FACULDADE ENGENHARIA
Departamento de Informática

Engenharia de Software
(14341, 16230, 15386)

Introduction

(adapted from lecture notes of Professor Sofia Ouhbi, Uppsala University,
Software Engineering: International Version, Ian Sommerville, Pearson, 2015, and
Engineering Software Products: An Introduction to Modern Software Engineering, Ian Sommerville, Pearson, 2020)

Topics covered

- ✧ Professional software development
- ✧ Software products
- ✧ Case studies
  - An introduction to three examples that are used in later lectures.

Professional software development

What is software?

1 Go to wooclap.com

2 Enter the event code in the top banner

Event code
UICIRZ

Software

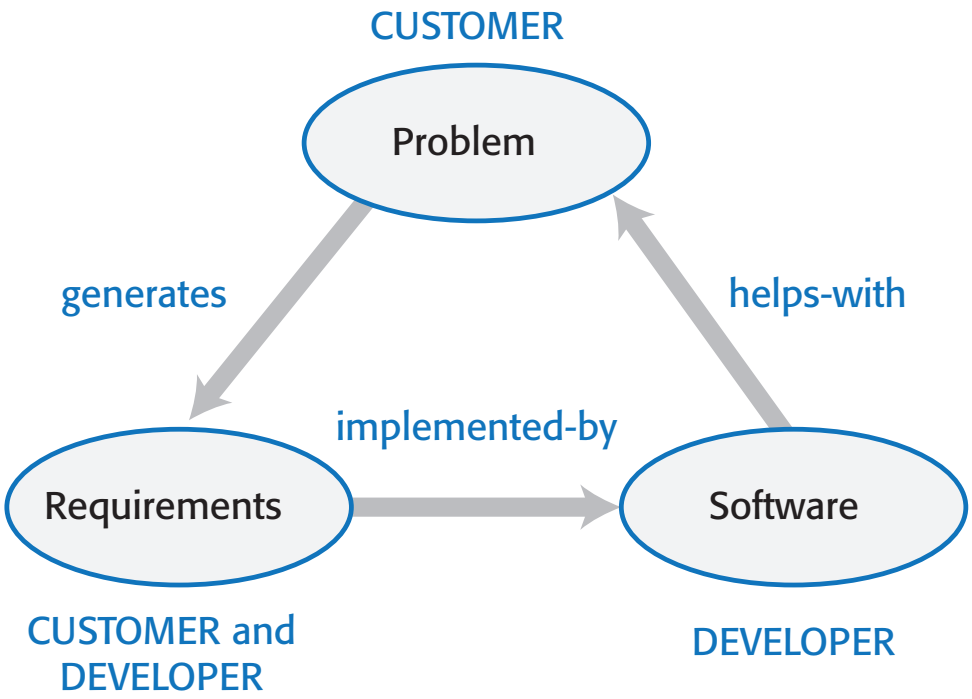
- Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.
- Generic products
  - Stand-alone systems that are marketed and sold to any customer who wishes to buy them.
  - Examples – PC software such as graphics 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.

Software products

Software products

- Software products are generic software systems that provide functionality that is useful to a range of customers.
- Many different types of products are available from large-scale business systems (e.g. MS Excel) through personal products (e.g. Evernote) to simple mobile phone apps and games (e.g. Sudoku).
- Software product engineering methods and techniques have evolved from software engineering techniques that support the development of one-off, custom software systems.
- Custom software systems are still important for large businesses, government and public bodies. They are developed in dedicated software projects.

Project-based software engineering



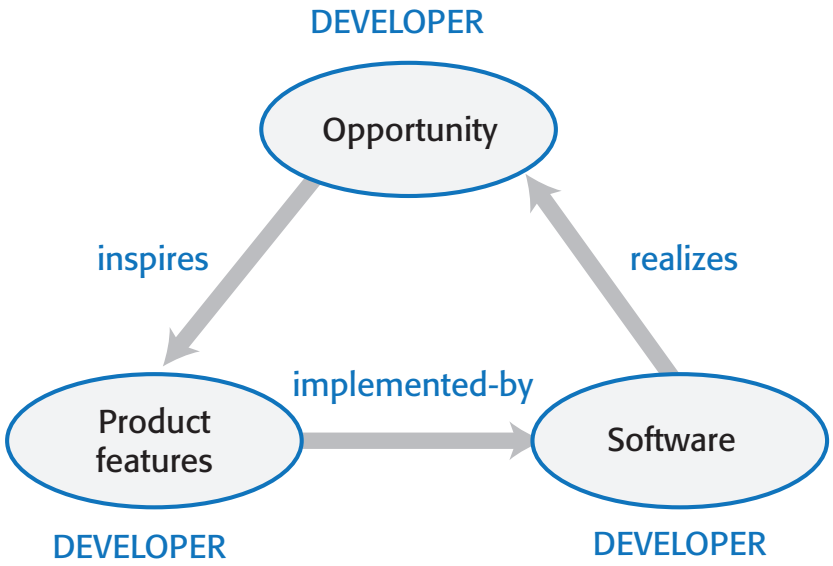
Project-based software engineering (1 of 2)

