Lecture #1

CS310 – Software Engineering

Sultan S. Alqahtani

Assistance Professor
PhD in Computer Science (Software Engineering)
ssalqahtani@imamu.edu.sa
Office: 3092

Course Information

Course Information		
Credit Hours	3 credit hours 3 hours regular lecture	
Instructor	Name	Eman Alzaid
	Email	eazaid@imamu.edu.sa
Textbook	Required	Ian Sommerville, Software Engineering, 9th edition, Addison Wesley, 2010
Pre-requisites	CS242	Data Structures

Assessment	Assessment Task	Proportion
1	Quizzes	10%
2	Midterm Exam	25%
3	Workshops	5%
4	Group Project	20%
5	Final Exam	40%

Grade Scheme

List of Topics	No. of Weeks	Contact hours
Introduction to software planning & management		6
Software processes		3
Agile software development		3
Requirements engineering		6
System modeling		9
Architectural design		3
Design & implementation		6
Software testing & QA		6
Software project management		3

Topics to be Covered

Introduction

What is Software?

The product that software professionals build and then support over the long term.

Software encompasses: (1) instructions (computer programs) that when executed provide desired features, function, and performance; (2) data structures that enable the programs to adequately store and manipulate information and (3) documentation that describes the operation and use of the programs.

Software History

The early years
- Single user
- Limited distribution
- Custom software

The second era
- Multiuse
- Real-time
- Database
- Product software

The third era
- Distributed systems
- Embedded "intelligence"
- Low-cost hardware
- Consumer impact

The fourth era
- Powerful desk-top systems
- Object-oriented technologies
- Expert systems
- Parallel computing

The fifth era
- Handheld systems
- Mobile devices
- Intelligent systems
- Big data

1930 1960 1970 1980 1990 2000 2010

Developing 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 editing, 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.

Why Software is Important?

- The economies of ALL developed nations are dependent on software.
- Expenturdie on software represents a significant fraction of Gross National Product (GNP in all developed countries).
- More and more systems are software controlled (transportation, medical, telecommunications, military, industrial, entertainment,)
- Software engineering is concerned with theories, methods and tools for professional software development.

Software Problems

- As we move from one era to the next era, the problem associated with computer software continue to intensify:

1. Hardware sophistication has outpaced our ability to build software to tap hardware's potential.
2. Our ability to build new programs cannot keep pace with the demand for new programs.
3. Our ability to maintain existing programs is threatened by poor design and inadequate resources.

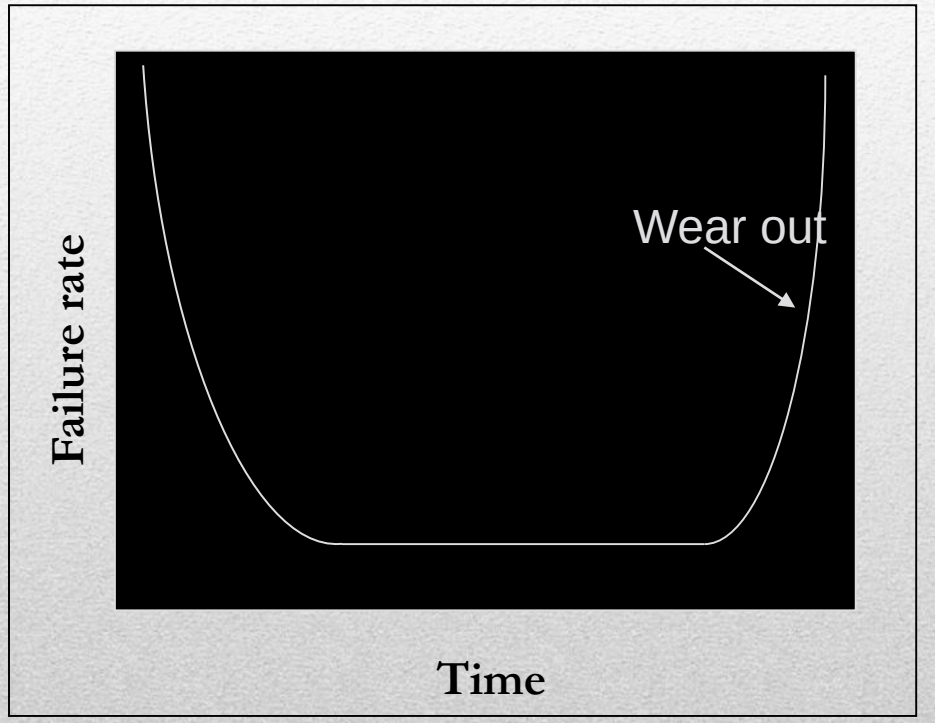
Solution to these problems is:

