

Chapter 1:

M.Mohmmad CIS
IT 

An Introduction to Software Engineering (**SE**)

Objectives

- To introduce (**SE**) and to explain its importance
- To understand the different types of software systems including critical systems
- To answer key questions and FAQs about **SE**
- To introduce ethical & professional issues concern **software engineers**

Importance of SE

- The economies of ALL developed nations are dependent on software.
إن اقتصادات جميع الدول المتقدمة تعتمد على البرمجيات.
- More and more systems are software controlled
المزيد والمزيد من النظم هي البرامج التي تسيطر عليها
- **SE** is concerned with theories, methods and tools for professional software development.
سي معنية بالنظريات والأساليب والأدوات لتطوير البرمجيات المهنية.
- Expenditure on software represents a significant fraction of **Gross National Product (GNP)** in all developed countries.
ويمثل الإنفاق على البرمجيات جزءا كبيرا من الناتج القومي الإجمالي في جميع البلدان المتقدمة النمو.

Software Costs

- Software costs often dominate computer system costs.
وكثيرا ما تهيمن تكاليف البرامج على تكاليف نظام الحاسوب.
- The costs of software on a PC are often greater than the hardware cost.
تكاليف البرمجيات على جهاز الكمبيوتر غالبا ما تكون أكبر من تكلفة الأجهزة.
- Software costs more to maintain than it does to develop.
تكاليف البرمجيات أكثر للحفاظ على مما هو عليه لتطوير.
- For systems with a long life, maintenance costs may be several times development costs.
بالنسبة للأنظمة ذات العمر الطويل، قد تكون تكاليف الصيانة عدة مرات تكاليف التطوير.
- **SE** is concerned with cost-effective software development.
هندسة البرمجيات المعنية تطوير البرمجيات فعالة من حيث التكلفة.

FAQs about SE

- What is software?
- What is SE?
- What is the relationship between SE and computer science?
- What is the difference between SE and system engineering?
- What is a software process?
- What is a software process model?

FAQs about SE (2)

- What are the costs of SE?
- What are SE methods?
- What is CASE-Tool (Computer-Aided Software Engineering)
- What are the attributes of good software?
- What are the key challenges facing SE?

What is Software?

- Computer programs and associated documentation such as requirements, design models and user manuals.
برامج الحاسوب والوثائق المرتبطة بها مثل المتطلبات ونماذج التصميم وأدلة المستخدم.
- Software products may be developed for a particular customer or may be developed for a general market.
ويمكن تطوير منتجات البرمجيات لعميل معين أو يمكن تطويرها لسوق عامة.
- Software products may be
– **Generic** (Horizontal Market application)- developed to be sold to a range of different customers e.g. PC software such as Excel or Word.
عام (تطبيق السوق الأفقي) - وضعت ليتم بيعها لمجموعة من العملاء المختلفة على سبيل المثال. برامج الكمبيوتر مثل إكسيل أو ورد.
– **Custom** (Vertical Market application)- developed for a single customer according to their specification.
مخصص (تطبيق السوق العمودي) - وضعت لعميل واحد وفقا لمواصفاتها.
- New software can be created by developing new programs, configuring generic software systems or reusing existing software.
يمكن إنشاء برامج جديدة من خلال تطوير برامج جديدة، وتكوين أنظمة برامج عامة أو إعادة استخدام البرامج الموجودة.

What is Software System?

- A software system consists of:
 - No. of separate programs operate correctly .
عدد البرامج المنفصلة يعمل بشكل صحيح.
 - Configuration files; used to setup the programs .
ملفات التكوين؛ تستخدم لإعداد البرامج.
 - System-documentation; describes sys. Structure
توثيق النظام. بنيه النظام.
 - User-documentation explain how to use the application /system
توضح وثائق المستخدم كيفية استخدام التطبيق / النظام
 - Web-site for users to download recent product information, and to get technical support from vendors.
موقع ويب للمستخدمين لتحميل معلومات المنتج الأخيرة، والحصول على الدعم الفني من البائعين.

What is **SE**?

- **SE** is an engineering discipline that is concerned with all aspects of software production.

هندسة البرمجيات هو الانضباط الهندسي الذي يهتم بجميع جوانب إنتاج البرمجيات.

- **Software engineers** should adopt a systematic and organised approach to their work and use appropriate tools and techniques depending on the problem to be solved, the development constraints and the resources available.

وينبغي لمهندسي البرمجيات أن يعتمدوا نهجا منظما ومنهجيا في عملهم وأن يستعملوا الأدوات والأساليب المناسبة تبعا للمشكلة التي يتعين حلها والقيود الإنمائية والموارد المتاحة.

