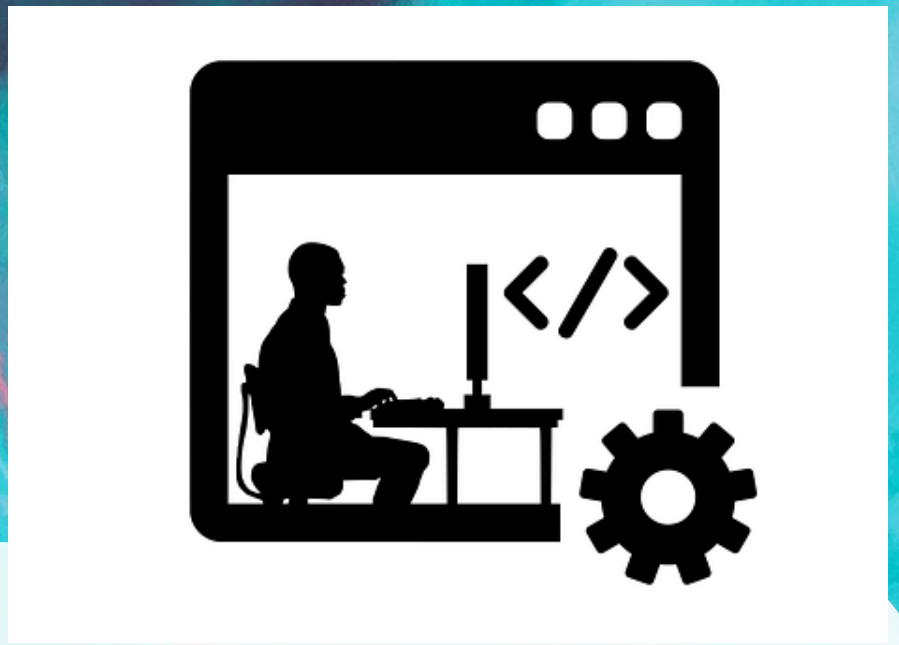


CSE241/CMM341
Foundations of Software Engineering

Chapter 1
Introduction to Software Engineering

SCHOOL OF COMPUTER SCIENCES
UNIVERSITI SAINS MALAYSIA

CSE241/CMM341 Course Content

Foundations of Software Engineering

1. **Introduction to Software Engineering**
2. Software Processes
3. Agile Software Development
4. Requirements Engineering
5. System Modeling
6. Software Design Strategies and Methods
7. Architectural Design & Implementation
8. Software Testing
9. Software Evolution
10. Project Management
11. Project planning
12. Quality Management
13. Configuration Management

