Instructor Romasha Khurshid

Welcome To My Software Engineering Class

Tentative Marks Distribution

- Assignment+ Class Participation --------- 10 Marks
- Sessional---------- 15+15 Marks
- Final---------- 50 Marks
- Project---------- 10 Marks

Course Material:

Books:

- Software Engineering by Ian Sommerville (10 edition)
- Software Engineering Practitioner's Approach by Roger S. Pressman (Seventh Edition)

Reading Material:

- Articles Readings.

Course Learning Outcomes:

- Have a sound understanding of the fundamental concepts of the software engineering paradigm.
- Understand and apply the different common practices used in software industry for the analysis, design and production of software.
- Analyze, design and implement practical systems of up to average complexity within a team.
- Familiarity with concepts of software quality and testing.

Warm-Up Activity:

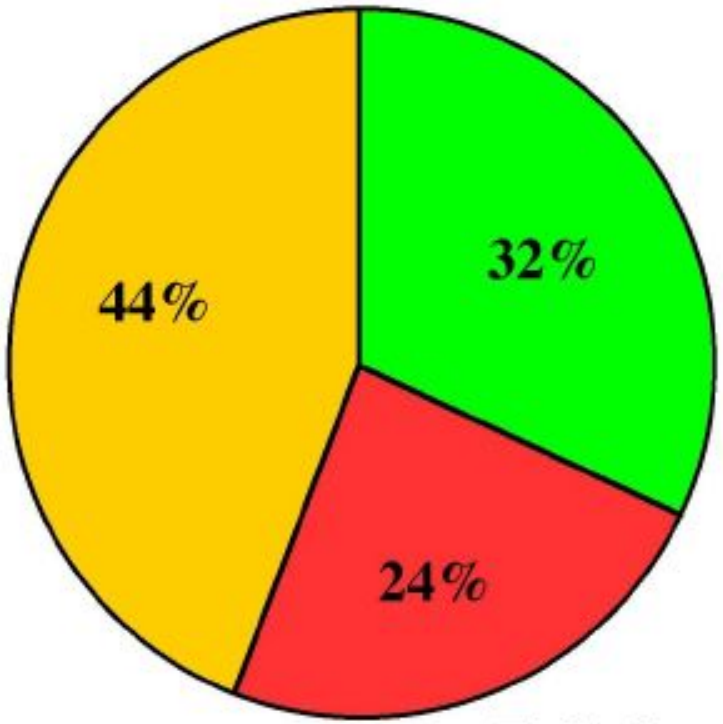
What are you expecting from this course?

Briefly describe your understanding in SDA?

Briefly Describe what do you know about Software engineering?

The Software Crisis

Challenged = late, overbudget, and/or withless than the required features and functions