Definition

(i). Software engineering is the study of the principles and methodologies for developing and maintaining software system.

هندسة البرمجيات هي دراسة المبادئ والمنهجيات لتطوير وصيانة نظام البرمجيات.

[Ref. Zelkowitz 1978]

Definition

(ii). Software engineering is the practical application of scientific knowledge in the **design** and construction of computer programs, and the associated documentation required to develop, **operate**, and maintain them.

هندسة البرمجيات هي التطبيق العملي للمعرفة العلمية في تصميم وبناء برامج الكمبيوتر، والوثائق المرتبطة بها لتطويرها وتشغيلها وصيانتها.

[Ref. Boehm 1976]

Definition

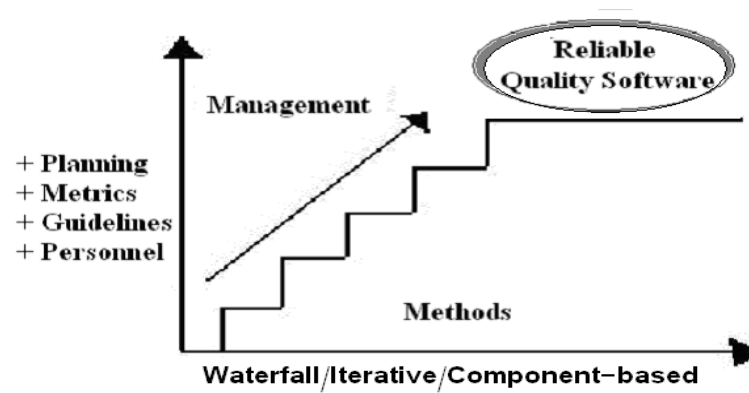
(iii). Software engineering deals with the establishment of sound engineering principles and methods in order to **economically** obtain software that is reliable and works on real machines .

. هندسة البرمجيات تتعامل مع وضع مبادئ وأساليب الهندسة السليمة من أجل الحصول اقتصاديا البرمجيات التي يمكن الاعتماد عليها ويعمل على الآلات الحقيقية.

[Ref. Baue 1972]

Two Aspects of the SE Process

Management Vs Methods



SE Vs Computer Science?

- **Computer Science** is concerned with theory and fundamentals.
علوم الحاسب الآلي المعنية النظرية والأساسيات.
- **SE** is concerned with the practicalities of developing and delivering useful software.
هندسة البرمجيات يهتم بالعملية من تطوير وتقديم برامج مفيدة.

SE Vs System Engineering?

- **System engineering** is concerned with all aspects of computer-based systems development including hardware, software and process engineering.

هندسة النظام هي المعنية بجميع جوانب تطوير النظم القائمة على الكمبيوتر بما في ذلك الأجهزة والبرمجيات وهندسة العمليات.

- **SE** is part of this process concerned with developing the software infrastructure, control, applications and databases in the system.

هندسة البرمجيات هو جزء من هذه العملية المعنية بتطوير البنية التحتية للبرمجيات والتحكم والتطبيقات وقواعد البيانات في النظام.

What is a **Software Process**?

- **A set of activities whose goal is the development or evolution of software.**

مجموعة من الأنشطة التي تهدف إلى تطوير أو تطور البرمجيات.

- Generic activities in all software processes are:

الأنشطة العامة في جميع عمليات البرمجيات هي:

- Specification - what the system should do and its development constraints

مواصفات - ما يجب على النظام القيام به و القيود

- Development - production of the software system

تطوير - إنتاج نظام البرمجيات

- Validation - checking that the software is what the customer wants

التحقق من صحة - التحقق من أن البرنامج هو ما يريده العميل

- Evolution - changing the software in response to changing demands.

التطور - تغيير البرنامج استجابة للمتطلبات المتغيرة.

