



Chapter 1- Introduction

Topics covered



✧ Professional software development

- What is meant by software engineering.

✧ Software engineering ethics

- ethical issues that affect software engineering.

✧ Case studies

Software engineering



- ✧ What
 - ✧ Theories, methods and tools for software development
- ✧ Why
 - ✧ Automation
 - ✧ Cost-effective (cheaper on the long run)
- ✧ How
 - ✧ Following SDLC

Cost



- ✧ What do you think about software cost
 - ✧ Compared to hardware
 - ✧ Development
 - ✧ Maintenance

Failure



- ✧ Why software projects fail?
 - ✧ Complexity
 - ✧ Poor communication
 - ✧ Lack of QC, QA
 - ✧ Tight architecture
 - ✧ Haphazard development (no SDLC)



Professional software development

FAQs



Question

What is software?

What are the attributes of good software?

What is software engineering?

What are the fundamental software engineering activities?

What is the difference between software engineering and computer science?

What is the difference between software engineering and system engineering?

Answer

Computer programs and associated documentation.

- Deliver required functionalities
- Maintainable
- Performance
- Usable

Software engineering is an engineering discipline that is concerned with all aspects of software production.

- Specification
- Development
- Validation
- Evolution

Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.

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.

FAQs



Question

What are the key challenges facing software engineering?

What are the costs of software engineering?

What are the best software engineering techniques and methods?

What differences has the web made to software engineering?

Answer

increasing diversity, reduced delivery times, trustworthy.

60% development,
40% testing.

Evolution: exceed development costs.

No one size fits all

- Games: Spiral, series of prototypes
- Safety critical systems: Water fall.
- Web portal: Agile

- Distributed services
- Web based portals
- Cloud platform

Essential attributes of good software



Attribute	Description
Maintainability	Evolvable
Dependability and security	<ul style="list-style-type: none">• Circuit breaker• Access control• Error containment
Efficiency	<ul style="list-style-type: none">• CPU utilization• Memory utilization <ol style="list-style-type: none">1. Responsive2. Resilient3. Elastic
Acceptability	Acceptable and understandable by users

Definition



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.

Definition



✧ Engineering discipline

- GIVEN theories & methods
- SOLVE technical & project problems
- CONSTRAINT organization & financial constraints

✧ All aspects of software production

- technical process of development.
project management
- development of tools, methods etc. to support software production.
- COTS

Activities



- ✧ Specification
 - customers and engineers define the software requirements
 - with conditions and constraints on operation
- ✧ Development
 - design and programming.
- ✧ Validation,
 - QC.
 - QA
- ✧ Evolution
 - Modification
 - New customer requests (CR)



No **universal** software engineering methods or techniques that may be used for all kinds of software. However, there are factors that affect different types of software

Issues that affect software



✧ Heterogeneity

- Channel: Web, Desktop, Mobile
- Backend: Java, PHP, NodeJS
- Frontend: React, Angular, VueJS

✧ Business and social change

- Social Logins
- Contactless payment
- eWallet
- Loyalty programs

Issues that affect software



✧ Security and trust

- Security.
- Trust

✧ Scale

- Embedded systems
- Wearable devices
- Web-based
- Smart phones
- Cloud-based

Software engineering fundamentals



- ☐ Adopt development process
- ☐ Dependability and performance
- ☐ Solid software specs and requirements
- ☐ Reuse existing software as possible
- ☐ Incremental delivery where possible



Software engineering ethics



Software engineering ethics

✧ Confidentiality

- respect the confidentiality of your employers or clients

✧ Competence

- Maintain your level of competence
- Don't accept a work outside your competences

✧ Computer misuse

- Back-door development
- Game playing on employer machine

✧ Intellectual property right

- Knowledge of governmental laws
- Copyrights

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

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:

The ACM/IEEE Code of Ethics



1. PUBLIC

2. CLIENT AND EMPLOYER

3. PRODUCT

4. JUDGMENT

5. MANAGEMENT

6. PROFESSION

7. COLLEAGUES

8. SELF

Ethical principles



1. PUBLIC –

1. Accept full responsibility for their own work
2. Approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality .
3. Be fair and avoid deception in all statements

2. CLIENT AND EMPLOYER –

1. Not knowingly use software that is obtained or retained illegally/unethically
2. Ensure that any document upon which they rely has been approved
3. Keep private any confidential information gained in their professional work

3. PRODUCT –

1. Strive for high quality, acceptable cost and a reasonable schedule, ensuring significant tradeoffs are clear to and accepted by the employer and the client
2. Ensure that they are qualified for any project on which they work
3. Strive to fully understand the specifications for software on which they work

Ethical principles



4. JUDGMENT–

1. Maintain professional objectivity with respect to any software or related documents they are asked to evaluate
2. Refuse to participate, in a private, governmental or professional body, in which they, their employers or their clients have potential conflicts of interest

5. MANAGEMENT –

1. Attract potential software engineers only by full and accurate description of the conditions of employment
2. Ensure good management for any project on which they work
3. Ensure realistic quantitative estimates of cost, scheduling, personnel, quality and outcomes

6. PROFESSION –

1. Help develop an organizational environment favorable
2. Promote and influence public knowledge of software engineering



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

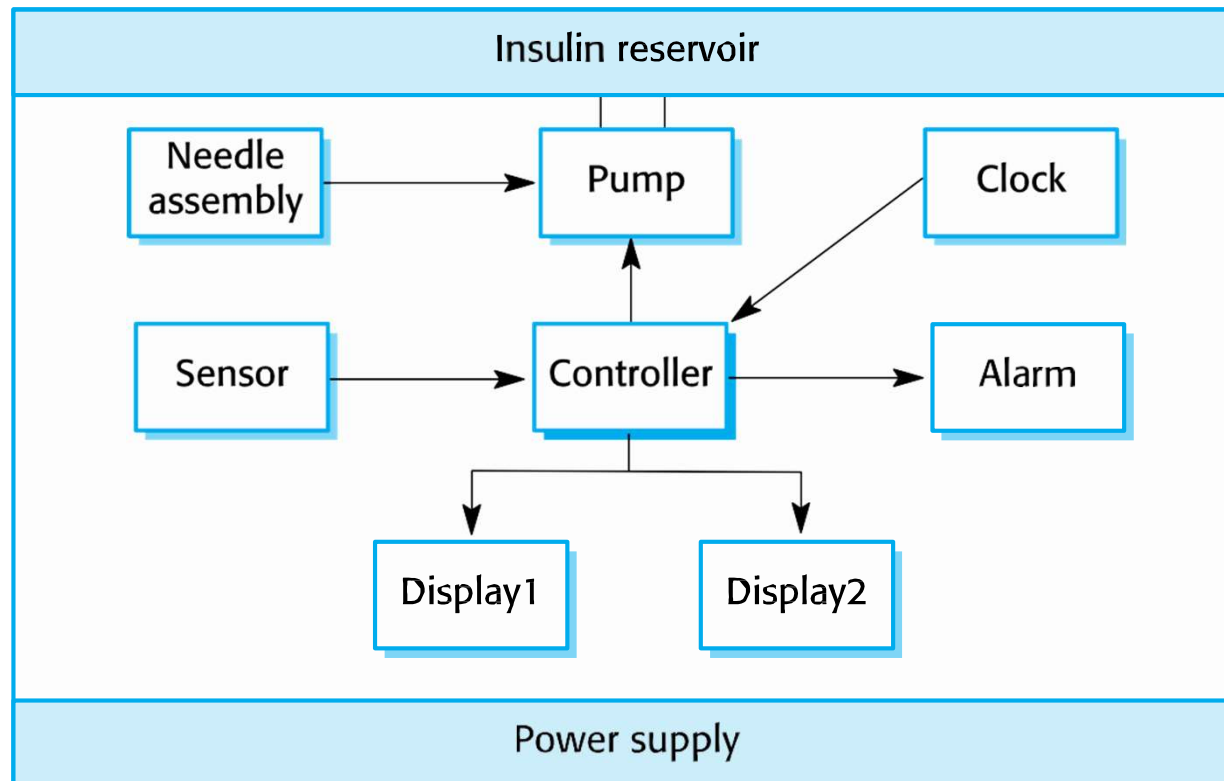
- A system to support learning in schools