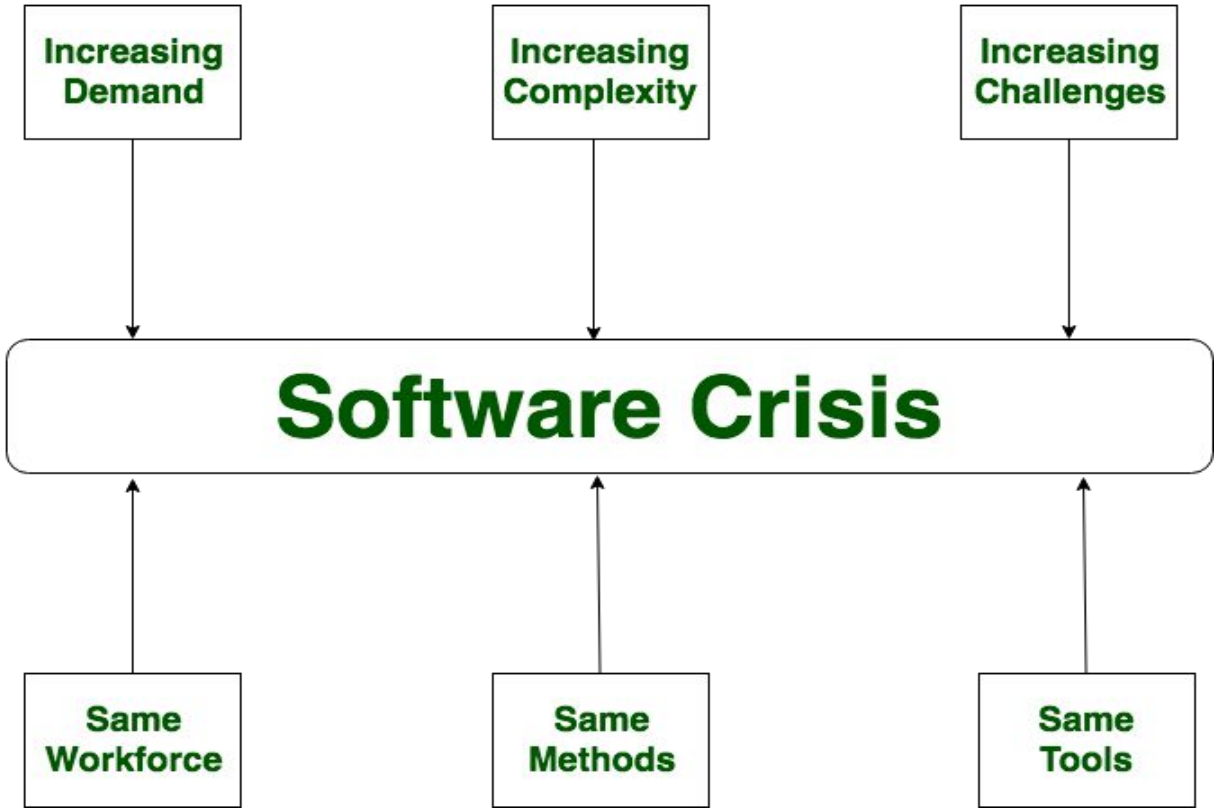


Successful = delivered on time, on budget, with required features and functions

Failed = cancelled prior to completion or delivered and never used

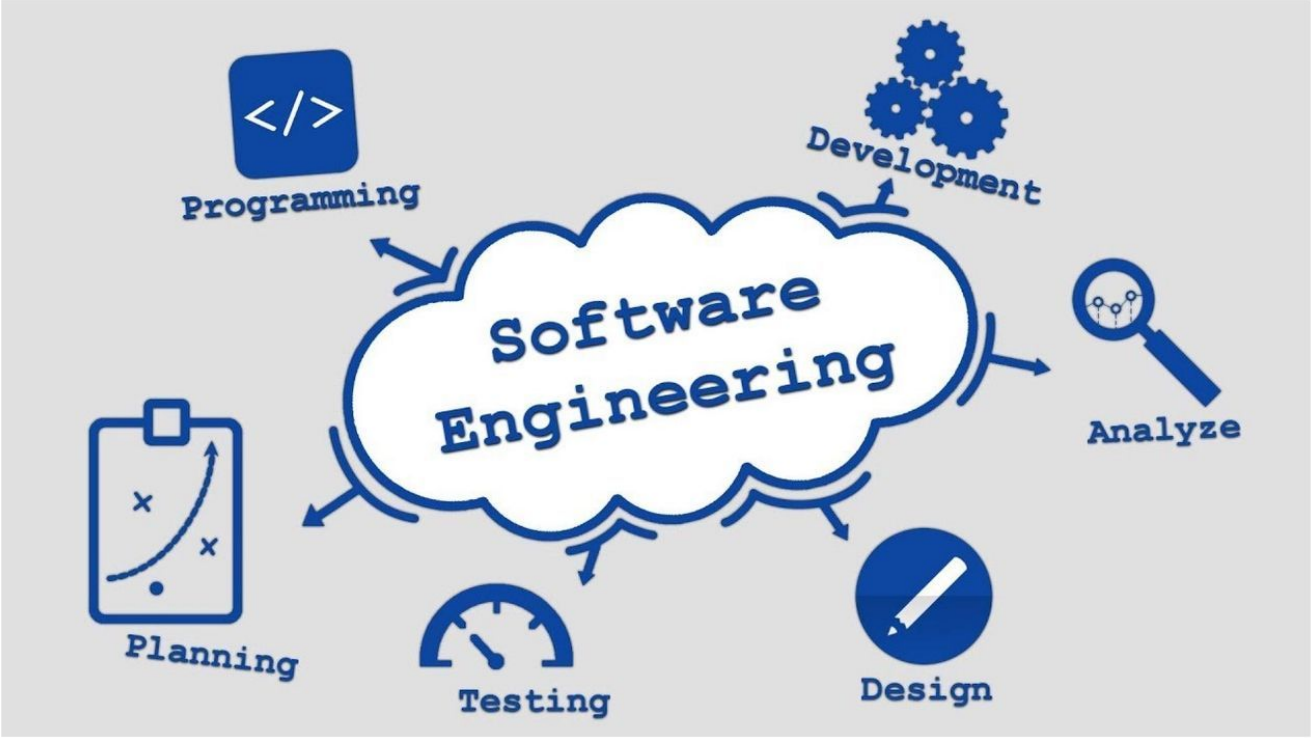
* Source: Standish Group, British Computer Society

- software industry’s inability to provide customers with high quality products on schedule.
- Projects ran over-budget.
- Damage to property even to life.