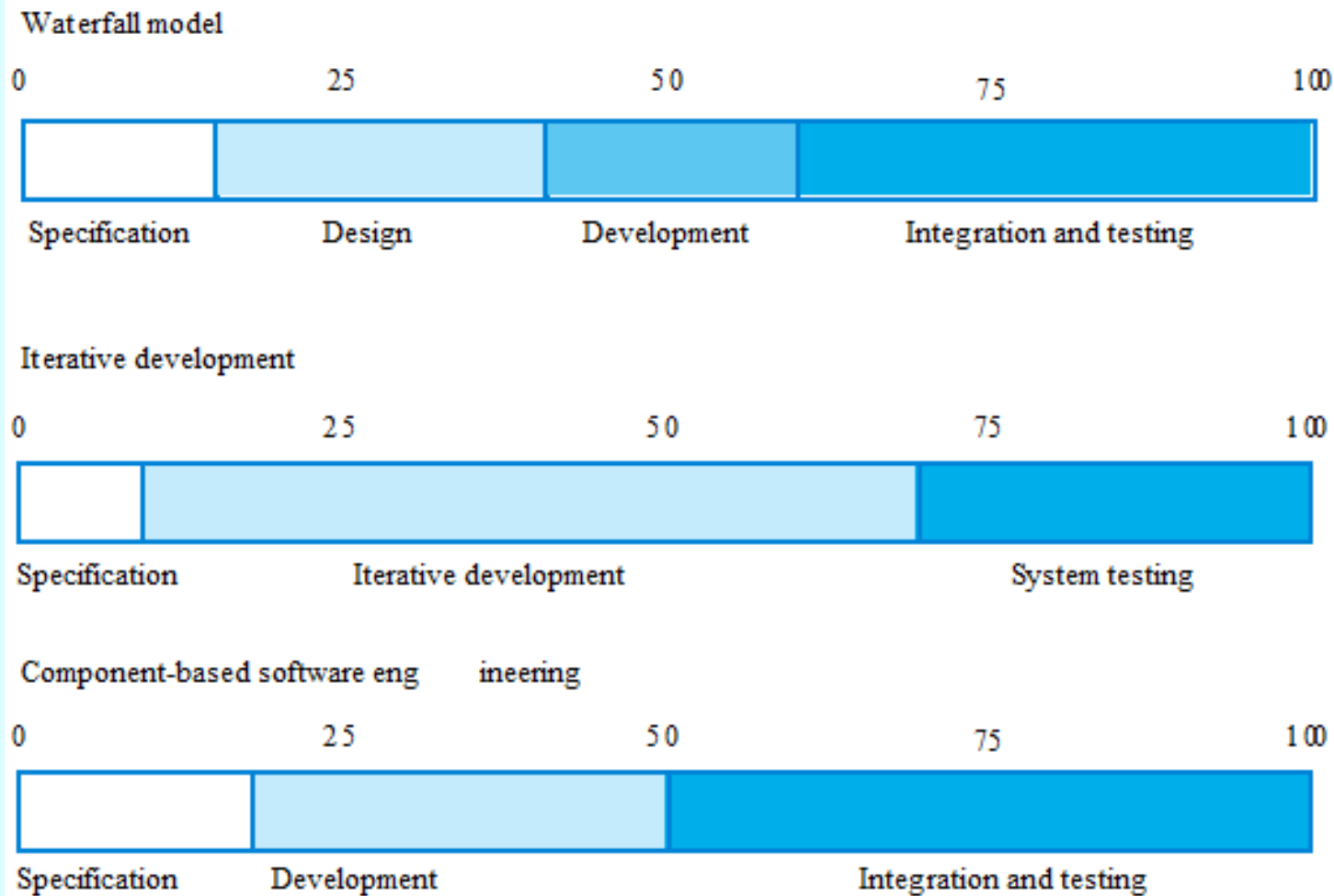
What is a **Software Process Model**?

- A simplified representation of a software process, presented (usually graphical) from a specific perspective.
تمثيل مبسط لعملية البرمجيات، عرض (عادة رسومية) من منظور معين.
- **Generic process models** نماذج عملية عامة
 - **Waterfall;**
 - **Iterative development (including Prototyping → Agile);**
مرنه/ذكيه).[?]التنمية التكرارية (بما في ذلك النماذج
 - **Component-based software engineering.**
هندسة البرمجيات القائمة على المكونات.

What are the costs of **SE**?

- Roughly 60% of costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs.
حوالي ٦٠٪ من التكاليف هي تكاليف التطوير، ٤٠٪ هي تكاليف الاختبار. بالنسبة لبرامج مخصصة، تكاليف التطوير غالبا ما تتجاوز تكاليف التطوير.
- Costs vary depending on the type of system being developed and the requirements of system attributes such as performance and system reliability.
وتختلف التكاليف تبعا لنوع النظام الجاري تطويره ومتطلبات خصائص النظام مثل الأداء وموثوقية النظام.
- Distribution of costs depends on the development model that is used.
يعتمد توزيع التكاليف على نموذج التطوير المستخدم.

Activity Cost Distribution



What are **SE Methods**?