- The starting point for the software development is a set of ‘software requirements’ that are owned by an external client and which set out what they want a software system to do to support their business processes.
- The software is developed by a software company (the contractor) who design and implement a system that delivers functionality to meet the requirements.

Project-based software engineering (2 of 2)

- The customer may change the requirements at any time in response to business changes (they usually do). The contractor must change the software to reflect these requirements changes.
- Custom software usually has a long-lifetime (10 years or more) and it must be supported over that lifetime.

Product-based software engineering



Product software engineering (1 of 2)

- The starting point for product development is a business opportunity that is identified by individuals or a company. They develop a software product to take advantage of this opportunity and sell this to customers.
- The company who identified the opportunity design and implement a set of software features that realize the opportunity and that will be useful to customers.

Product software engineering (2 of 2)

- The software development company are responsible for deciding on the development timescale, what features to include and when the product should change.
- Rapid delivery of software products is essential to capture the market for that type of product.

Software product lines and platforms (1 of 2)

Technology	Description
Software product line	Software product line A set of software products that share a common core. Each member of the product line includes customer-specific adaptations and additions. Software product lines may be used to implement a custom system for a customer with specific needs that can't be met by a generic product. For example, a company providing communication software to the emergency services may have a software product line where the core product includes basic communication services such as receive and log calls, initiate an emergency response, pass information to vehicles, and so on. However, each customer may use different radio equipment and their vehicles may be equipped in different ways. The core product has to be adapted for each customer to work with the equipment that they use.

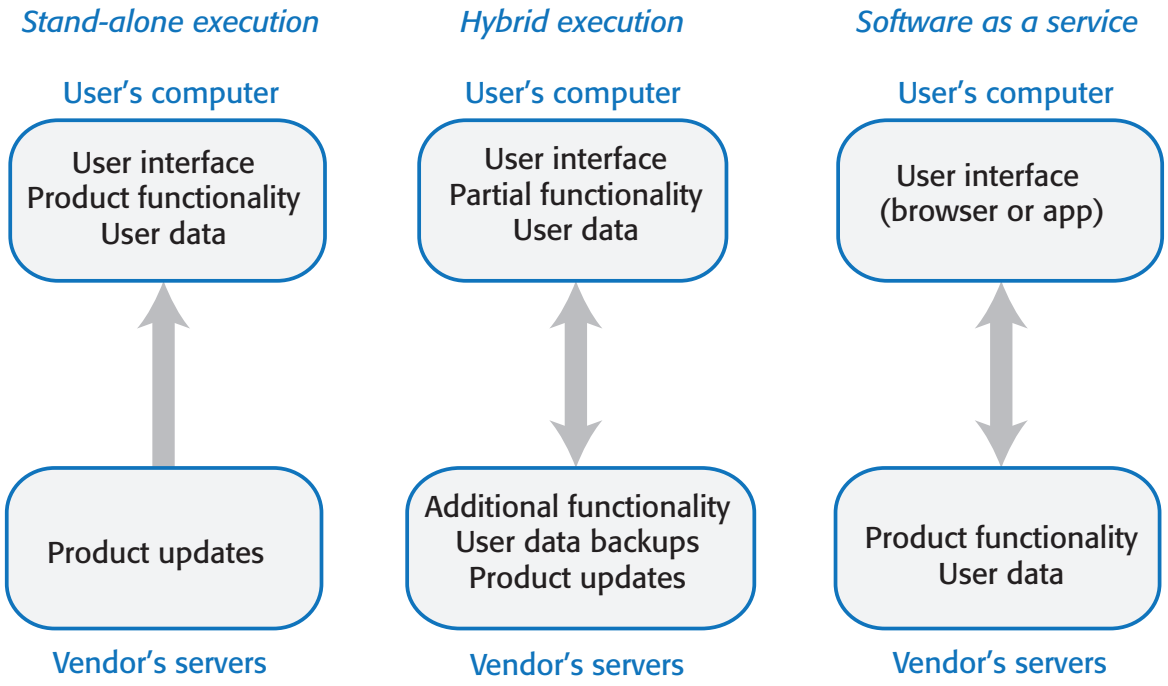
Software product lines and platforms (2 of 2)

Technology	Description
Platform	A software (or software+hardware) product that includes functionality so that new applications can be built on it. An example of a platform that you probably use is Facebook. It provides an extensive set of product functionality but also provides support for creating “Facebook apps.” These add new features that may be used by a business or a Facebook interest group.

