Software Engineering

Chapter 1- Introduction

Topics covered

- Professional software development
  - What is meant by software engineering.
- Software engineering ethics
  - A brief introduction to ethical issues that affect software engineering.
- Case studies
  - An introduction to three examples that are used in later chapters in the book.

Software engineering

- 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

Software Engineers are concerned with **developing software products** (i.e **software which can be sold to a customer).**

There ate two kinds of 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.

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 aspect**s 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.**
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.**

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.
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.

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

For two reasons:

1. 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***.

2. 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.

The **Systematic approach** that is used in **software engineering** is sometimes called a **software process.**

A software process is a ***sequence of activities*** that leads to the production of a **software product.**

Software process activities

1. Software **specification,** where **customers and engineers** define the ***software that is to be produced*** and the ***constraints on its operation.***
2. Software **development**, where the software is ***designed*** and ***programmed.***
3. Software **validation**, where the ***software is checked*** to ***ensure*** that ***it is what the customer requires***.
4. Software **evolution,** where 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 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

1. Confidentiality

- Engineers should normally respect the ***confidentiality*** of ***their employers or clients*** irrespective of **whether or not a formal confidentiality agreement has been signed**.

2. Competence

- Engineers ***should not misrepresent their level of competence.*** They ***should not knowingly accept work which is out with their competence.***

Issues of professional responsibility (cont..)

3. 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.***

4. 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 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 significantopportunities 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.

Ethical dilemmas

- **Disagreement in principle with the policies of senior management.**

- Your employer acts in an ***unethical way*** and **releases a safety-critical system without finishing the testing of the system.**

- Participation in the development of **military weapons systems** or **nuclear systems.**

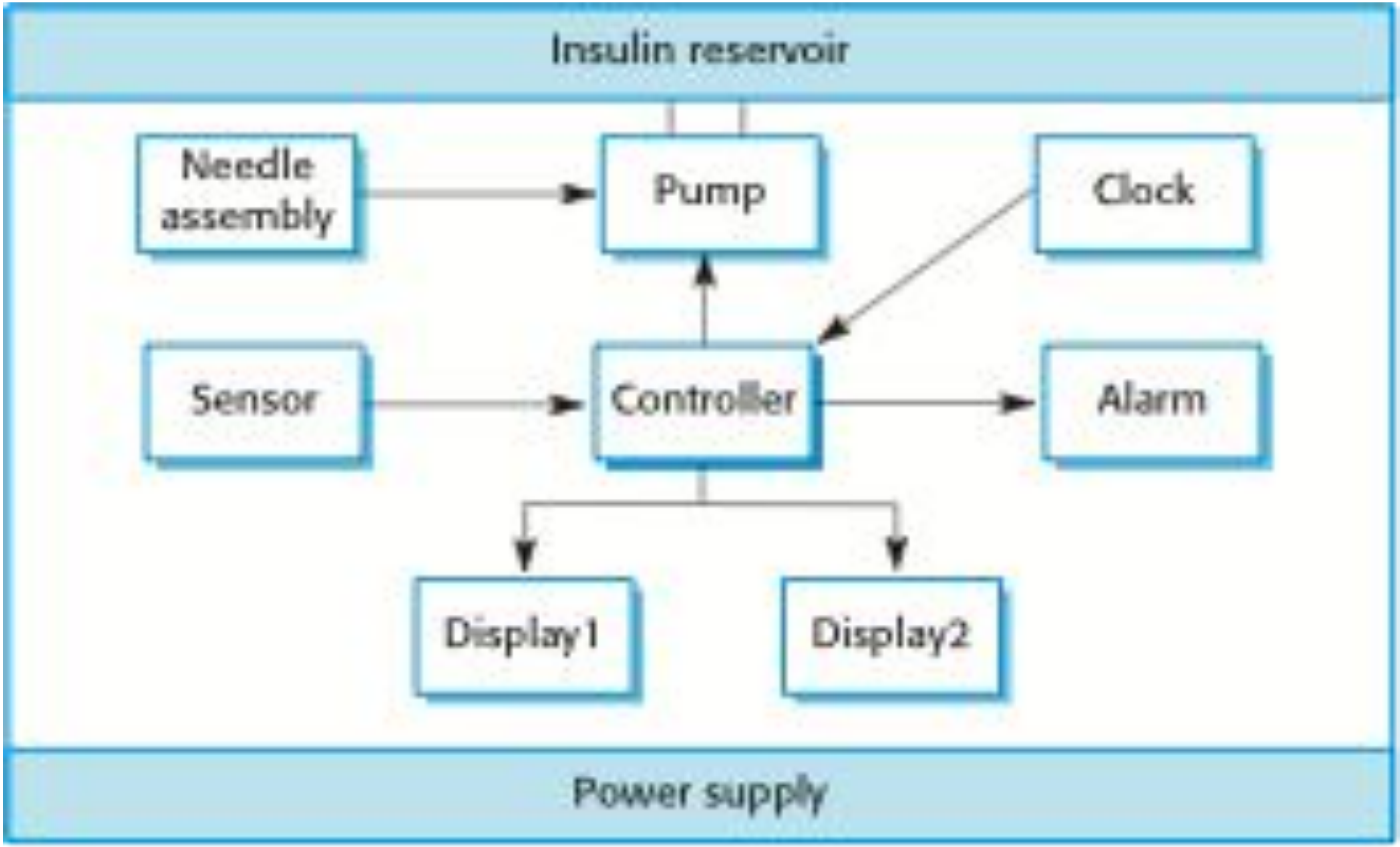
Case studies

- 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.