- Structured approaches to software development which include system models, notations, rules, design advice and process guidance.
نهج منظمة لتطوير البرمجيات التي تشمل نماذج النظام، والرموز، والقواعد، وتقديم المشورة في مجال التصميم وتوجيه العملية.
- Model descriptions وصف النموذج
 - Descriptions of graphical models which should be produced;
مواصفات النماذج الرسومية التي ينبغي إنتاجها؛
- Rules قواعد
 - Constraints applied to system models;
القيود المطبقة على نماذج النظام؛
- Recommendations توصيات
 - Advice on good design practice;
المشورة بشأن ممارسة التصميم الجيد؛
- Process guidance ارشادات عملية
 - What activities to follow.
ما هي الأنشطة التي يجب اتباعها.

What are the Critical Systems?

- **Safety**-critical systems أنظمة السلامة الحرجة
 - Failure results in loss of life, injury or damage to the environment;
ويؤدي الفشل إلى فقدان الأرواح أو الإصابة أو الإضرار بالبيئة؛
eg. Chemical plant protection system;
على سبيل المثال. نظام حماية النباتات الكيميائية.
- **Mission**-critical systems النظم ذات الأهمية الحرجة
 - Failure results in failure of some goal-directed activity;
فشل النتائج في فشل بعض الأنشطة الموجهة نحو الهدف؛
Eg. Spacecraft navigation system;
على سبيل المثال. نظام الملاحة بالمركبات الفضائية؛
- **Business**-critical systems نظم الأعمال الهامة
 - Failure results in high economic losses;
ويؤدي الفشل إلى خسائر اقتصادية عالية؛
Eg. Customer accounting system in a bank;
على سبيل المثال. نظام المحاسبة العملاء في البنك.

اعتماد النظام **System Dependability**

- For critical systems, the most important property is the **dependability** of the system.

وبالنسبة للأنظمة الحرجة، فإن أهم الممتلكات هي إمكانية الاعتماد على النظام.

- It reflects the user's degree of trust & confidence in that system not to 'fail' in normal use.

وهو يعكس درجة المستخدم من الثقة في هذا النظام لا 'فشل' في الاستخدام العادي.

- A system that is useful does not have to be trusted.

ولا ينبغي الوثوق بالنظام المفيد.

➔ أبعاد الإعتمادية تشمل: ➔ The dimensions of dependability include:

1. Availability 2. Reliability 3. Safety & Security

١. التوفر ٢. الموثوقية ٣. السلامة والأمن

Development methods for critical systems

- The development methods for critical system would not be cost-effective for other sys.