Software Engineering Practices

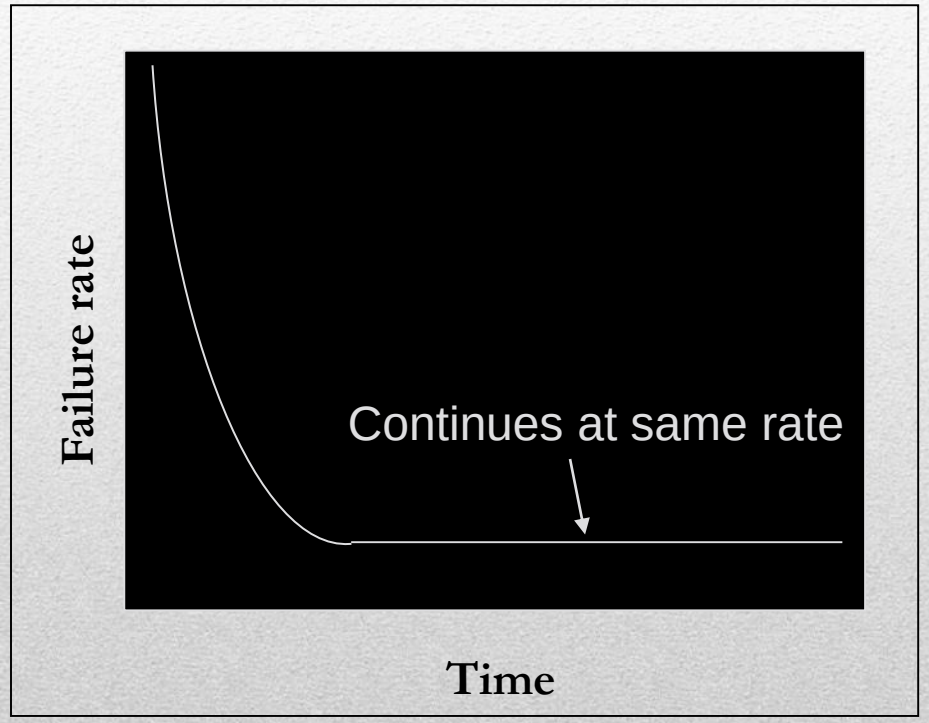
Software costs

- 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.