Insulin pump control system

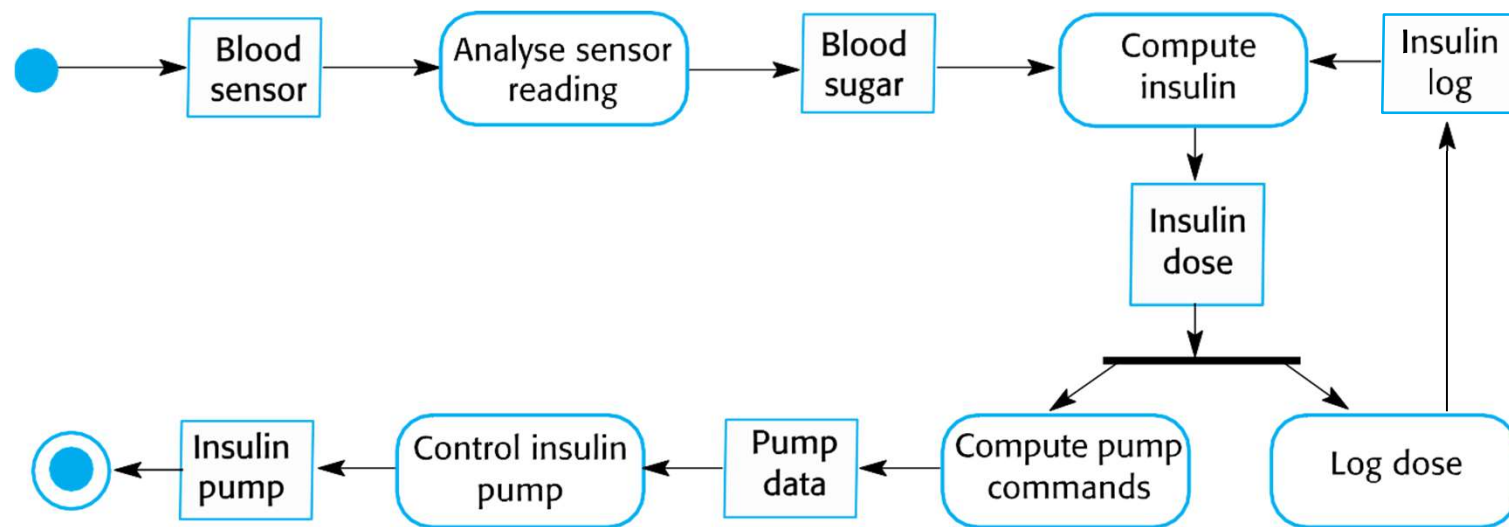


- ✧ Collects data from a blood sugar sensor and calculates the amount of insulin required to be injected.
- ✧ Calculation based on the rate of change of blood sugar levels.
- ✧ Sends signals to a micro-pump to deliver the correct dose of insulin.
- ✧ Safety-critical system as low blood sugars can lead 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