Chapter 1

Introduction to Software Engineering

Topics covered

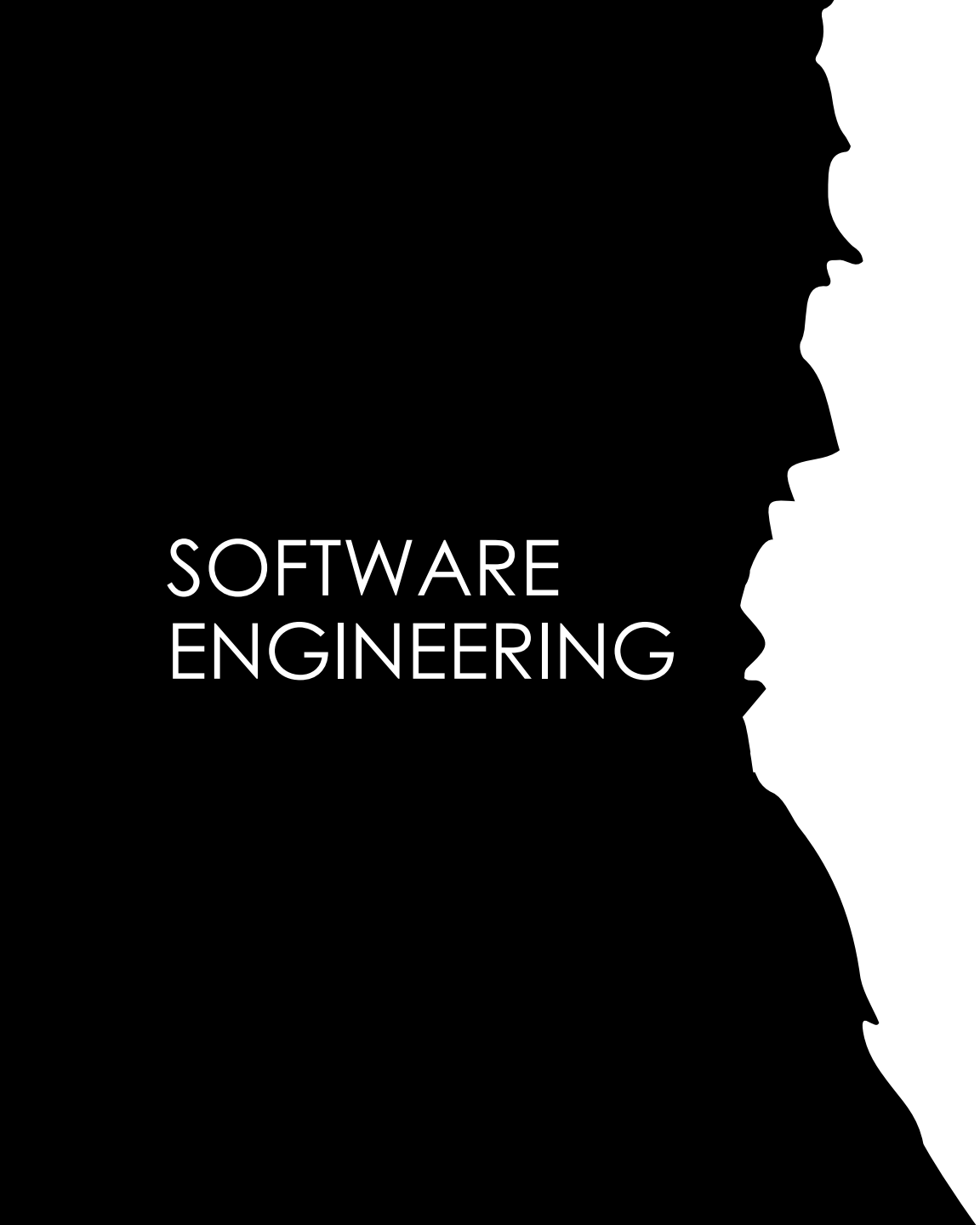
- Professional software development
  - What is meant by software engineering.

- Software engineering ethics
  - A brief introduction to ethical issues that affect software engineering.

- Case studies
  - An introduction to three examples that are used in later chapters in the book.

Image sources:

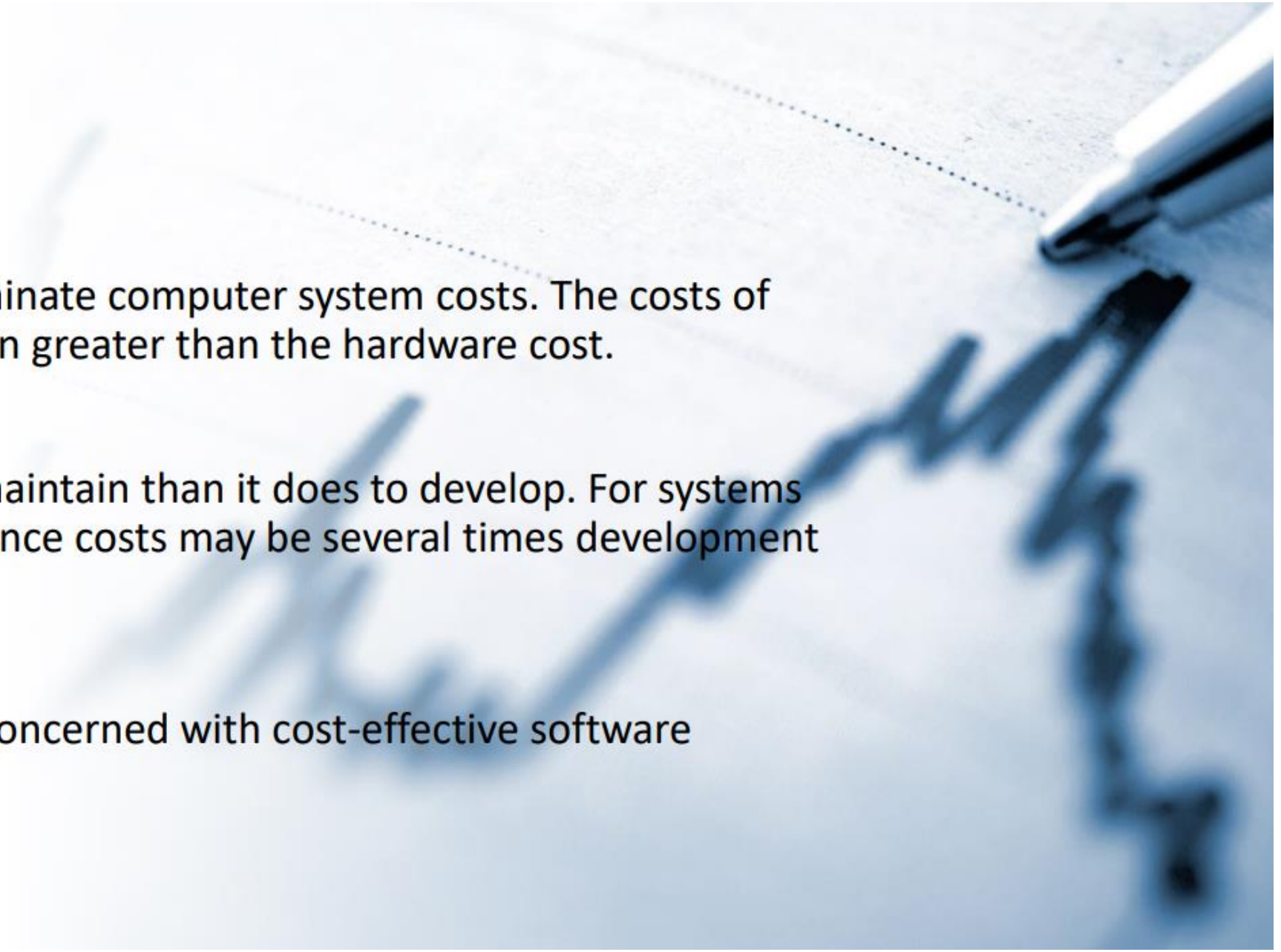
https://sevenpillarsinstitute.org/case-studies/list-all-case-studies/
https://bestweb.com.my/3-mistakes-not-to-make-when-choosing-a-software-development-compan
https://growthanimals.com/5-growing-ethical-issues-in-online-marketing-you-need-to-be-aware-of/



SOFTWARE ENGINEERING

- The economies of ALL developed nations are dependent on software.
- More and more systems are software controlled
- Software engineering is concerned with theories, methods and tools for professional software development.
- Software systems – abstract and intangible

SOFTWARE COSTS



- Software costs often dominate computer system costs. The costs of software on a PC are often greater than the hardware cost.