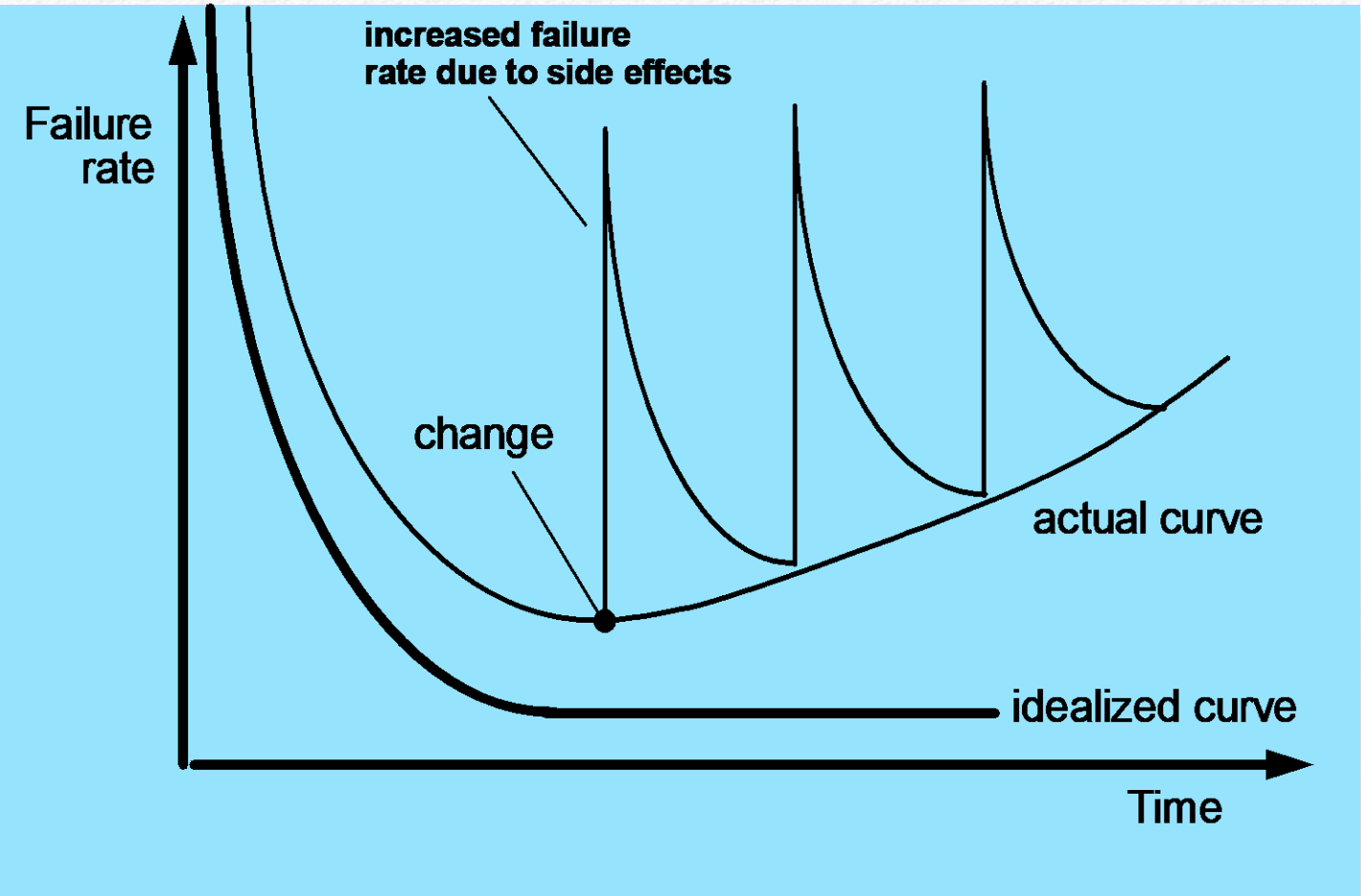
Software costs



Failure curve for hardware



Failure curve for software



Actual failure curve for software

Software Engineering Definition

The seminal definition:

[Software engineering is] the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.

The IEEE definition:

Software Engineering: (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1).