What is 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.



Key Phrases:

- 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.

Software project failure

- *Increasing system complexity*
  - As new software engineering techniques help us to build larger, more complex systems, the demands change. Systems must be built and delivered more quickly; larger, even more complex systems are required; systems must have new capabilities that were previously thought to be impossible.
- *Failure to use software engineering methods*
  - It is easy to write computer programs without using software engineering methods and techniques. Many companies have drifted into software development as their products and services have evolved. They do not use software engineering methods in their everyday work. Consequently, their software is often more expensive and less reliable than it should be.

Professional software development:

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 aspects 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.

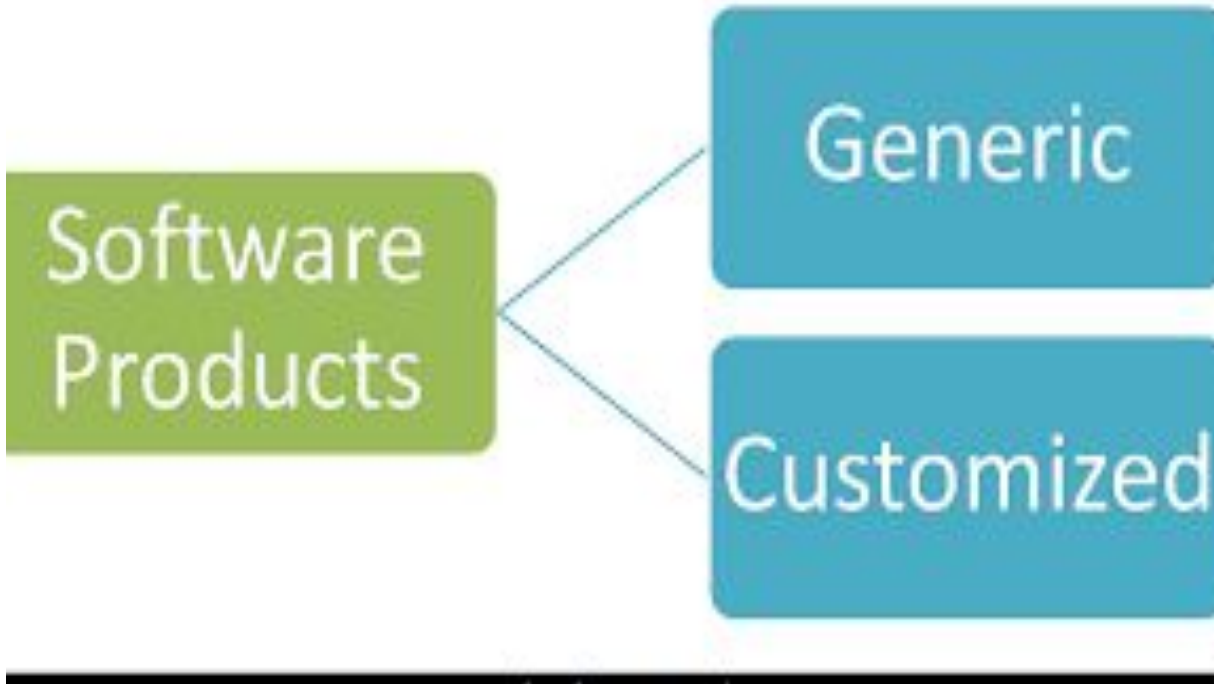
Software products

Generic products:

Stand-alone systems that are marketed and sold to any customer who wishes to buy them.

Customized products:

Software that is commissioned by a specific customer to meet their own needs.