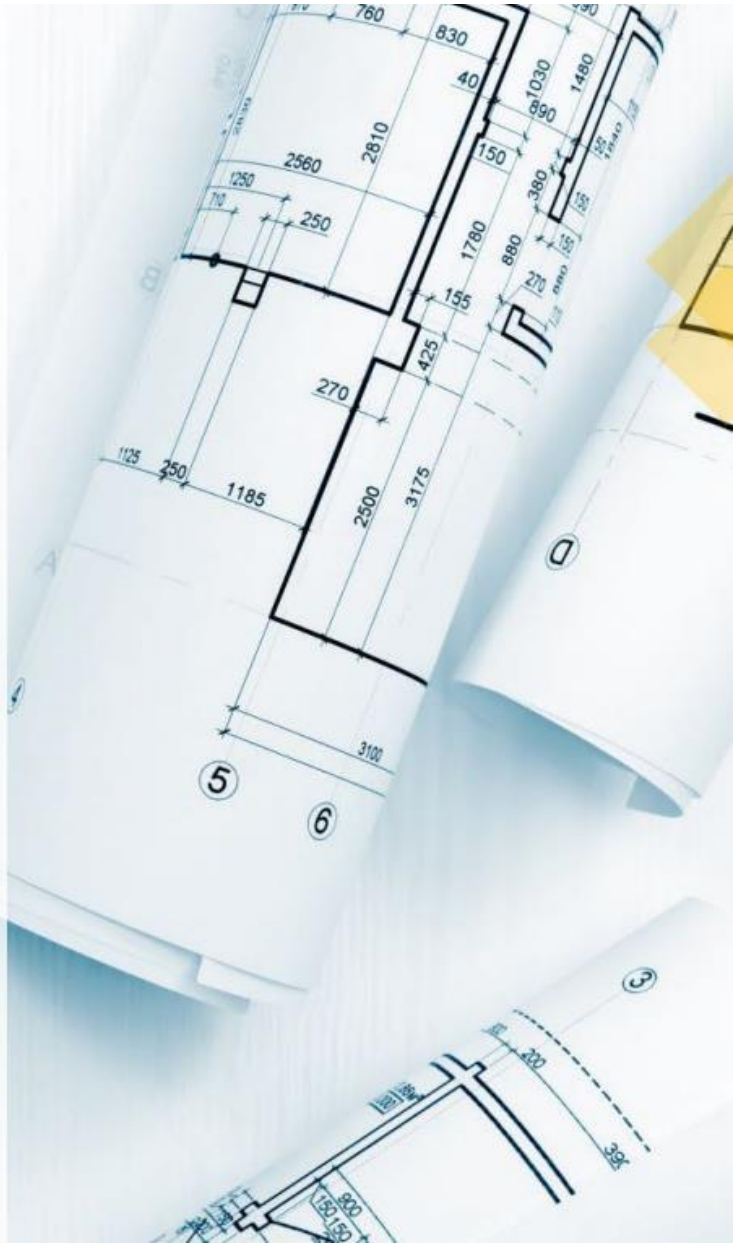
- Software costs more to maintain than it does to develop. For systems with a long life, maintenance costs may be several times development costs.

- Software engineering is concerned with cost-effective software development.

Software project failure

- **Increasing system complexity**
  - larger and more complex systems
  - The demands change
  - have to be built and delivered more quickly
  - new capabilities that were previously thought to be impossible.

- **Failure to use software engineering methods**
  - software is often more expensive and less reliable than it should be.



Professional software development

Lots of people write programs

People in business, scientists, hobbyists write programs

However, most software development is a professional activity in which software is developed for business purposes, for inclusion in other devices etc.

Professional software is intended for use by people (who are not develop the software)

It is maintained and changed throughout its life.