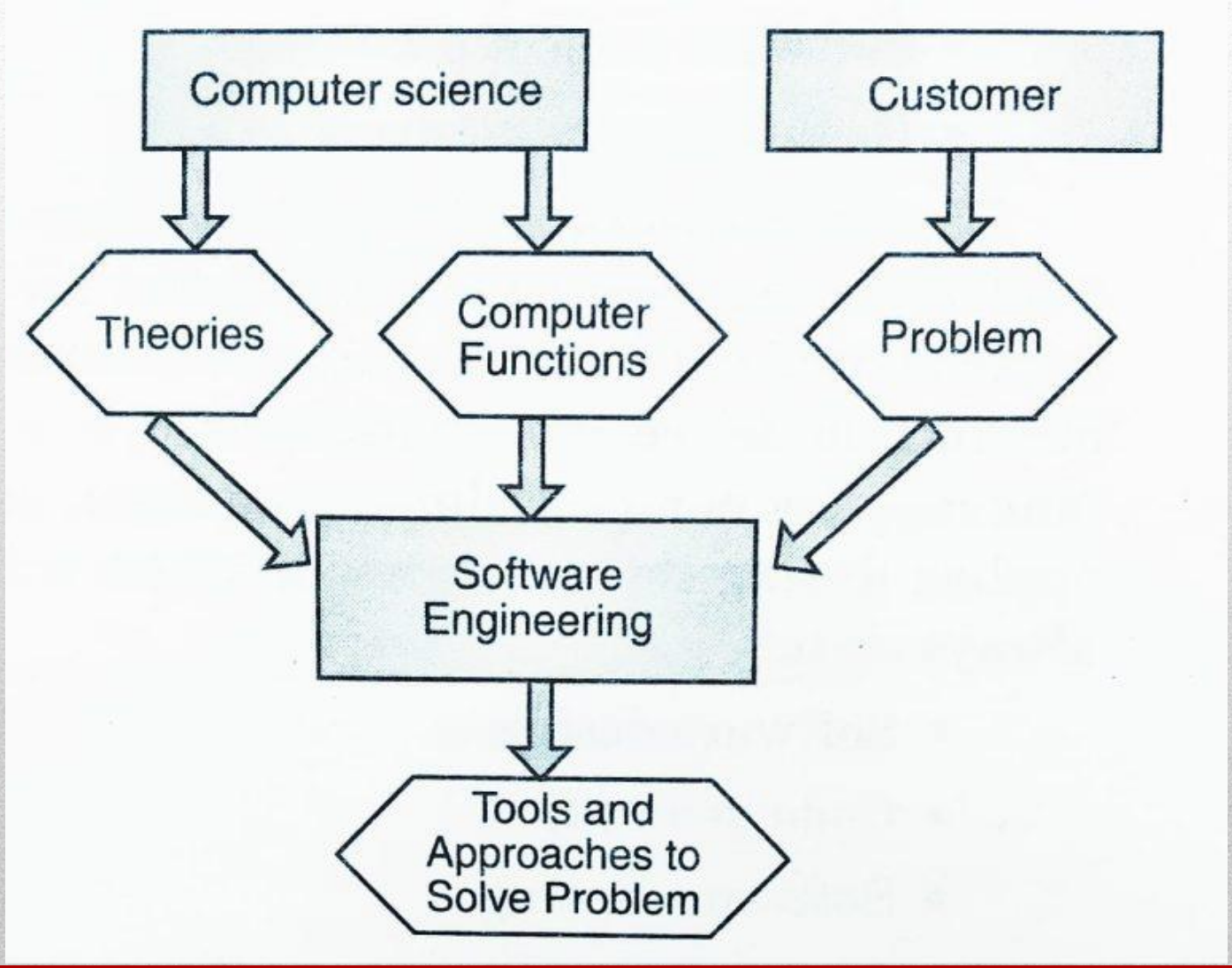
Importance of Software Engineering

- 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.

FAQ about software engineering

Question	Answer
What is software?	Computer programs, data structures 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.
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.

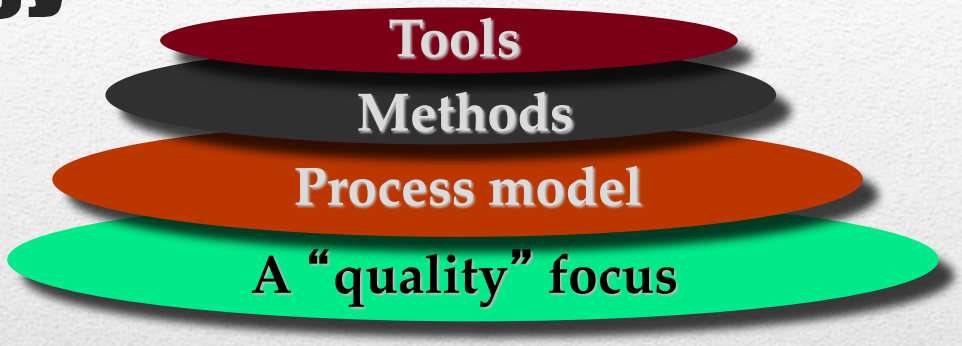
Software Engineering vs. Computer Science



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.

A Layered Technology



- Any engineering approach must rest on organizational commitment to **quality** which fosters a continuous process improvement culture.
- **Process** layer as the foundation defines a framework with activities for effective delivery of software engineering technology. Establish the context where products (model, data, report, and forms) are produced, milestone are established, quality is ensured and change is managed.
- **Method** provides technical how-to's for building software. It encompasses many tasks including communication, requirement analysis, design modeling, program construction, testing and support.
- **Tools** provide automated or semi-automated support for the process and methods.