Software execution models

- **Stand-alone** The software executes entirely on the customer's computers.
- **Hybrid** Part of the software's functionality is implemented on the customer's computer but some features are implemented on the product developer's servers.
- **Software service** All of the product's features are implemented on the developer's servers and the customer accesses these through a browser or a mobile app.

Software execution models



Comparable software development

- The key feature of product development is that there is no external customer that generates requirements and pays for the software. This is also true for other types of software development:
  - ***Student projects*** Individuals or student groups develop software as part of their course. Given an assignment, they decide what features to include in the software.
  - ***Research software*** Researchers develop software to help them answer questions that are relevant to their research.
  - ***Internal tool development*** Software developers may develop tools to support their work - in essence, these are internal products that are not intended for customer release.

The product vision

- The starting point for software product development is a ‘product vision’.
- Product visions are simple statements that define the essence of the product to be developed.
- The product vision should answer three fundamental questions:
  - What is the product to be developed?
  - Who are the target customers and users?
  - Why should customers buy this product?

Moore's vision template

- ✧ FOR (target customer)
- ✧ WHO (statement of the need or opportunity)
- ✧ The (PRODUCT NAME) is a (product category)
- ✧ THAT (key benefit, compelling reason to buy)
- ✧ UNLIKE (primary competitive alternative)
- ✧ OUR PRODUCT (statement of primary differentiation)

Vision template example

“FOR a mid-sized company”s marketing and sales departments WHO need basic CRM functionality, THE CRM-Innovator is a Web-based service THAT provides sales tracking, lead generation, and sales representative support features that improve customer relationships at critical touch points. UNLIKE other services or package software products, OUR product provides very capable services at a moderate cost.”

Information sources for developing a product vision
(1 of 2)

Information source	Explanation
Domain experience	The product developers may work in a particular area (say,marketing and sales) and understand the software support that they need. They may be frustrated by the deficiencies in the software they use and see opportunities for an improved system.
Product experience	Users of existing software (such as word processing software) may see simpler and better ways of providing comparable functionality and propose a new system that implements this. New products can take advantage of recent technological developments such as speech interfaces.

Information sources for developing a product vision
(2 of 2)

Information source	Explanation
Customer experience	The software developers may have extensive discussions with prospective customers of the product to understand the problems that they face; constraints, such as interoperability, that limit their flexibility to buy new software; and critical attributes of the software that they need.
Prototyping and "playing around"	Developers may have an idea for software but need to develop a better understanding of that idea and what might be involved in developing it into a product. They may develop a prototype system as an experiment and "play around" with ideas and variations using that prototype system as a platform.

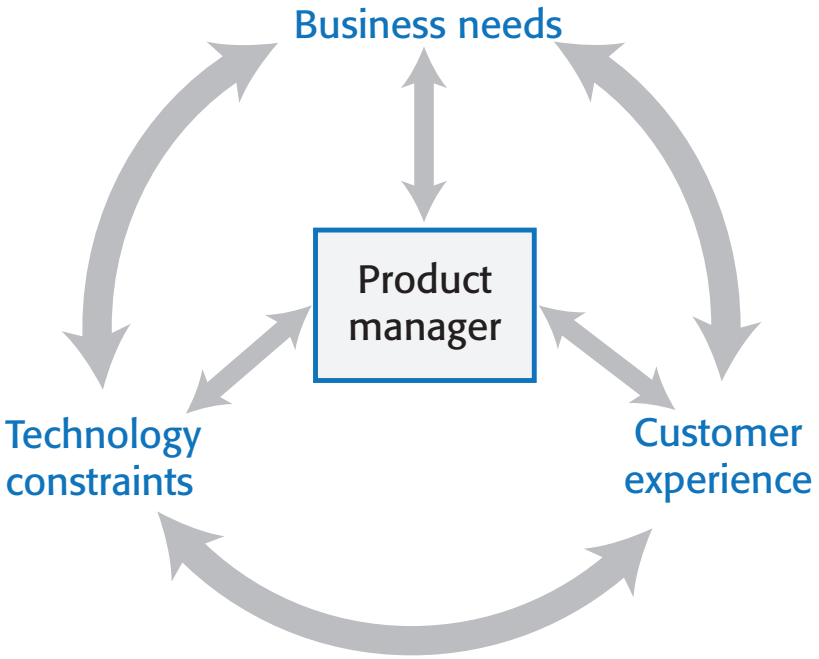
Software product management (1 of 2)

- Software product management is a business activity that focuses on the software products developed and sold by the business.
- Product managers (PMs) take overall responsibility for the product and are involved in planning, development and product marketing.