إن أساليب تطوير النظام الحرج لن تكون فعالة من حيث التكلفة لغيرها من الأنظمة

- أمثلة على أساليب التنمية Examples of development methods

- **Formal methods** of software development

الطرق الرسمية لتطوير البرمجيات

- **Static analysis** تحليل ثابت

- **External quality assurance**

ضمان الجودة الخارجية

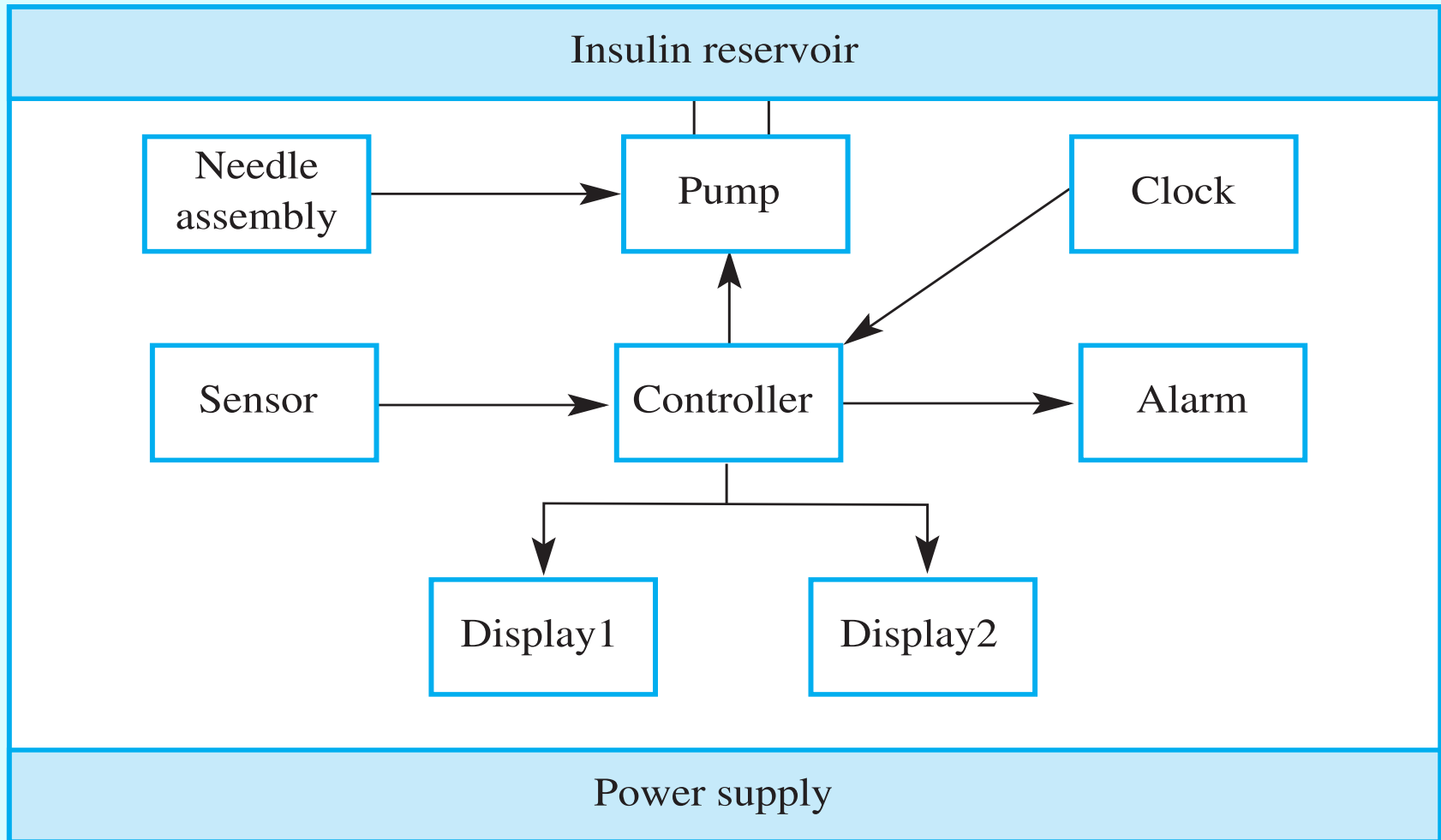
Socio-technical critical systems

- **Hardware failure** فشل في الأجهزة
 - Hardware fails because of design and **manufacturing errors** or because components have reached the **end of their natural life**.
فشل الأجهزة بسبب التصميم وتصنيع الأخطاء أو لأن المكونات قد وصلت إلى نهاية حياتها الطبيعية.
- **Software failure** فشل في البرمجيات
 - Software fails due to **errors in its specification, design or implementation**.
فشل البرنامج بسبب أخطاء في مواصفاته أو تصميمه أو تنفيذه.
- **Operational failure** فشل التشغيل
 - Human operators make **mistakes**. Now perhaps the largest single cause of system failures.
مشغلي الإنسان يخطئون. وربما ربما يكون أكبر سبب وحيد لفشل النظام.