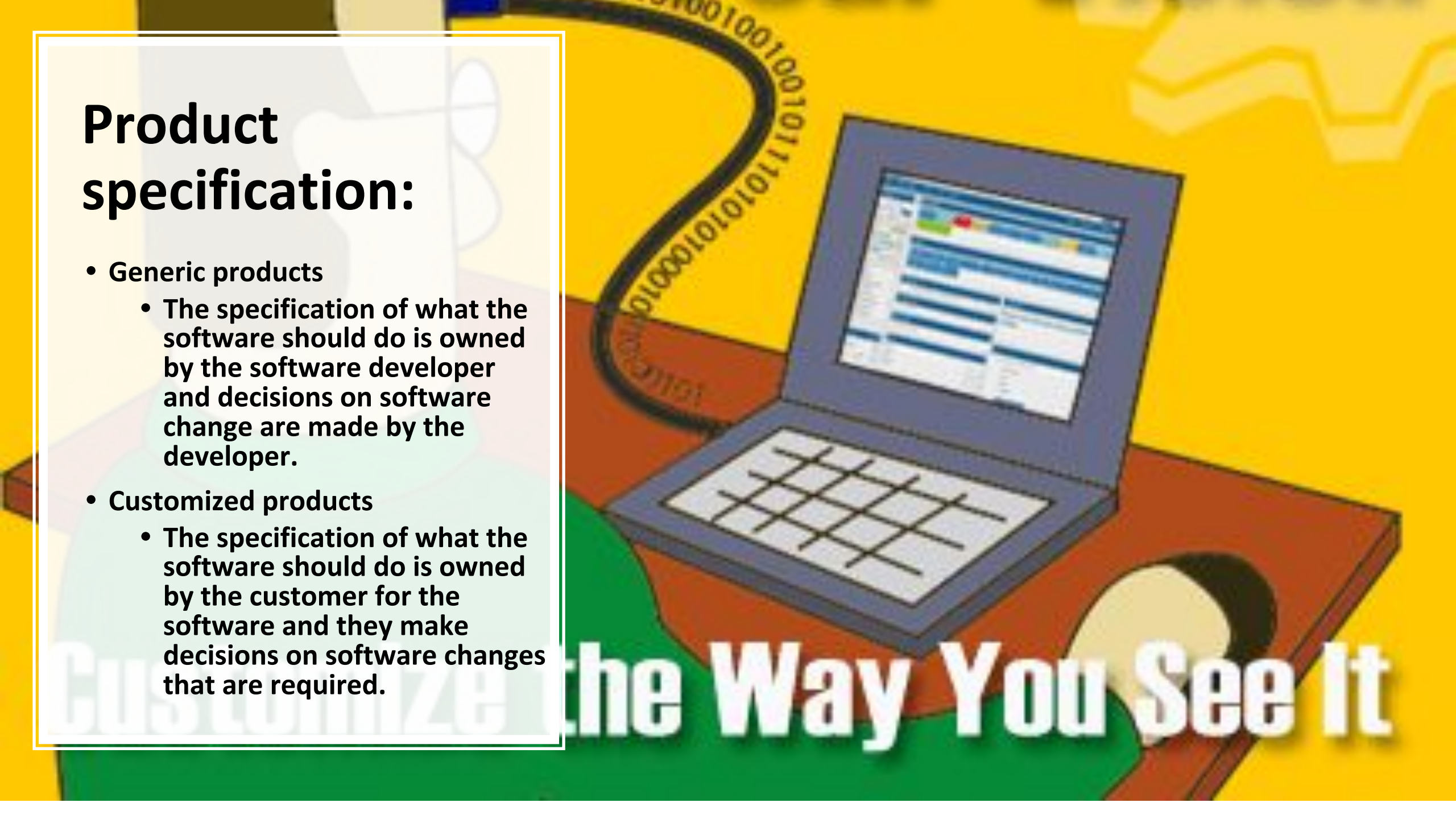


Examples of Generic Software

- Word processor – e.g. Word, WordPerfect
- Spreadsheet – e.g. Excel, Lotus 1-2-3
- Database applications – e.g. Access, Approach
- Presentation – e.g. PowerPoint, Impress
- Electronic Mail & Diary – e.g. Outlook, Notes
- Web-browser – e.g. Firefox, Opera
- Application generator – Access, Paradox

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.



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.

Importance of software engineering

- 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.
- For most types of system, the majority of costs are the costs of changing the software after it has gone into use.

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.

Software process activities:

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.

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.

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.

Continue…

- **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.

Continue…

- **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.