Software process

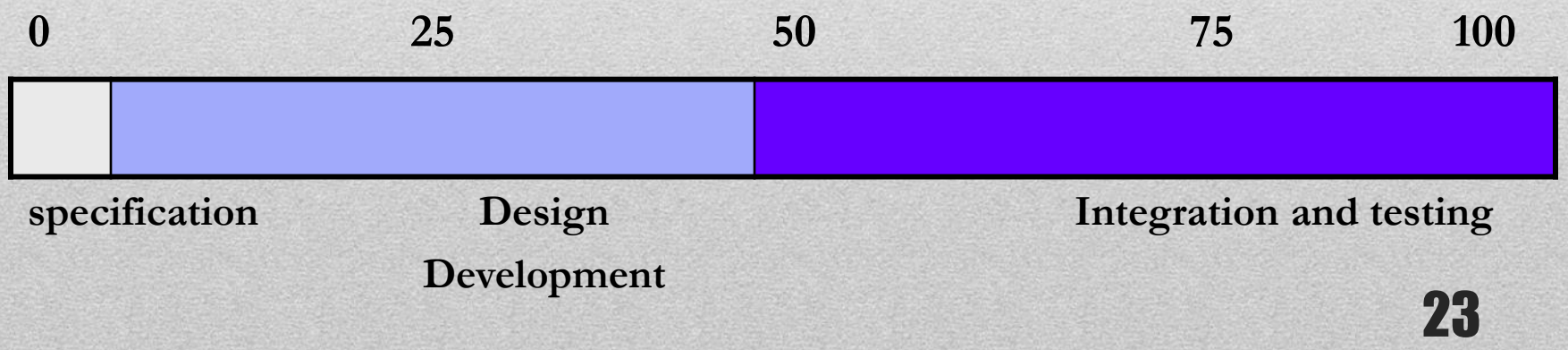
- A sequence of activities that leads to the production of a software product.
- Fundamental activities
  - Software specification (define the software that is to be produced and the constraints on its operation).
  - Software development (software is designed and programmed).
  - Software validation (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).

Software engineering diversity

- Application type
  - Stand-alone applications
  - Interactive transaction-based applications
  - Embedded control systems
  - Batch processing systems
  - Entertainment systems
  - Systems for modelling and simulation
  - Data collection systems
  - Systems of systems
- Customer's requirements
- Background of the development team.

Cost of Software Engineering

Roughly 20-50% of costs are development costs, 50-80% are testing costs.



Who does Software Engineering?