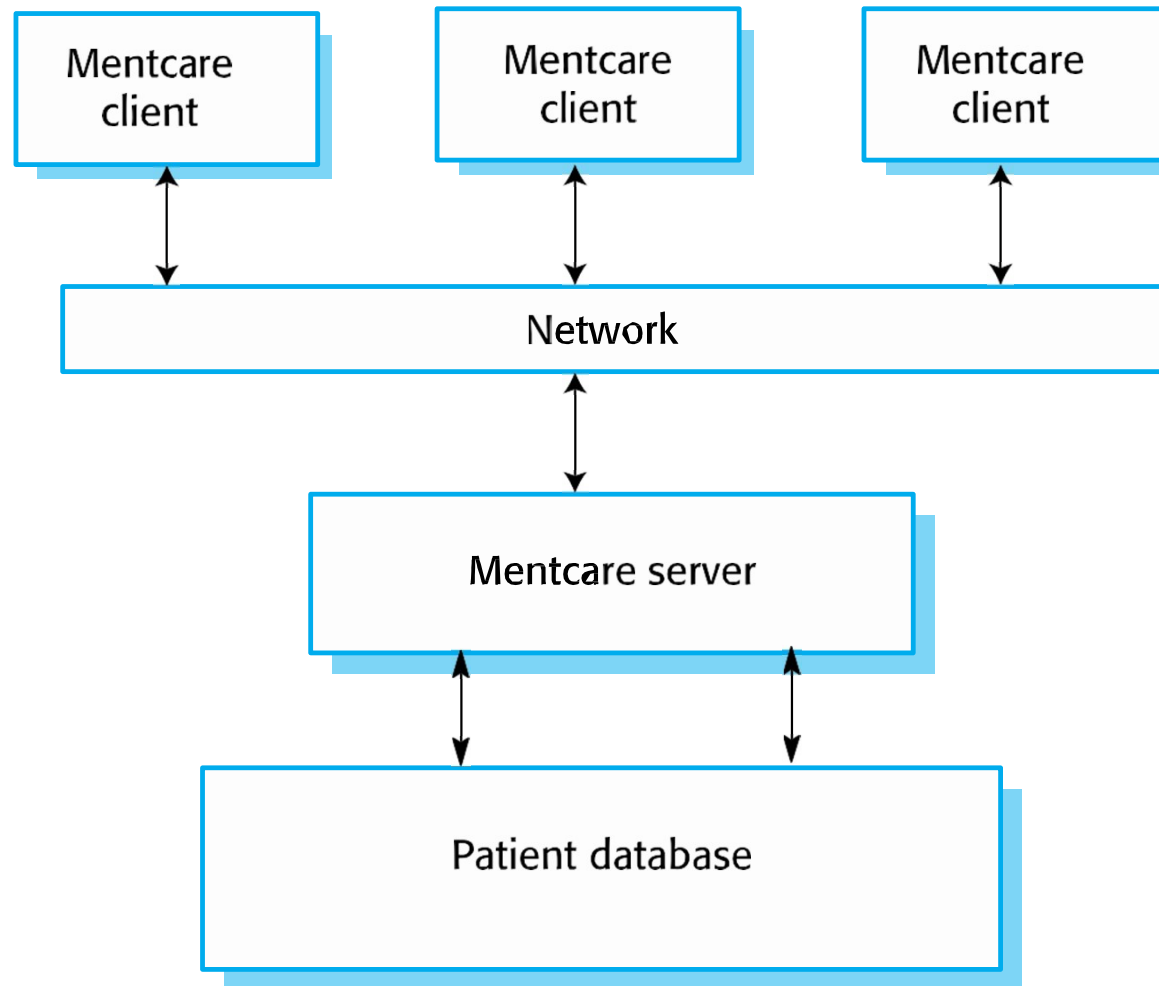
- ✧ Mentcare is an information system that is intended for use in clinics.
- ✧ It makes use of a centralized database of patient information but has also been designed to run on a PC, so that it may be accessed and used from sites that do not have secure network connectivity.
- ✧ When the local systems 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