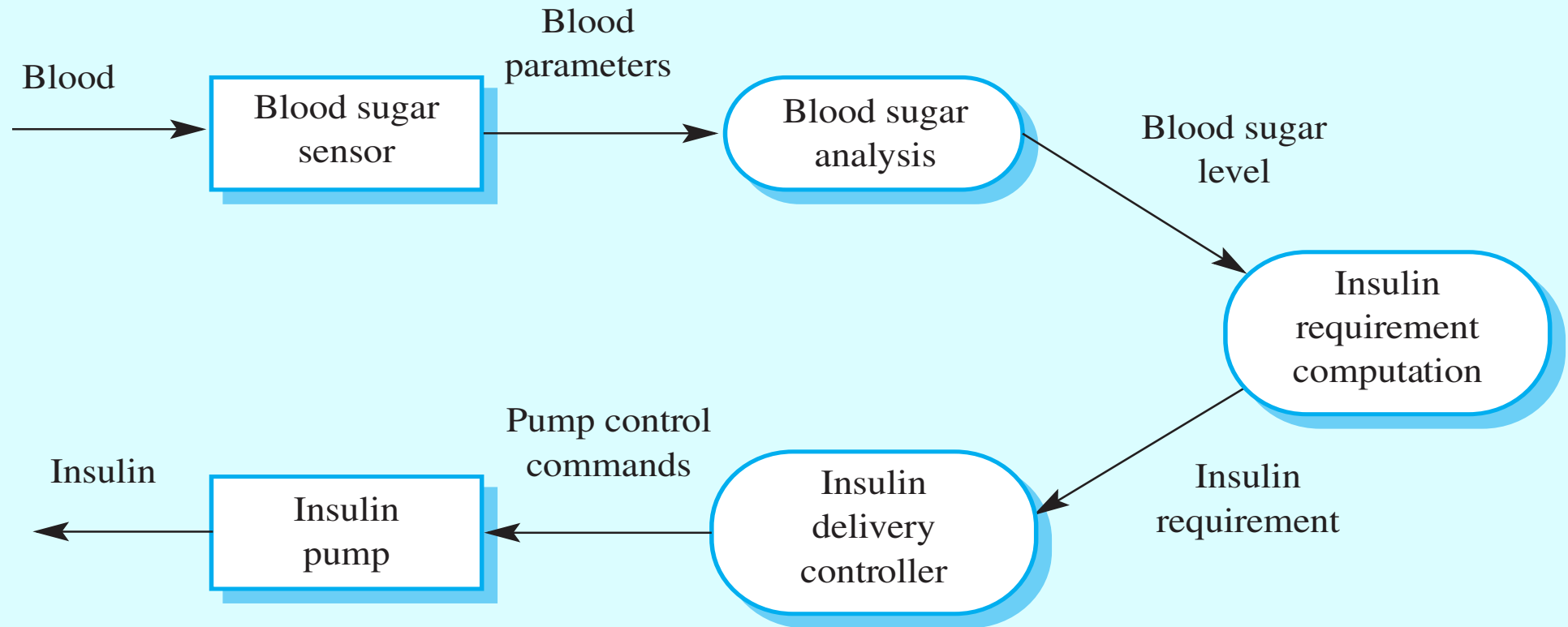
A software-controlled insulin pump

- Used by diabetics to simulate the function of the pancreas which manufactures insulin, an essential hormone that metabolizes blood glucose.
يستخدمها مرضى السكري لمحاكاة وظيفة البنكرياس الذي يقوم بتصنيع الأنسولين، وهو هرمون أساسي يستقلب الجلوكوز في الدم.
- Measures blood glucose (sugar) using a micro-sensor and computes the insulin dose required to metabolize the glucose.
يقيس الجلوكوز في الدم (السكر) باستخدام جهاز استشعار دقيق ويحسب جرعة الأنسولين اللازمة لاستقلاب الجلوكوز.

Insulin pump organisation



Insulin pump data-flow



Dependability requirements

- The system shall be **available** to deliver insulin when required to do so.
يجب أن يكون النظام متوفرا لتقديم الأنسولين عند الحاجة للقيام بذلك.
- The system shall perform **reliability** and deliver the correct amount of insulin to counteract the current level of blood sugar.
يجب على النظام أداء الموثوقية وتسليم كمية الصحيح من الأنسولين لمواجهة المستوى الحالي من السكر في الدم.
- The essential **safety** requirement is that excessive doses of insulin should never be delivered as this is potentially life threatening.
شرط السلامة الأساسية هو أن الجرعات المفرطة من الأنسولين لا ينبغي أبدا أن يتم تسليمها لأن هذا من المحتمل أن تهدد الحياة.

Dependability

- A dependable system is a system that is trusted by its users.

نظام يمكن الاعتماد عليه هو نظام موثوق به من قبل المستخدمين.

- Principal dimensions of dependability are:

الأبعاد الرئيسية للاعتمادية هي:

- **Availability**; *to deliver service when it is required*

توفر. لتقديم الخدمة عندما يكون مطلوباً

- **Reliability**; *always reliable to perform what it is required*

الموثوقية؛ دائماً موثوق بها لأداء ما هو مطلوب

- **Safety**; *to deliver service as specified*

سلامة؛ لتقديم الخدمة كما هو محدد

- **Security**; *to protect itself against accidental and deliberate intrusion.*

الأمان؛ لحماية نفسها ضد عرضي و التطفل المتعمد

What is **CASE**?

(**C**omputer-**A**ided **S**oftware **E**ngineering)

- Software systems that are intended to provide automated support for software process activities.
أنظمة البرمجيات التي تهدف إلى توفير الدعم الآلي لأنشطة عملية البرمجيات.
- CASE systems are often used for method support.
وكثيرا ما تستخدم أنظمة حالة لدعم الطريقة.
- **Upper-CASE**
 - Tools to support the early process activities of **planing, requirements and design**;
أدوات لدعم أنشطة العملية المبكرة من التخطيط والمتطلبات والتصميم؛
- **Lower-CASE**
 - Tools to support later activities such as **programming, debugging and testing**.
أدوات لدعم الأنشطة اللاحقة مثل البرمجة والتصحيح والاختبار.

What are the **Attributes** of Good Software?