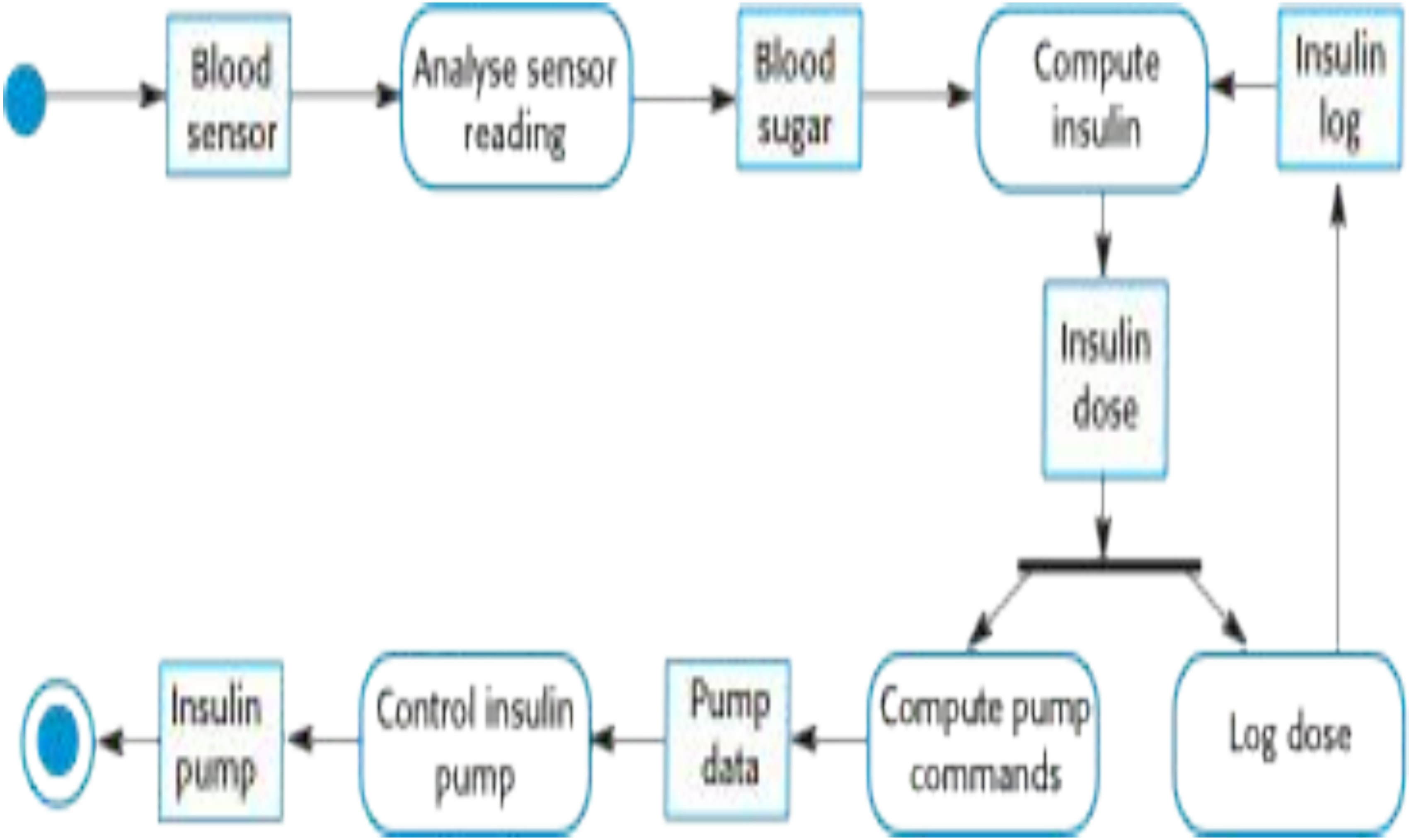
Insulin pump control system

- 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.***

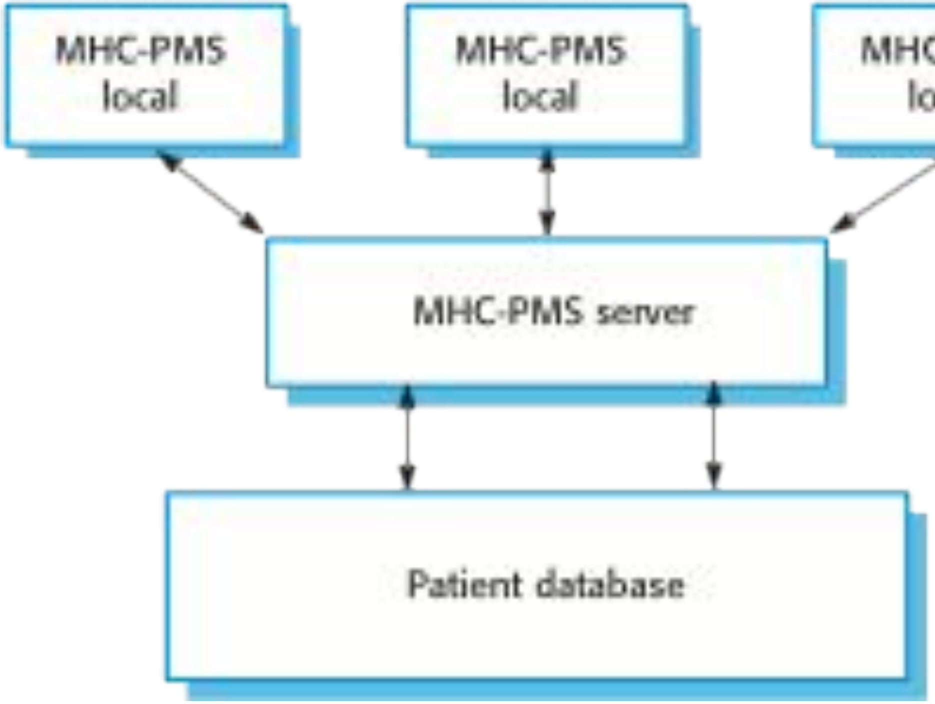
- When the local systems have secure network access, they use patient information in the database ***but* they can download and use local copies of patient records *when they are disconnected.***

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.***

Essential high-level requirements

- The overall design of the system has to take into account ***privacy*** and ***safety*** requirements.

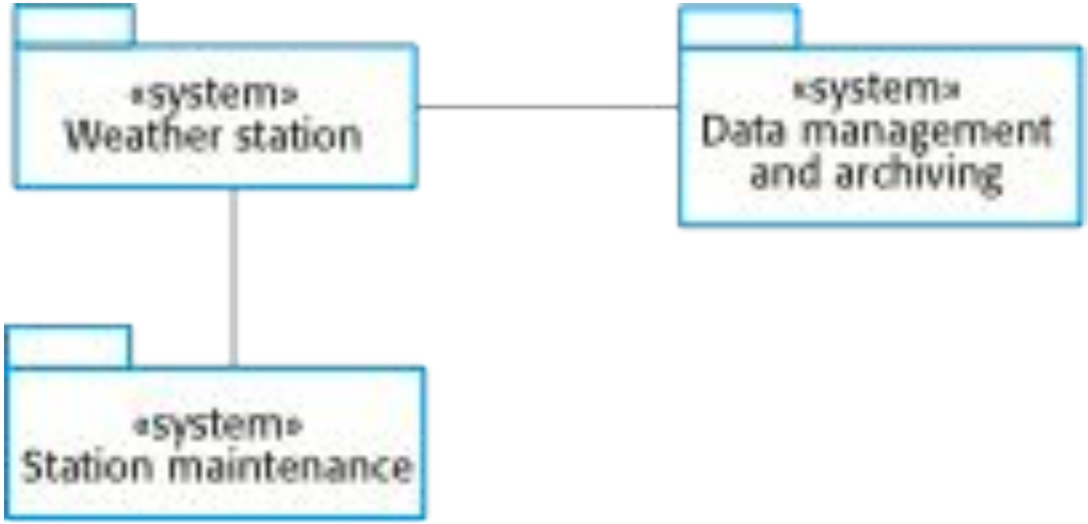
- The system must be ***availab***le when needed otherwise **safety must be compromised** and it may be **impossible to prescribe the correct medication to patients.**

- There is a **potential conflict here** – ***privacy is easiest to maintain when there is only a*** *single copy of the system data*. However, ***to ensure availability in the event of server failure*** or ***when disconnected from a network, multiple copies of the data should be maintained.***

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.

Key points

- 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.
- Three case studies are used in the book:
  - An embedded insulin pump control system
  - A system for mental health care patient management
  - A wilderness weather station

UNIT-1: INTRODUCTION_Review Questions

1. Answer following frequently asked questions about software Engineering? L1 6M

   i)Difference between software engineering and system engineering

   ii)What are the key challenges facing software engineering

   iii) what are the fundamental software engineering activities.

2. Define and Explain the difference between Generic and Customized product with example? L1,2 5M

3. List and Explain the essential attributes of good software? L1,2 4M

4. 4. Define Software Engineering? And Explain the four fundamental activities that are common to all software process? L1,2 6M

5. List and explain Issues that affect many different type of software? 6M

6. Define Software Engineering, List and explain essential attributes of good
software?L1,2 6M

7. List and explain any five Software Engineering (ACM/IEEE) Code of Ethics and
Professional Practices? L1,2 8M

8. List and Explain the issues related to software engineering ethics? L1,2 5M
9. Define software engineering. Explain any five FAQ of software engineering?L1,2 8M