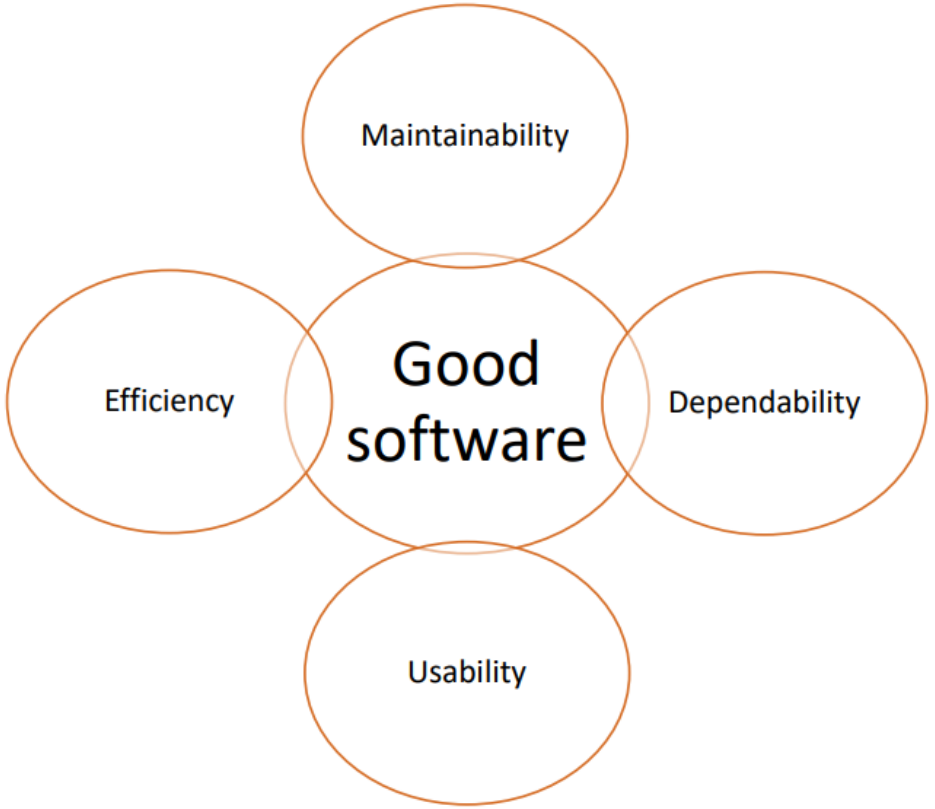
Software Engineering is intended to support professional software development rather than individual programming.

FAQ: 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
- New software can be created by developing new programs, configuring generic software systems or reusing existing software.
- Information systems

FAQ: What are the attributes of good software?

- Good software should deliver the required functionality and performance to the user, and
- It should be maintainable, dependable (reliability, security & safety) and usable



Frequently asked questions about SE

Question	Answer
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 SE

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?	Different techniques are appropriate for different types of system. For example: • games should always be developed using a series of prototypes • safety critical control systems require a complete and analyzable specification to be developed.
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.
  - specification owned by the software developer
  - decisions on software change are made by the developer
  - 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.
  - specification owned by the customer • customer make decisions on software changes that are required
  - Examples – embedded control systems, air traffic control software, traffic monitoring systems.

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.

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.
- 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.

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.

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.

Software process activities

Software specification	• customers and engineers define the software that is to be produced and the constraints on its operation.
Software development	• software is designed and programmed
Software validation	• software is checked to ensure that it is what the customer requires.
Software evolution	• software is modified to reflect changing customer and market requirements.

General issues that affect software

Heterogeneity

- systems are required to operate as distributed systems across networks that include different types of computer and mobile devices.

Business and social change

- Business and society are changing incredibly quickly as emerging economies develop and new technologies become available. T
- need to be able to change their existing software and to rapidly develop new software.

Security and trust

- As software is intertwined with all aspects of our lives, it is essential that we can trust that software.

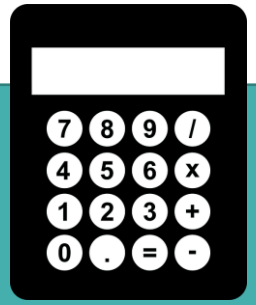
Scale

- Software has to be developed across a very wide range of scales, from very small embedded systems in portable or wearable devices through to Internet-scale, cloud-based systems that serve a global community.

Software engineering diversity – application types

Stand-alone applications

- run on a local computer
- do not need to be connected to a network.
- E.g. calculator, notepad

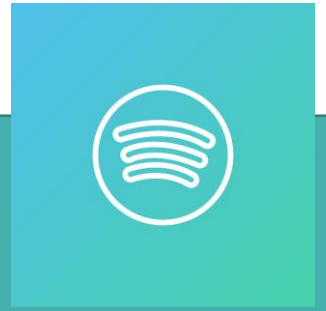


Interactive transaction-based applications

- execute on a remote computer accessed by users from their own PCs or terminals
- e.g. web applications- commerce applications.

