health problems and the treatments that they have received.
- ✧ Most mental health patients do not require dedicated hospital treatment but need to attend specialist clinics regularly where they can meet a doctor who has detailed knowledge of their problems.
- ✧ To make it easier for patients to attend, these clinics are not just run in hospitals. They may also be held in local medical practices or community centres.

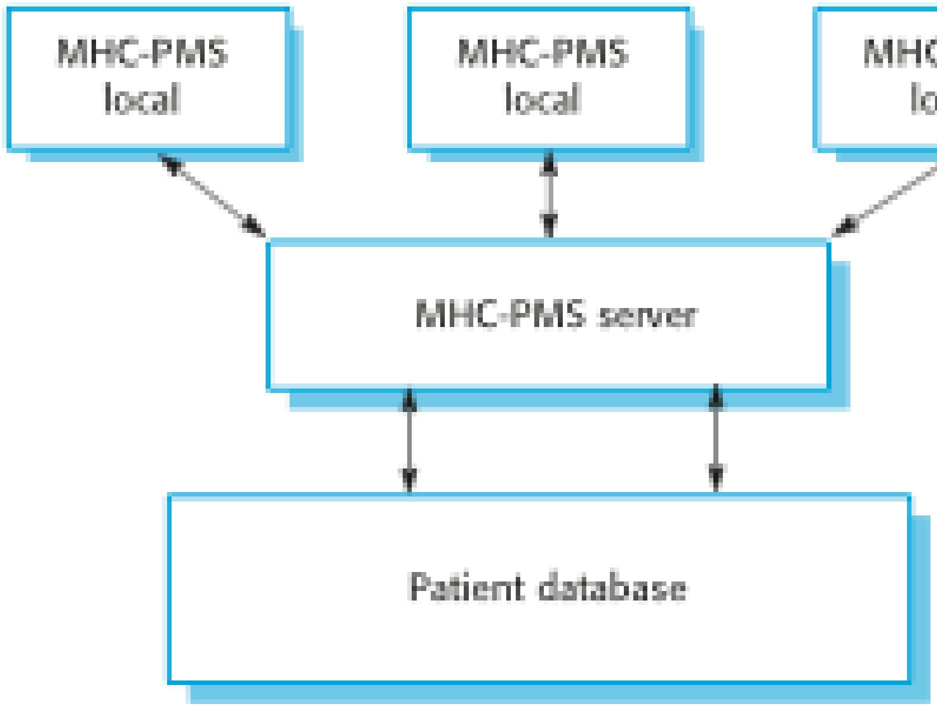
MHC-PMS

- ✧ The MHC-PMS (Mental Health Care-Patient Management System) is an information system that is intended for use in clinics.
- ✧ It makes use of a centralized database of patient information but has also been designed to run on a PC, so that it may be accessed and used from sites that do not have secure network connectivity.
- ✧ When the local systems have secure network access, they use patient information in the database but they can download and use local copies of patient records when they are disconnected.

MHC-PMS goals

- ✧ To generate management information that allows health service managers to assess performance against local and government targets.
- ✧ To provide medical staff with timely information to support the treatment of patients.

The organization of the MHC-PMS



MHC-PMS key features

- ✧ Individual care management
  - Clinicians can create records for patients, edit the information in the system, view patient history, etc. The system supports data summaries so that doctors can quickly learn about the key problems and treatments that have been prescribed.
- ✧ Patient monitoring
  - The system monitors the records of patients that are involved in treatment and issues warnings if possible problems are detected.
- ✧ Administrative reporting
  - The system generates monthly management reports showing the number of patients treated at each clinic, the number of patients who have entered and left the care system, number of patients sectioned, the drugs prescribed and their costs, etc.

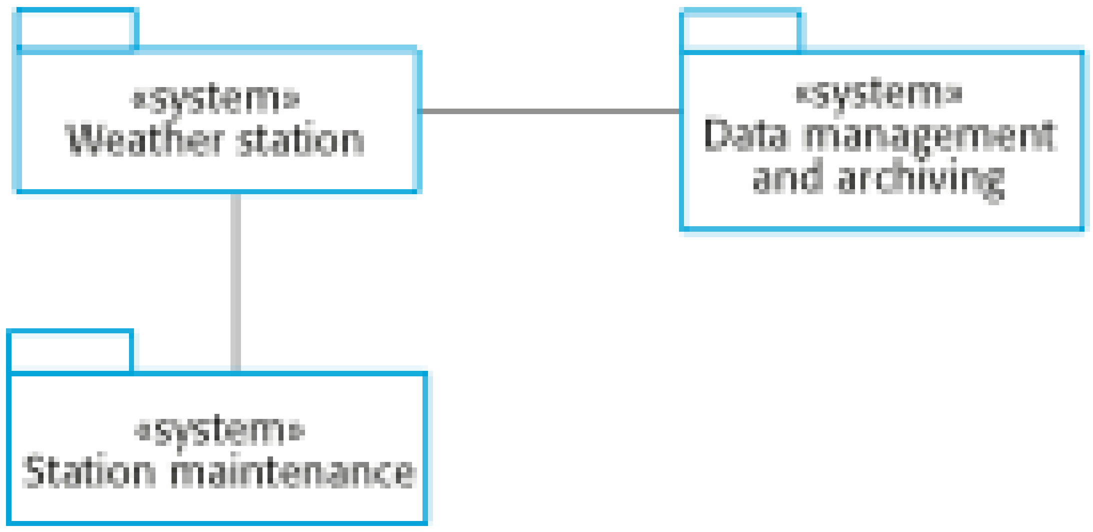
MHC-PMS concerns

- Privacy
  - It is essential that patient information is confidential and is never disclosed to anyone apart from authorised medical staff and the patient themselves.
- Safety
  - Some mental illnesses cause patients to become suicidal or a danger to other people. Wherever possible, the system should warn medical staff about potentially suicidal or dangerous patients.
  - The system must be available when needed otherwise safety may be compromised and it may be impossible to prescribe the correct medication to patients.

Wilderness weather station

- The government of a country with large areas of wilderness decides to deploy several hundred weather stations in remote areas.
- Weather stations collect data from a set of instruments that measure temperature and pressure, sunshine, rainfall, wind speed and wind direction.
  - The weather station includes a number of instruments that measure weather parameters such as the wind speed and direction, the ground and air temperatures, the barometric pressure and the rainfall over a 24-hour period. Each of these instruments is controlled by a software system that takes parameter readings periodically and manages the data collected from the instruments.

The weather station’s environment



Weather information system

- ✧ The weather station system
  - ▪ This is responsible for collecting weather data, carrying out some initial data processing and transmitting it to the data management system.
- ✧ The data management and archiving system
  - ▪ This system collects the data from all of the wilderness weather stations, carries out data processing and analysis and archives the data.
- ✧ The station maintenance system
  - ▪ This system can communicate by satellite with all wilderness weather stations to monitor the health of these systems and provide reports of problems.

Additional software functionality

- ✧ Monitor the instruments, power and communication hardware and report faults to the management system.
- ✧ Manage the system power, ensuring that batteries are charged whenever the environmental conditions permit but also that generators are shut down in potentially damaging weather conditions, such as high wind.
- ✧ Support dynamic reconfiguration where parts of the software are replaced with new versions and where backup instruments are switched into the system in the event of system failure.

Your Role as a Software Engineer

- Ethical Role

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.

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 outwit 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).

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.

Rationale for the 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:

Summary

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.

Key points

- ✧ Three case studies Explained from the book by Ian Sommerville:
  - An embedded insulin pump control system
  - A system for mental health care patient management
  - A wilderness weather station
- ✧ 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.