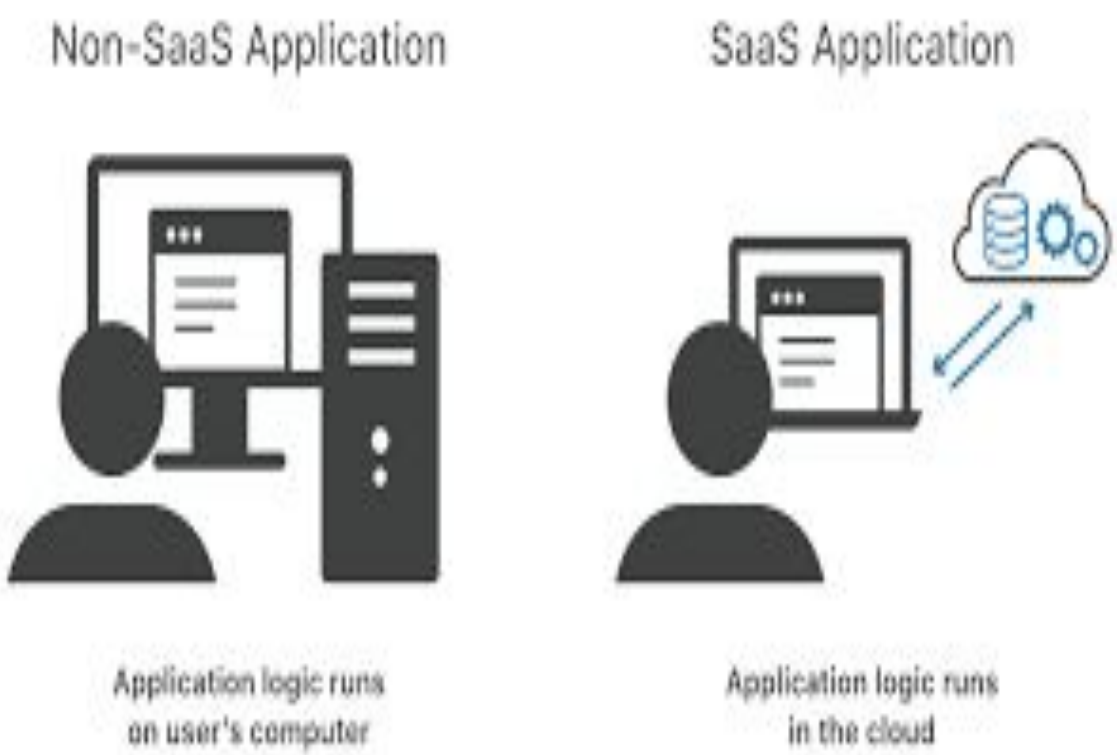
- **Systems of systems:**

  These are systems that are composed of a number of other software systems.

Internet software engineering:

- The Web is now a platform for running application and organizations are increasingly developing web-based systems rather than local systems.
- 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.
- The fundamental ideas of software engineering apply to web-based software in the same way that they apply to other types of software system



Software engineering ethics

Software engineers must be honest and ethically responsible about his job.

Read Ian Somerville's chap 01 for further details

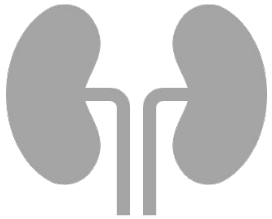
Case Studies

1. Insulin Pump Control System

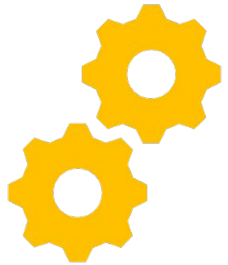
- Collects data from a blood sugar sensor and calculates the amount of insulin required to be injected.
- Calculation based on the rate of change of blood sugar levels.
- Sends signals to a micro-pump to deliver the correct dose of insulin.
- **Safety-critical system as** low blood sugars can lead to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

Essential high-level requirements

The system shall be available to deliver insulin when required.