Software product management (2 of 2)

- Product managers are the interface between the organization, its customers and the software development team. They are involved at all stages of a product's lifetime from initial conception through to withdrawal of the product from the market.
- Product managers must look outward to customers and potential customers rather than focus on the software being developed.

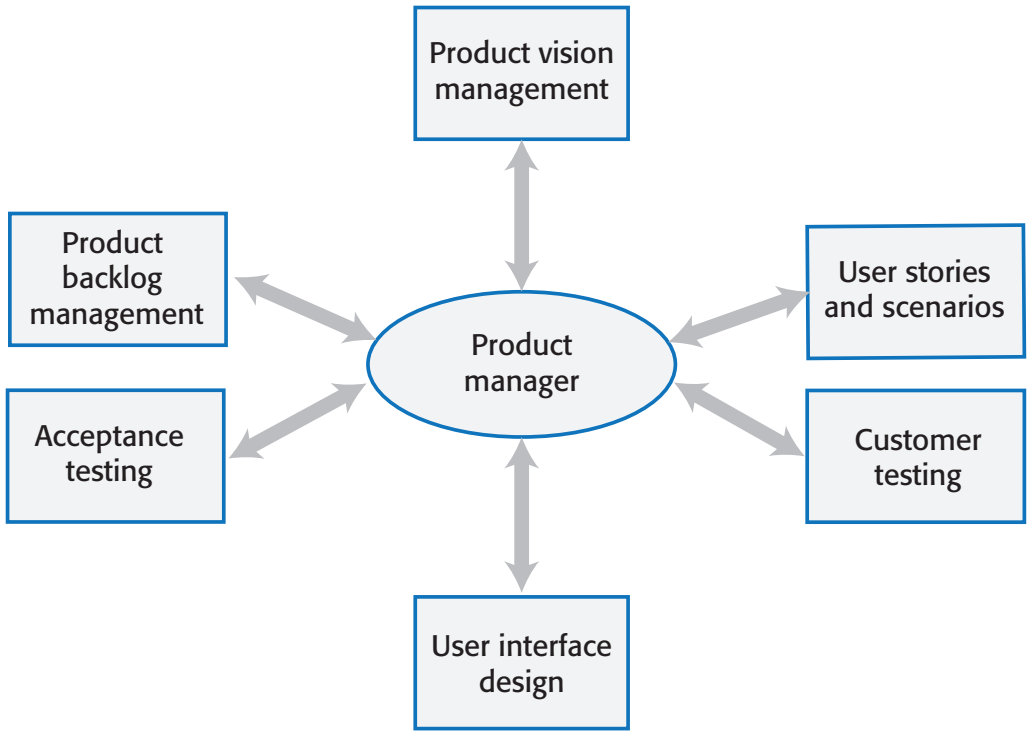
Product management concerns Business



Product management concerns

- ***Business needs*** PMs have to ensure that the software being developed meets the business goals of the software development company.
- ***Technology constraints*** PMs must make developers aware of technology issues that are important to customers.
- ***Customer experience*** PMs should be in regular contact with customers and potential customers to understand what they are looking for in a product, the types of users and their backgrounds and the ways that the product may be used.

Technical interactions of product managers



Technical interactions of product managers (1 of 4)

- Product vision management
  - The product manager may be responsible for helping with the development of the product vision. The should always be responsible for managing the vision, which involves assessing and evaluating proposed changes against the product vision. They should ensure that there is no ‘vision drift’
- Product roadmap development
  - A product roadmap is a plan for the development, release and marketing of the software. The PM should lead roadmap development and should be the ultimate authority in deciding if changes to the roadmap should be made.

Technical interactions of product managers (2 of 4)

- ✧ User story and scenario development
  - User stories and scenarios are used to refine a product vision and identify product features. Based on his or her knowledge of customers, the PM should lead the development of stories and scenarios.
- ✧ Product backlog creation and management
  - The product backlog is a prioritized ‘to-do’ list of what has to be developed. PMs should be involved in creating and refining the backlog and deciding on the priority of product features to be developed.