- **Maintainability قابليه الصيانه**
 - Software must evolve to meet changing needs (eg. modularity & sys doc.)
يجب أن يتطور البرنامج لتلبية الاحتياجات المتغيرة (على سبيل المثال، نمطية و سبيل الوثيقة).
- **Reliability (Dependability) : الموثوقيه / الاعتماديه**
 - Software must be trustworthy (eg. Data validation & fault tolerance)
يجب أن تكون البرامج جديرة بالثقة (على سبيل المثال التحقق من صحة البيانات والتحمل للخطأ)
- **Efficiency الكفاءه**
 - Software should not make wasteful use of system resources (eg. Data Types, printing diagonal elements in a square-matrix)
ولا ينبغي أن يؤدي البرنامج إلى استخدام موارد النظام بطريقة مهدمة (مثل أنواع البيانات، وطباعة العناصر القطرية في مصفوفة مربعة)
- **Usability (Acceptability) القبول / الاستخدام**
 - Software must be accepted by the users for which it was designed. This means it must be understandable, usable and compatible with other systems (eg. Input & output design guidelines)
يجب قبول البرامج من قبل المستخدمين الذين تم تصميمها. وهذا يعني أنه يجب أن يكون مفهوما وقابلا للاستعمال ومتوافقا مع الأنظمة الأخرى (مثل إرشادات التصميم والإخراج)

What are the **Attributes** of Good Software?

➔ *The software should deliver the required functionality and performance to the user and should be maintainable, dependable and acceptable.*

يجب أن يوفر البرنامج الوظائف والأداء المطلوب للمستخدم ويجب أن يكون [?] قابلاً للصيانة ويمكن الاعتماد عليه ومقبولاً.

What are the key **Challenges** Facing **SE**?

- Heterogeneity & The Legacy Challenge عدم التجانس والتحدي تراث
 - Developing techniques for building software that can cope with heterogeneous platforms and execution environments (eg. Standardization & Adapters)
تطوير تقنيات لبناء البرمجيات التي يمكن التعامل مع منصات غير متجانسة وبيئات التنفيذ (على سبيل المثال، المقاييس والمحاولات)
- Delivery Challenge تحدي التسليم
 - Developing techniques that lead to faster delivery of software (eg. CASE-Tools & modern techniques)
تطوير التقنيات التي تؤدي إلى سرعة تسليم البرمجيات (مثل كيس-تولس والتقنيات الحديثة)
- Trust الثقة
 - Developing techniques that demonstrate that software can be trusted by its users.
تطوير التقنيات التي تثبت أنه يمكن الوثوق بها من قبل مستخدميها.

Professional & Ethical Responsibility

→ **Software engineers** must behave in an honest and ethically responsible way if they are to be respected as professionals.

يجب على مهندسي البرمجيات أن يتصرفوا بطريقة صادقة ومسؤولة أخلاقيا إذا ما أريد احترامهم كمهنيين.

→ Ethical behaviour is more than simply upholding the law.

السلوك الأخلاقي هو أكثر من مجرد التمسك بالقانون.

Issues of Professional Responsibility

- **Confidentiality**

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

يجب على مهندسي البرمجيات أن يحترموا عادة سرية أصحاب العمل أو العملاء بغض النظر عما إذا كان قد تم التوقيع على اتفاقية سرية رسمية أم لا.

- **Competence المهارة**

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

يجب على مهندسي البرمجيات عدم تحريف مستوى كفاءتهم. وينبغي ألا يقبلوا عن علم العمل الذي يقع خارج نطاق اختصاصهم.

Issues of Professional Responsibility (Cont.)

- **حقوق الملكية (حقوق النسخ) Property Rights (copy rights)**

- **Software Engineers** should be aware of local laws governing the use of intellectual property such as patents, copyright, etc. They ensure that the intellectual property of employers/clients is protected.

وينبغي أن يكون مهندسو البرمجيات على علم بالقوانين المحلية التي تحكم استخدام الملكية الفكرية مثل براءات الاختراع وحق المؤلف وما إلى ذلك. ويضمنون حماية الملكية الفكرية لأصحاب العمل / العملاء.

- **إساءة استخدام الكمبيوتر Computer Misuse**

- **Software Engineers** should not use their technical skills to misuse other people's computers. Misuse ranges from trivial (game playing) to extremely serious (spreading of viruses) on an employer's machine.

يجب على مهندسي البرمجيات عدم استخدام مهاراتهم التقنية لإساءة استخدام أجهزة الكمبيوتر الأخرى. ويتراوح سوء الاستخدام من تافهة (لعبة اللعب) إلى خطيرة للغاية (نشر الفيروسات) على الجهاز صاحب العمل.

ACM/IEEE Code of Ethics

- The professional societies in the US have cooperated to produce a code of ethical practice.

وقد تعاونت الجمعيات المهنية في الولايات المتحدة لإنتاج مدونة للممارسات الأخلاقية.

- Members of these organisations sign up to the code of practice when they join.