Entertainment systems

- primarily for personal use
- intended to entertain the user.
- e.g. Spotify



Embedded control systems

- control and manage hardware devices
- e.g. Traffic control system, smartwatches

Software engineering diversity – application types

Batch processing systems

- designed to process data in large batches for business
- process large numbers of individual inputs to create corresponding outputs.
- e.g. credit card billing system

Systems of systems

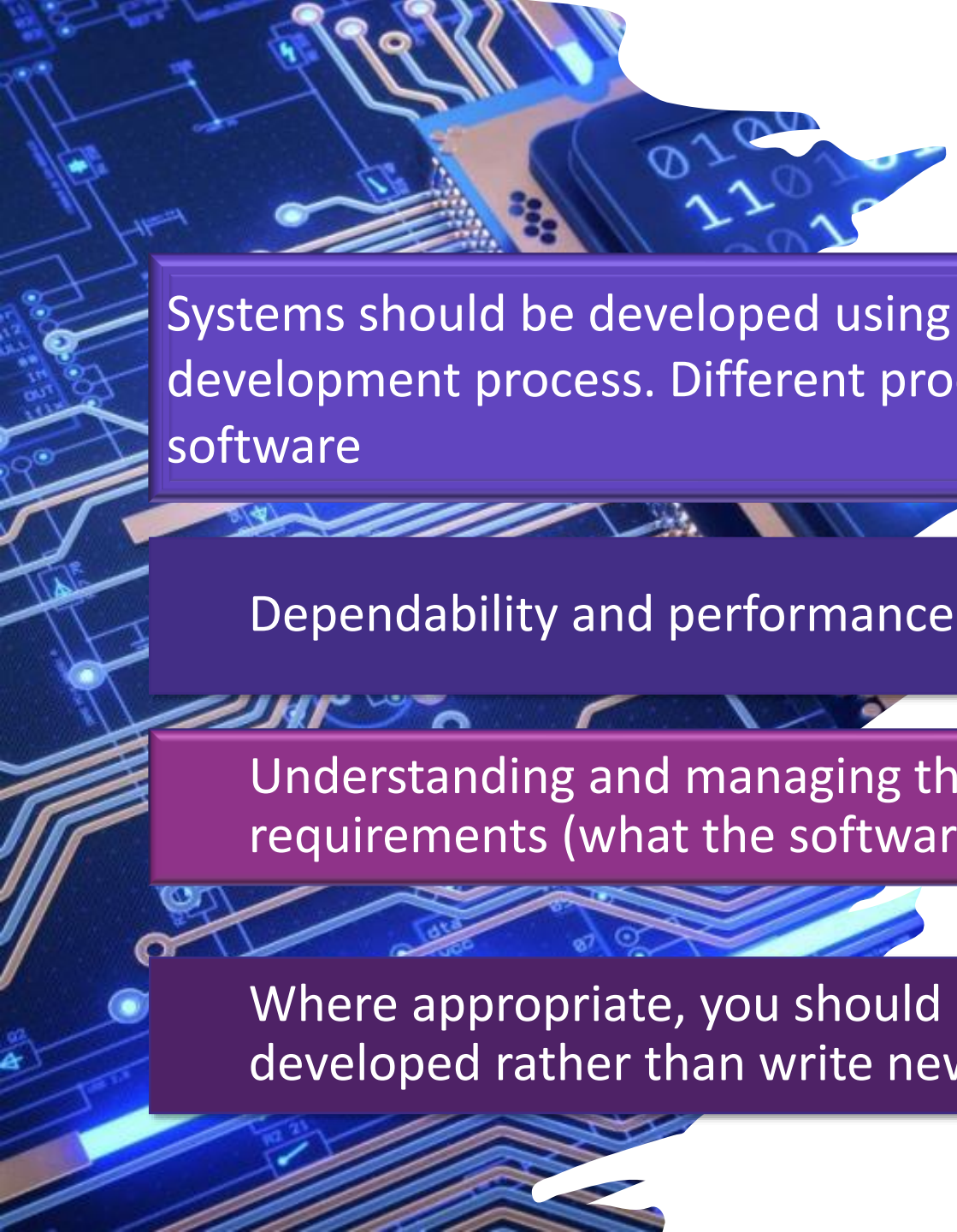
- systems that are composed of a number of other software systems.
- e.g. Disaster Response system = fires + Flood + disease