Customer

- The company, organization, or person who pays for the software system

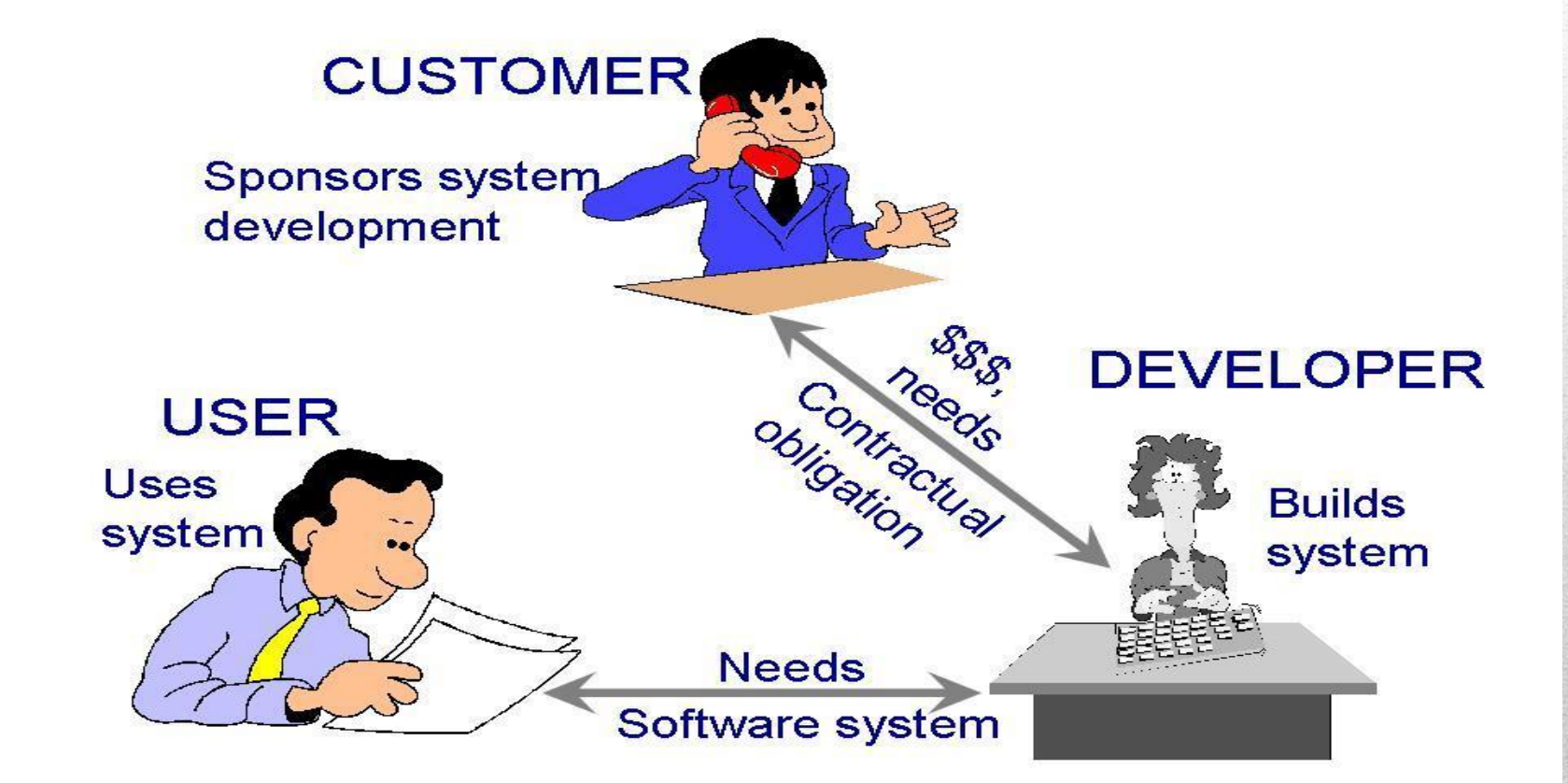
Developer

- The company, organization, or person who is building the software system

User

- The person or people who will actually use the system

Who does Software Engineering?



- **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.

Case studies

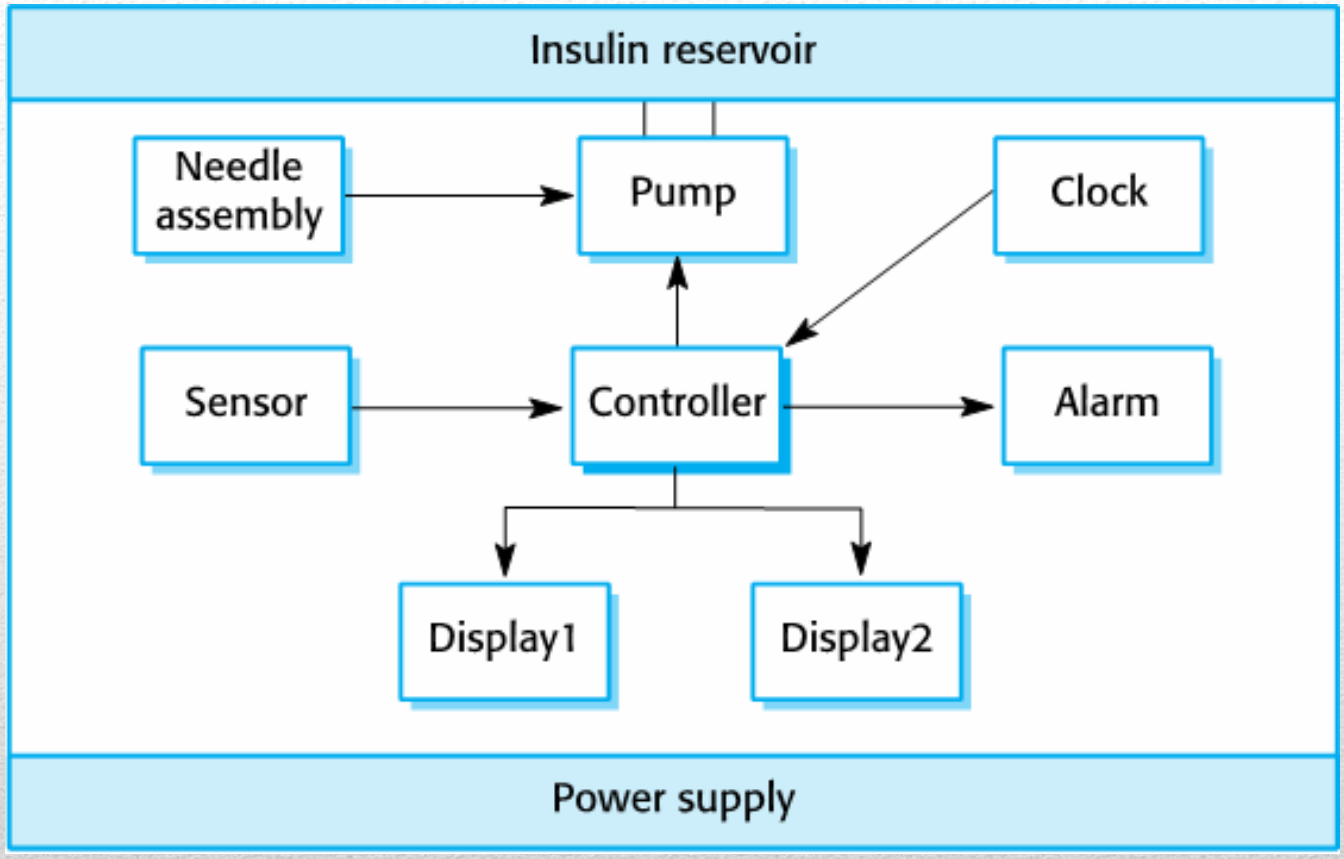
Collects data from a blood sugar sensor and calculates the amount of insulin required to be injected.