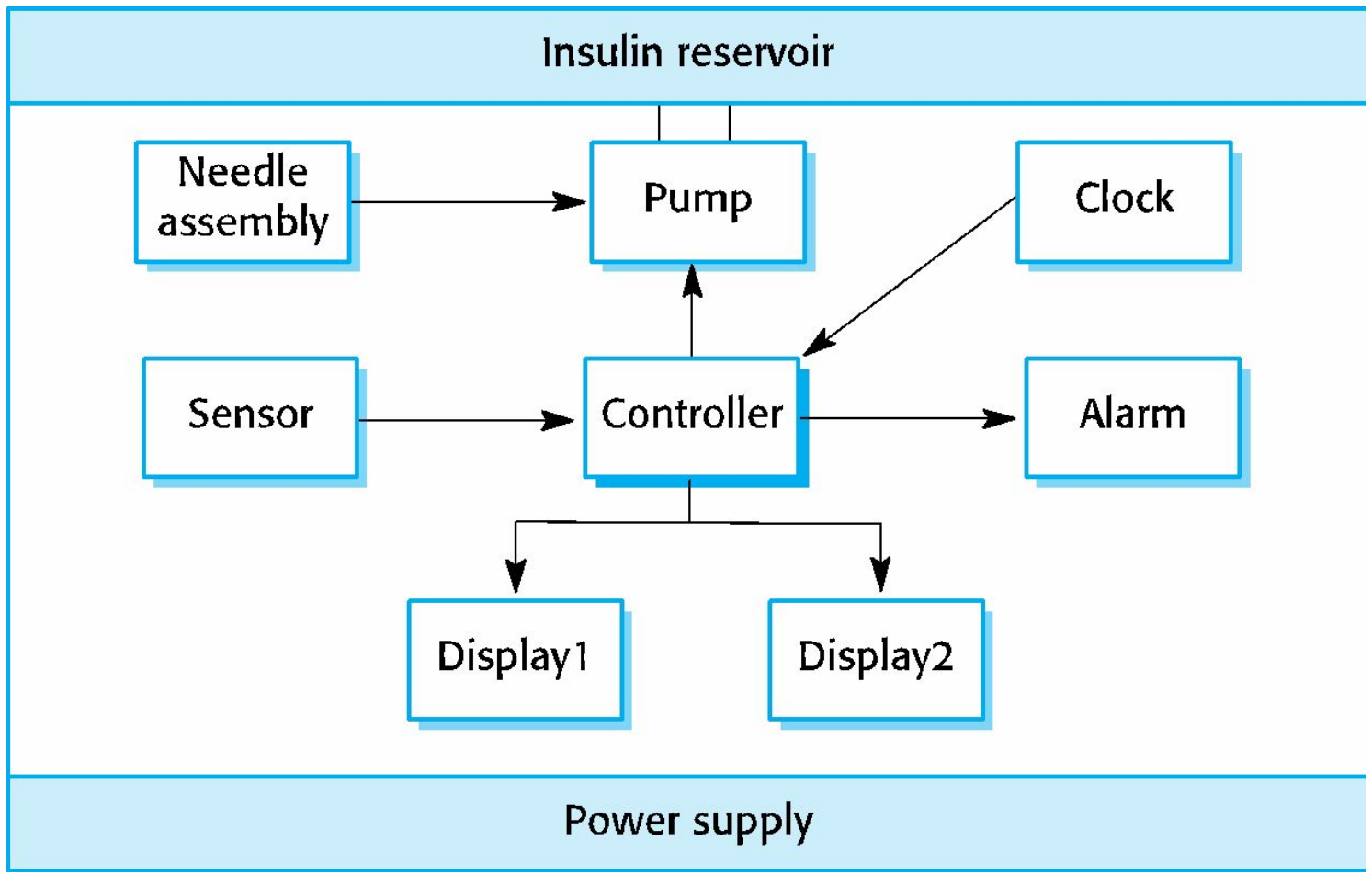


The system shall perform reliably and deliver the correct amount of insulin to counteract the current level of blood sugar.

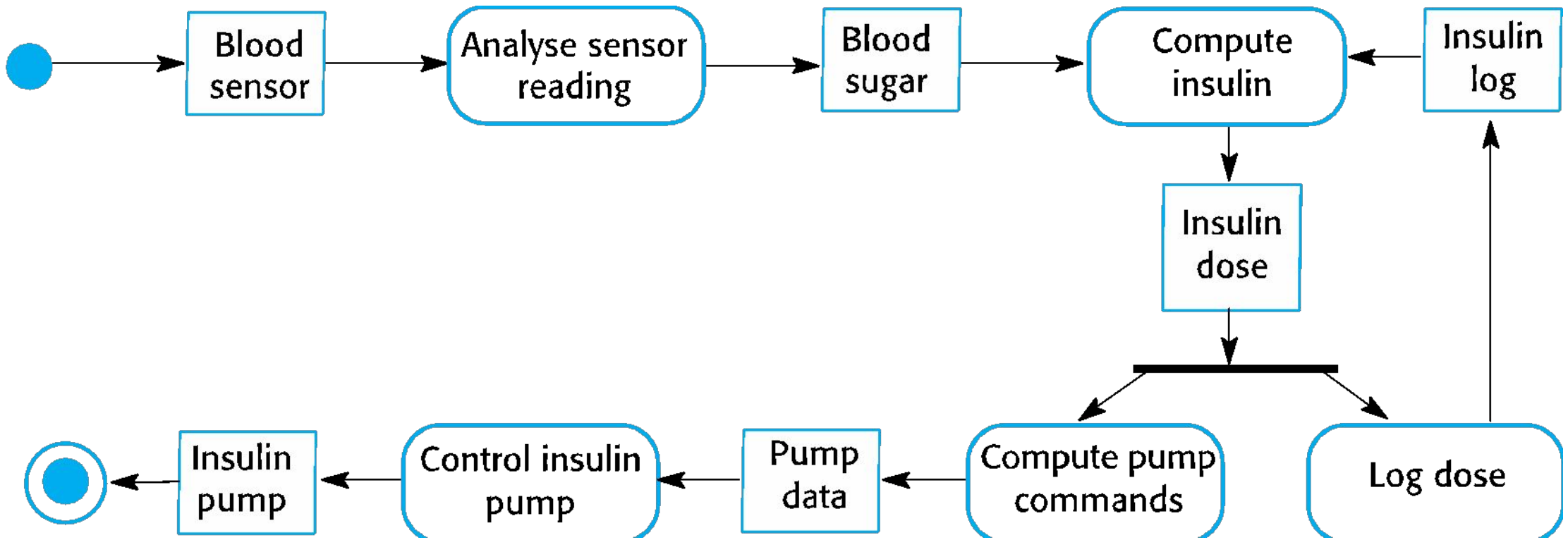


The system must therefore be designed and implemented to ensure that the system always meets these requirements.