Data collection systems

- collect data from their environment using a set of sensors
- send that data to other systems for processing.
- e.g. water quality controller

Systems for modeling and simulation

- developed by scientists and engineers
- model physical processes or situations
- include many, separate, interacting objects.
- e.g. healthcare simulation



Software engineering fundamentals principles

- Systems should be developed using a managed and understood development process. Different processes are used for different types of software
- Dependability and performance are important for all types of systems.
- Understanding and managing the software specification and requirements (what the software should do) are important.
- Where appropriate, you should reuse software that has already been developed rather than write new software.

Internet software engineering

The Web is now a platform for running application and organizations are increasingly developing web-based systems rather than local systems.

Web services allow application functionality to be accessed over the web.

Cloud computing is an approach to the provision of computer services where applications run remotely on the ‘cloud’.

- Users do not buy software buy pay according to to use.

Web-based software engineering

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 reuse

how to assemble system from pre-existing software components and systems.

Incremental and agile development

It is impractical to specify all the requirements for such systems in advance.

Service-oriented systems

software components are stand-alone web services.

Rich interfaces

Interface development technologies such as AJAX and HTML5

Topics covered

- Professional software development
  - What is meant by software engineering.

- Software engineering ethics
  - A brief introduction to ethical issues that affect software engineering.

- Case studies
  - An introduction to three examples that are used in later chapters in the book.

Image sources:

https://sevenpillarsinstitute.org/case-studies/list-all-case-studies/
https://bestweb.com.my/3-mistakes-not-to-make-when-choosing-a-software-development-compan
https://growthanimals.com/5-growing-ethical-issues-in-online-marketing-you-need-to-be-aware-of/

Software engineering ethics

Software engineering involves wider responsibilities than simply the application of technical skills.

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

Ethical behaviour is more than simply upholding the law but involves following a set of principles that are morally correct.

Issues of professional responsibility

Confidentiality

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

Competence

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

Intellectual property rights

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

Computer misuse

- Software engineers should not use their technical skills to misuse other people's computers. Computer misuse ranges from relatively trivial (game playing on an employer's machine, say) to extremely serious (dissemination of viruses).

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

The short version of the code summarizes aspirations at a high level of the abstraction; the clauses that are included in the full version give examples and details of how these aspirations change the way we act as software engineering professionals. Without the aspirations, the details can become legalistic and tedious; without the details, the aspirations can become high sounding but empty; together, the aspirations and the details form a cohesive code.

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:

1. PUBLIC – Software engineers shall act consistently with the public interest.
2. CLIENT AND EMPLOYER – Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.
3. PRODUCT – Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
4. JUDGMENT – Software engineers shall maintain integrity and independence in their professional judgment.
5. MANAGEMENT – Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
6. PROFESSION – Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
7. COLLEAGUES – Software engineers shall be fair to and supportive of their colleagues.
8. SELF – Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Rationale for the code of ethics

- Computers have a central and growing role in commerce, industry, government, medicine, education, entertainment and society at large. Software engineers are those who contribute by direct participation or by teaching, to the analysis, specification, design, development, certification, maintenance and testing of software systems.

- Because of their roles in developing software systems, software engineers have significant opportunities to do good or cause harm, to enable others to do good or cause harm, or to influence others to do good or cause harm. To ensure, as much as possible, that their efforts will be used for good, software engineers must commit themselves to making software engineering a beneficial and respected profession.

Topics covered

- Professional software development
  - What is meant by software engineering.

- Software engineering ethics
  - A brief introduction to ethical issues that affect software engineering.

- Case studies
  - An introduction to three examples that are used in later chapters in the book.