Calculation based on the rate of change of blood sugar levels.

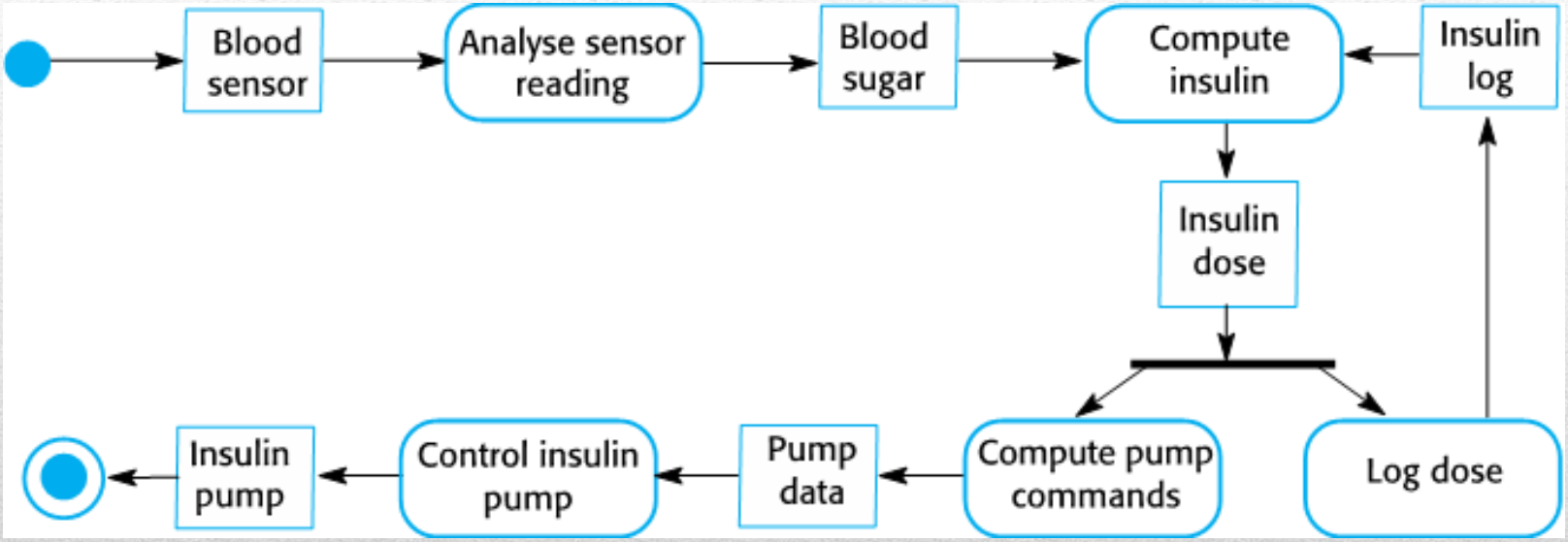
Sends signals to a micro-pump to deliver the correct dose of insulin.