Insulin pump hardware architecture



Activity model of the insulin pump



Issues with embedded systems

- Physical size
- Responsiveness
- Power Management

Mentcare

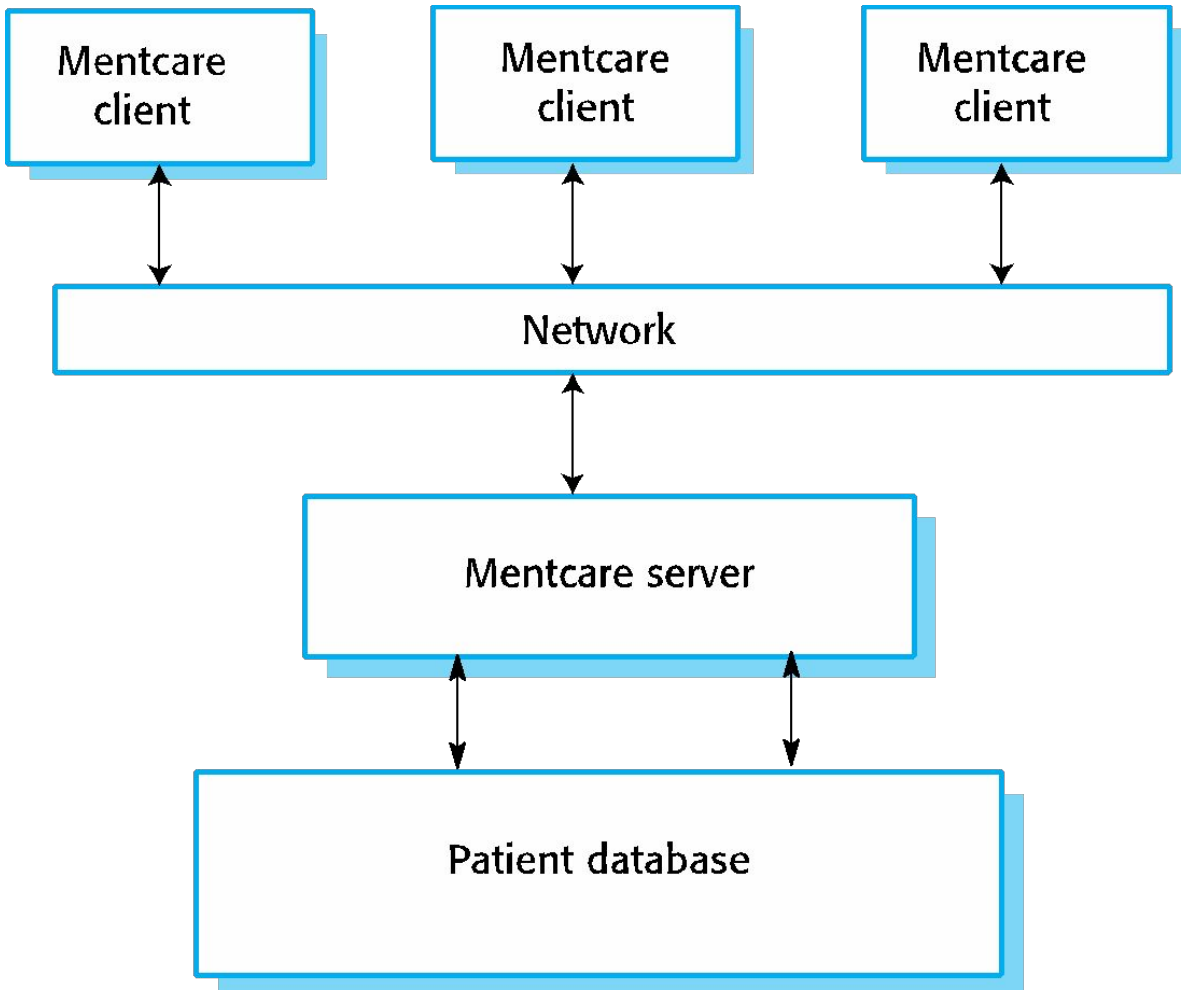
A patient information system for mental health care:

- Mentcare is an information system that is intended for use in clinics.
- Maintains information about patients suffering from mental health
- Maintains treatments records that they have received.
- Centralized database of patient information but also been designed to run on a PC.