وينضم أعضاء هذه المنظمات إلى مدونة الممارسات عند انضمامهم إليها.

- 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.

يحتوي القانون على ثمانية مبادئ تتعلق بسلوك وقرارات مهندسي البرمجيات المحترفين، بما في ذلك الممارسين والمربين والمديرين والمشرفين وصانعي السياسات، وكذلك المتدربين وطلاب المهنة.

Code of ethics - principles

- (1). **Software Engineers** should act consistently with the public interest.
(١). يجب على مهندسي البرمجيات العمل بشكل متسق مع المصلحة العامة.
- (2). **Software Engineers** should act in a manner that is in the best interests of their client and employer consistent with the public interest.
(٢). يجب على مهندسي البرمجيات التصرف بطريقة تخدم مصلحة زبائنهم وأصحاب العمل بما يتفق مع المصلحة العامة.
- (3). **Software Engineers** should ensure that their products and related modifications meet the highest professional standards possible.
(٣). يجب على مهندسي البرمجيات التأكد من أن منتجاتهم والتعديلات ذات الصلة تلبى أعلى المعايير المهنية الممكنة.

Code of ethics – principles (Cont.)

(4). **Software Engineers** should maintain integrity and independence in their professional judgment.

(٤). يجب على مهندسي البرمجيات الحفاظ على النزاهة والاستقلال في حكمهم المهني.

(5). **Software Engineering Managers & Leaders** shall subscribe to and promote an ethical approach to the management of software development and maintenance.

(٥). يجب على مدراء هندسة البرمجيات والقادة الاشتراك في وتعزيز نهج أخلاقي لإدارة تطوير البرمجيات وصيانتها.

(6). **Software Engineers** should advance the integrity and reputation of the profession consistent with the public interest.

(٦). يجب على مهندسي البرمجيات تعزيز سلامة وسمعة المهنة بما يتفق مع المصلحة العامة.

Code of ethics – principles (Cont.)

(7). **Software Engineers** should be fair to and supportive of their colleagues.

(٧). يجب أن يكون مهندسو البرمجيات عادلة وداعمة لزملائهم.

(8). **Software Engineers** should participate in lifelong learning regarding the practice of their profession.

(٨). يجب على مهندسي البرمجيات المشاركة في التعلم مدى الحياة فيما يتعلق بممارسة مهنتهم.

Key Points (*Summary*)

- Software engineering is an engineering discipline that is concerned with all aspects of software production.
هندسة البرمجيات هي الانضباط الهندسي الذي يهتم بجميع جوانب إنتاج البرمجيات.
- Software products consist of developed programs and associated documentation. Essential product attributes are maintainability, dependability, efficiency and usability.
تتكون منتجات البرمجيات من البرامج المتقدمة والوثائق المرتبطة بها. سمات المنتج الأساسية هي الصيانة، والاعتمادية، والكفاءة وسهولة الاستخدام.
- The software process consists of activities that are involved in developing software products. Basic activities are software specification, development, validation and evolution.
تتكون عملية البرمجيات من الأنشطة التي تشارك في تطوير منتجات البرمجيات. الأنشطة الأساسية هي مواصفات البرمجيات، والتنمية، والتحقق من الصحة والتطور.
- Methods are organised ways of producing software. They include suggestions for the process to be followed, the notations to be used, rules governing the system descriptions which are produced and design guidelines.
الطرق هي طرق منظمة لإنتاج البرمجيات. وهي تتضمن اقتراحات بشأن العملية التي ينبغي اتباعها، والرموز التي ستستخدم، والقواعد التي تنظم توصيفات النظام التي يتم إعدادها وتصميم المبادئ التوجيهية.

Key Points (Summary Cont.)

- CASE tools are software systems which are designed to support routine activities in the software process such as editing design diagrams, checking diagram consistency and keeping track of program tests which have been run.
أدوات هي أنظمة البرمجيات التي تم تصميمها لدعم الأنشطة الروتينية في عملية البرمجيات مثل تحرير الرسوم البيانية تصميم والتحقق من اتساق الرسم البياني وتتبع الاختبارات البرنامج التي تم تشغيلها.
- Software engineers have responsibilities to the engineering profession and society. They should not simply be concerned with technical issues.
مهندسي البرمجيات لديهم مسؤوليات لمهنة الهندسة والمجتمع. ولا ينبغي أن يكونوا مهتمين بالقضايا التقنية.
- Professional societies publish codes of conduct which set out the standards of behaviour expected of their members.
وتنشر الجمعيات المهنية مدونات لقواعد السلوك تحدد معايير السلوك المتوقعة من أعضائها.