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 control system



Insulin pump hardware architecture



Activity model of the insulin pump

- 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.

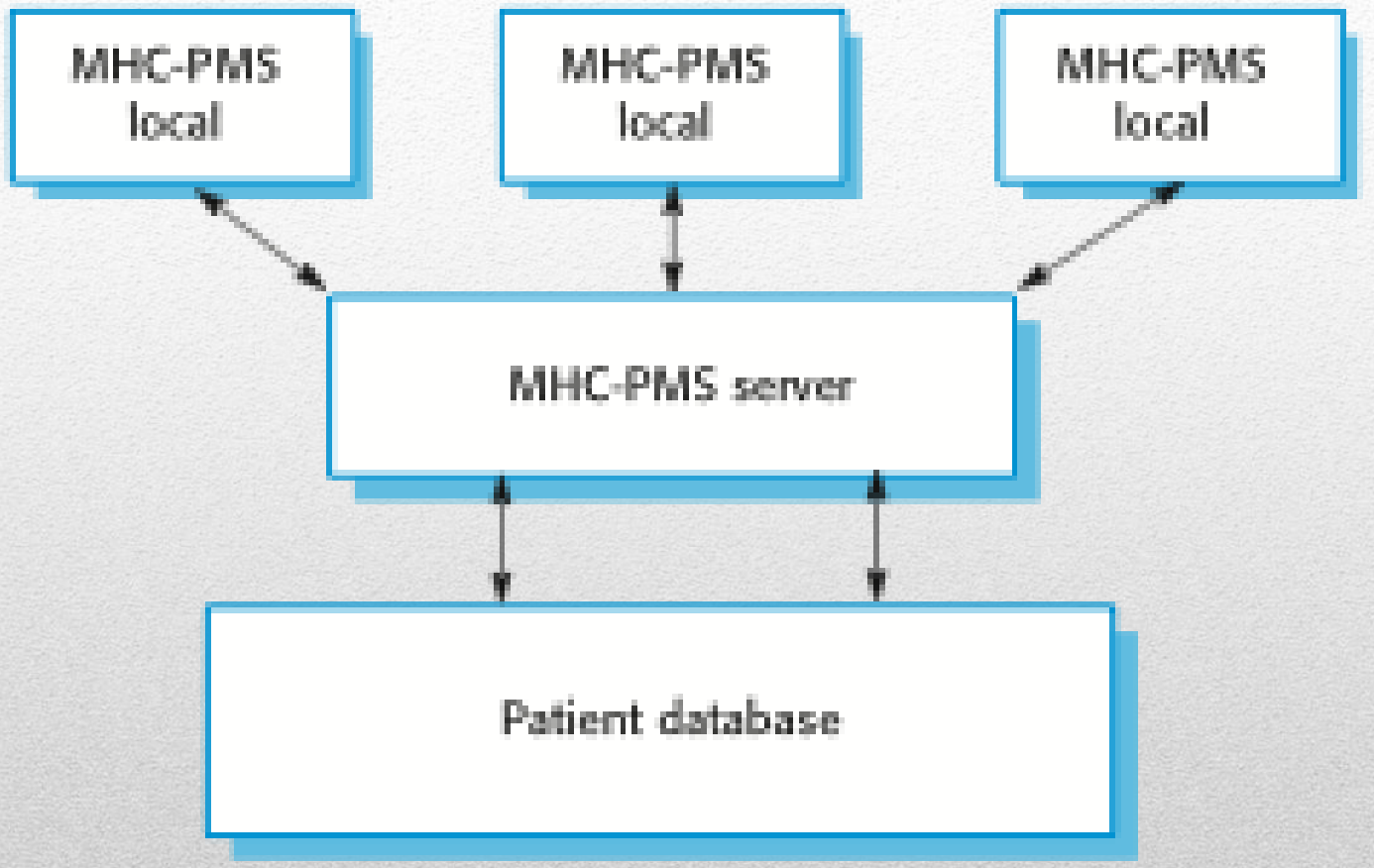
A patient information system for mental health care

- 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

- 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.

MHC-PMS goals



The organization of the MHC-PMS

- **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.

MHC-PMS concerns

Home assignment (reading) ☺

Wilderness weather station