Critical Requirements:

- **Safety and Privacy are the critical requirement of the given system.**
- The system must be available when needed otherwise **safety** may be compromised and it may be impossible to prescribe the correct medication to patients.
- **Privacy** is easiest to maintain with single copy of record.
- However, to ensure availability, multiple copies of the data should be maintained.
- I discuss the trade-offs between these requirements in later chapters

The organization of the Mentcare system



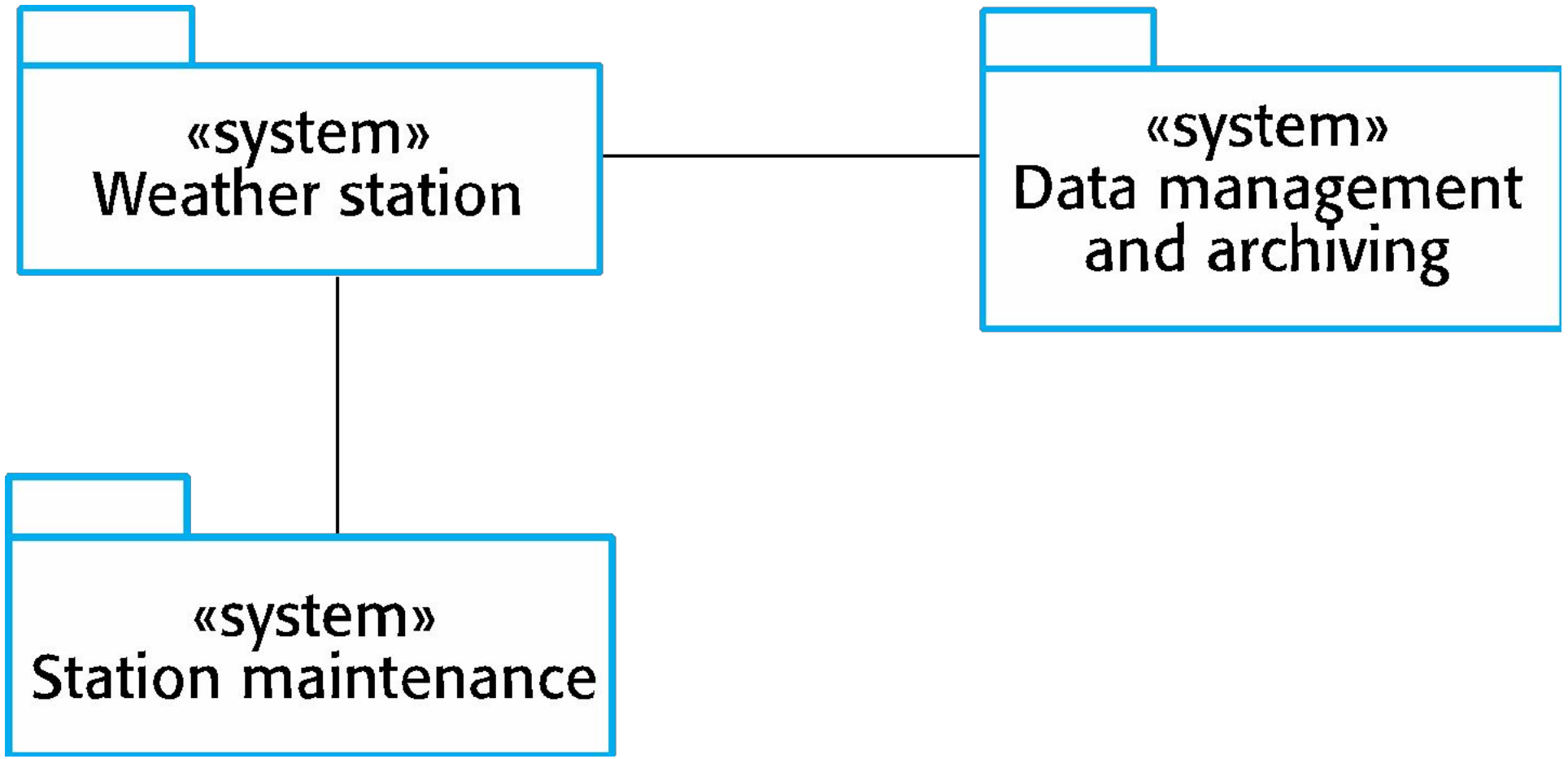
Key features of the Mentcare system

- Individual care management
- Patient monitoring
- Administrative reporting

Wilderness weather station:

- Weather stations collect data from a set of instruments that measure temperature and pressure, sunshine, rainfall, wind speed and wind direction
- 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 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.

A Fifteen minutes activity

Go out of the Class
and visit faculties of your campus like The Director, HOD, Professors ,senior lecturer, lecturer, instructors etc.

Ask them the role or the importance of Software Engineering, in software projects and for software industry as well?