Technical interactions of product managers (3 of 4)

✧ Acceptance testing

- Acceptance testing is the process of verifying that a software release meets the goals set out in the product roadmap and that the product is efficient and reliable. The PM should be involved in developing tests of the product features that reflect how customers use the product.

✧ Customer testing

- Customer testing involves taking a release of a product to customers and getting feedback on the product's features, usability and business. PMs are involved in selecting customers to be involved in the customer testing process and working with them during that process.

Technical interactions of product managers (4 of 4)

✧ User interface design

- Product managers should understand user limitations and act as surrogate users in their interactions with the development team. They should evaluate user interface features as they are developed to check that these features are not unnecessarily complex or force users to work in an unnatural way.

Product prototyping (1 of 2)

- Product prototyping is the process of developing an early version of a product to test your ideas and to convince yourself and company funders that your product has real market potential.
  - You may be able to write an inspiring product vision, but your potential users can only really relate to your product when they see a working version of your software. They can point out what they like and don't like about it and make suggestions for new features.
  - A prototype may be also used to help identify fundamental software components or services and to test technology.

Product prototyping (2 of 2)

- Building a prototype should be the first thing that you do when developing a software product. Your aim should be to have a working version of your software that can be used to demonstrate its key features.
- You should always plan to throw-away the prototype after development and to re-implement the software, taking account of issues such as security and reliability.

Two-stage prototyping

- ***Feasibility demonstration*** You create an executable system that demonstrates the new ideas in your product. The aims at this stage are to see if your ideas actually work and to show funders and/or company management the original product features that are better than those in competing products.
- ***Customer demonstration*** You take an existing prototype created to demonstrate feasibility and extend this with your ideas for specific customer features and how these can be realized. Before you develop this type of prototype, you need to do some user studies and have a clearer idea of your potential users and scenarios of use.

Case studies

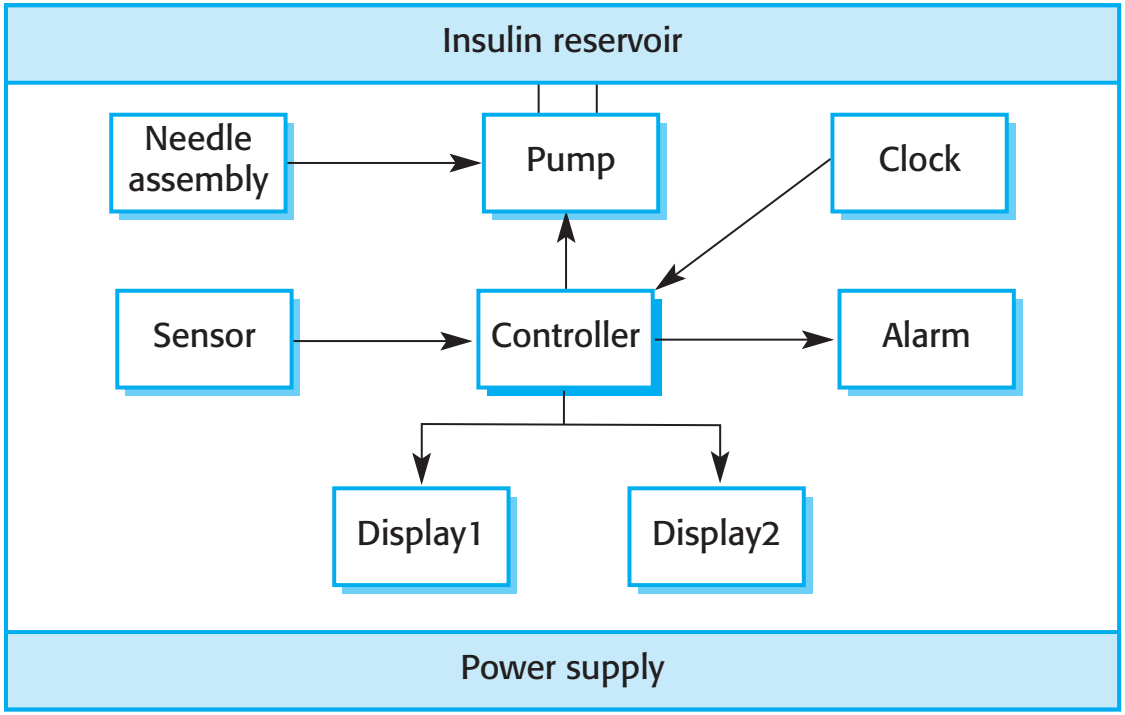
Case studies

- 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
  - Mentcare. 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.
- iLearn: a digital learning environment
  - A system to support learning in schools