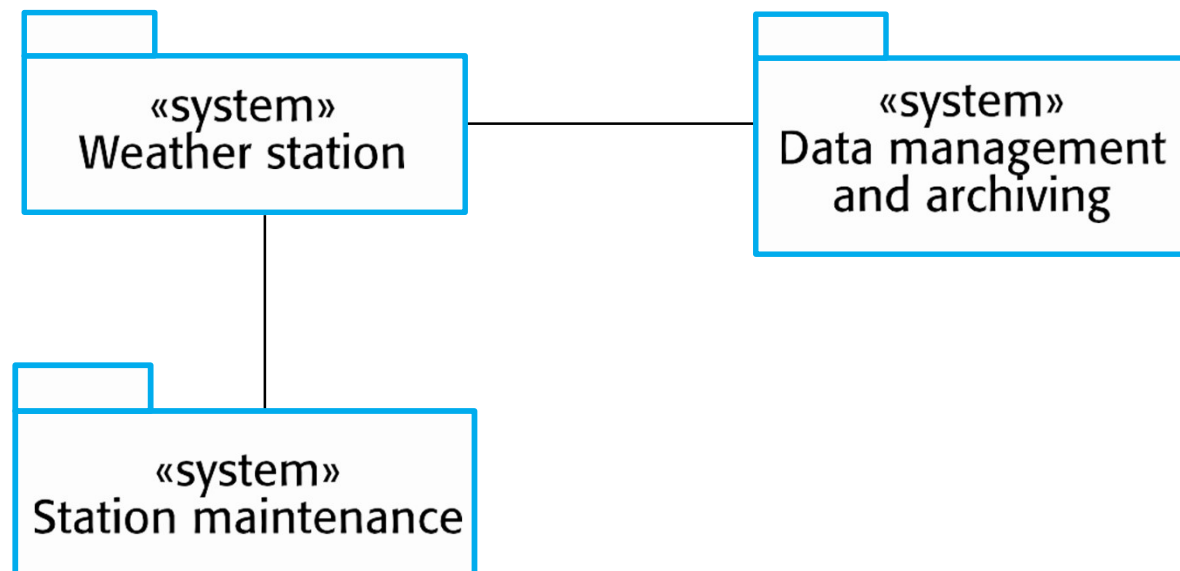
- Some mental illnesses cause patients to become suicidal or a danger to other people. Wherever possible, the system should warn medical staff about potentially suicidal or dangerous patients.
- The system must be available when needed otherwise safety may be compromised and it may be impossible to prescribe 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.



- ✧ *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



Browser-based user interface

iLearn app

Configuration services

Group
management

Application
management

Identity
management

Application services

Email Messaging Video conferencing Newspaper archive
Word processing Simulation Video storage Resource finder
Spreadsheet Virtual learning environment History archive

Utility services

Authentication
User storage

Logging and monitoring
Application storage

Interfacing
Search

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



- Group of 5

- Discuss what have you learnt in this chapter
 - Concepts
 - New facts

Questions



1. Based on your understanding, what are the reasons for software failure?
2. Software engineering is an engineering discipline that is concerned with all aspects of software production, discuss!
3. What are the fundamental activities of software engineering?
4. **no ONE method fits all**, discuss the concept in the context of software engineering

Questions



5. Based on your understanding, what are the main factors that affect the software cost?
6. What is the major difference between generic & custom software product types?
7. Why is maintainability an essential attribute for good software products?
8. Discuss and mention the pillars of reactive software
9. Based on your understanding, discuss the activities of software process (no more than one statement per activity)

Questions



10. Based on your understanding, mention some issues that might affect software
11. Consider you have been promoted to a team leader in your company, what are the fundamental factors that you would adopt for better software development?
12. Confidentiality and computer misuse are two factors in software engineering ethics, discuss!
13. Briefly discuss why it is usually cheaper in the long run to use software engineering methods and techniques for software systems

References



1. <https://iansommerville.com/software-engineering-book/static/about/>
2. <https://codingsans.com/state-of-software-development-2019>
3. <https://hired.com/page/state-of-software-engineers/#about-hired>