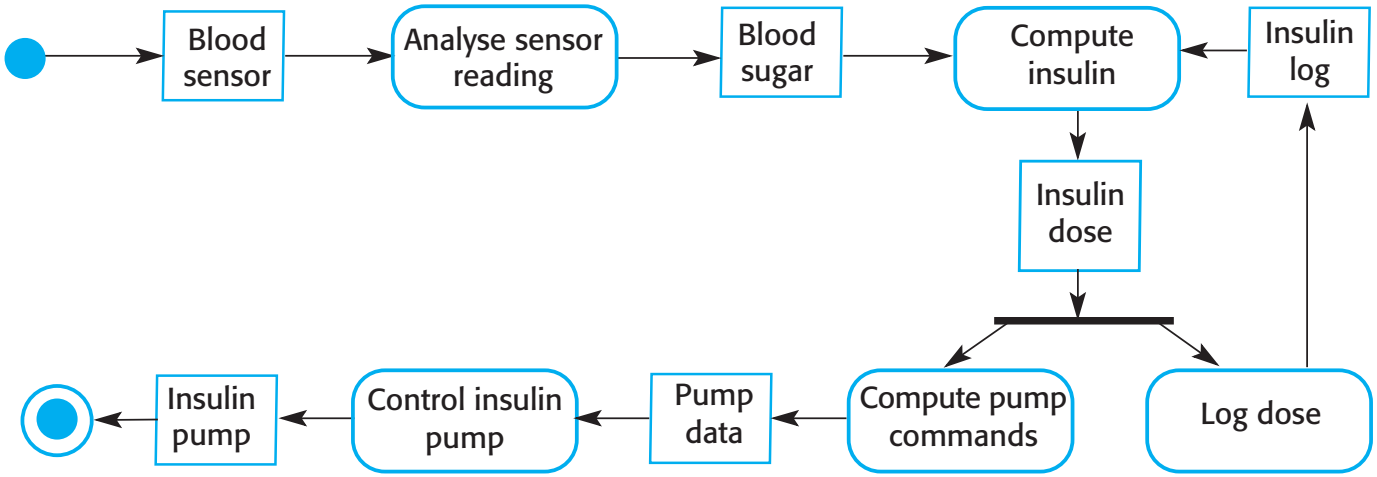
Insulin pump control system

- ✧ 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.

Mentcare: 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.

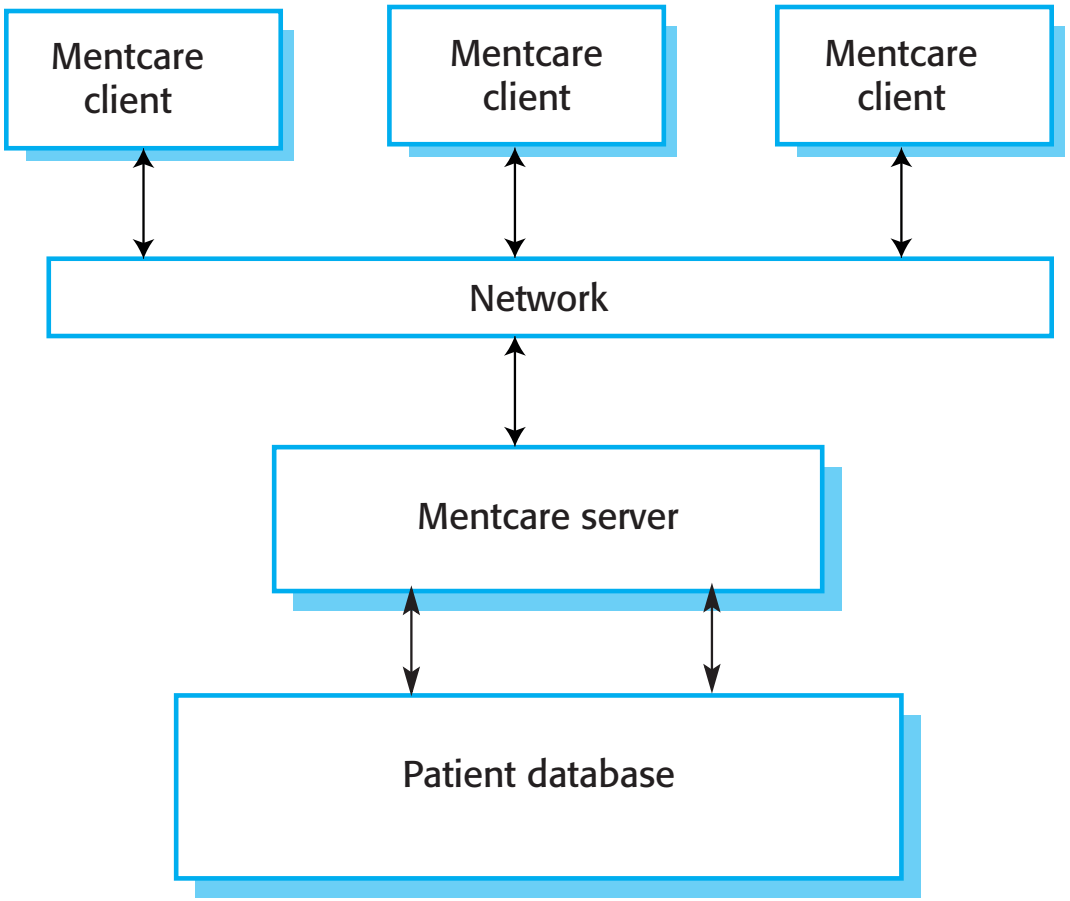
Mentcare

- Mentcare 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.

Mentcare 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 Mentcare system



Key features of the Mentcare system

- ✧ 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.

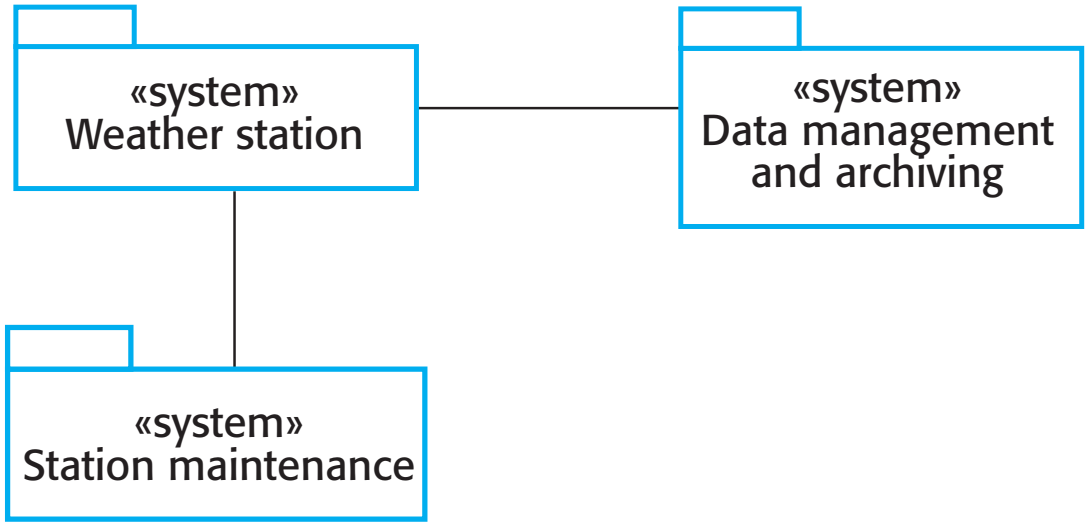
Mentcare system concerns

- ✧ 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.

iLearn: A digital learning environment

- A digital learning environment is a framework in which a set of general-purpose and specially designed tools for learning may be embedded plus a set of applications that are geared to the needs of the learners using the system.
- The tools included in each version of the environment are chosen by teachers and learners to suit their specific needs.
  - These can be general applications such as spreadsheets, learning management applications such as a Virtual Learning Environment (VLE) to manage homework submission and assessment, games and simulations.

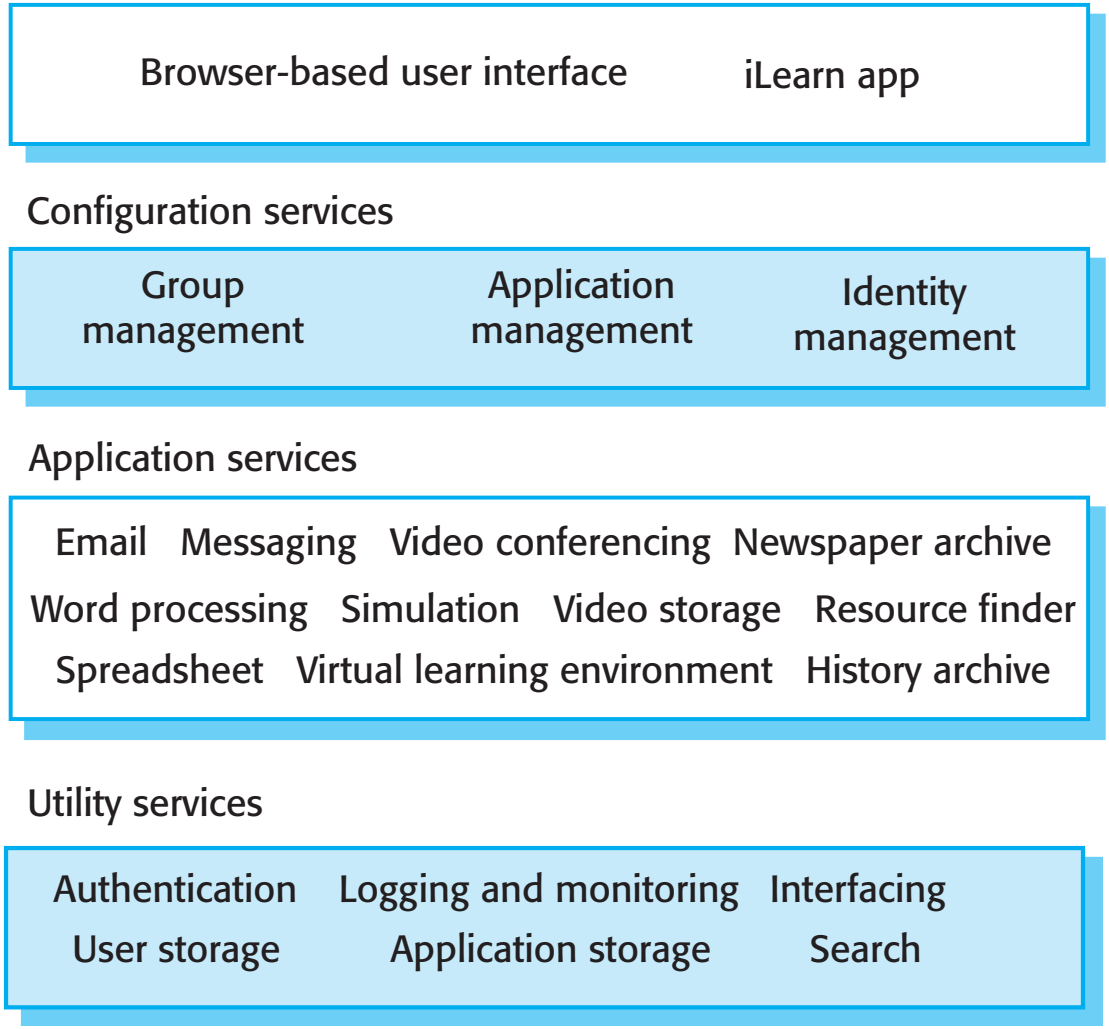
Service-oriented systems

- The system is a service-oriented system with all system components considered to be a replaceable service.
- This allows the system to be updated incrementally as new services become available.
- It also makes it possible to rapidly configure the system to create versions of the environment for different groups such as very young children who cannot read, senior students, etc.

iLearn services

- *Utility services* that provide basic application-independent functionality and which may be used by other services in the system.
- *Application services* that provide specific applications such as email, conferencing, photo sharing etc. and access to specific educational content such as scientific films or historical resources.
- *Configuration services* that are used to adapt the environment with a specific set of application services and do define how services are shared between students, teachers and their parents.

iLearn architecture



iLearn service integration

- *Integrated services* are services which offer an API (application programming interface) and which can be accessed by other services through that API. Direct service-to-service communication is therefore possible.
- *Independent services* are services which are simply accessed through a browser interface and which operate independently of other services. Information can only be shared with other services through explicit user actions such as copy and paste; re-authentication may be required for each independent service.

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 systems, 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.

Key points

- Software products are software systems that include general functionality that is likely to be useful to a wide range of customers.
- In product software engineering, the same company is responsible for deciding on the features that should be part of the product and the implementation of these features.
- Domain experience, product experience, customer experience and an experimental software prototype may all contribute to the development of the product vision.
- Product managers work at the interface between the business, the software development team and the product customers. They facilitate communications between these groups.


HOW
WHY
WHEN
WHERE
WHERE
WHY
WHERE
WHO
WHEN
WHAT
WHAT
WHY
WHO
WHAT
WHAT
WHEN
HOW
WHO
WHERE
WHEN
WHY
HOW