Image sources:

https://sevenpillarsinstitute.org/case-studies/list-all-case-studies/
https://bestweb.com.my/3-mistakes-not-to-make-when-choosing-a-software-development-compan
https://growthanimals.com/5-growing-ethical-issues-in-online-marketing-you-need-to-be-aware-of/

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

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

What is Mentcare? (in this study)

An information system that is intended for use in clinics

Use a centralized database of patient information

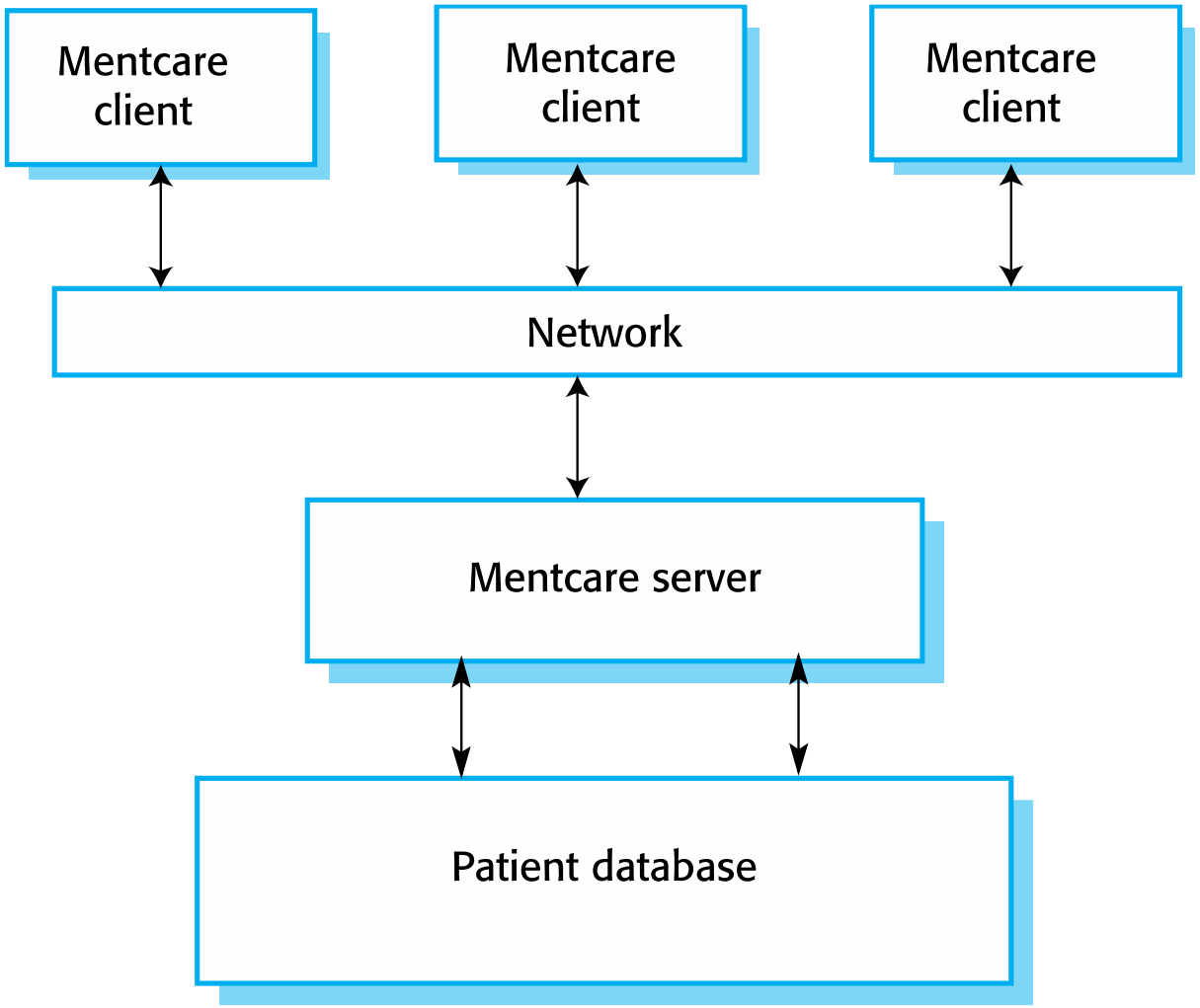
designed to run on a PC

GOAL?

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.

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.

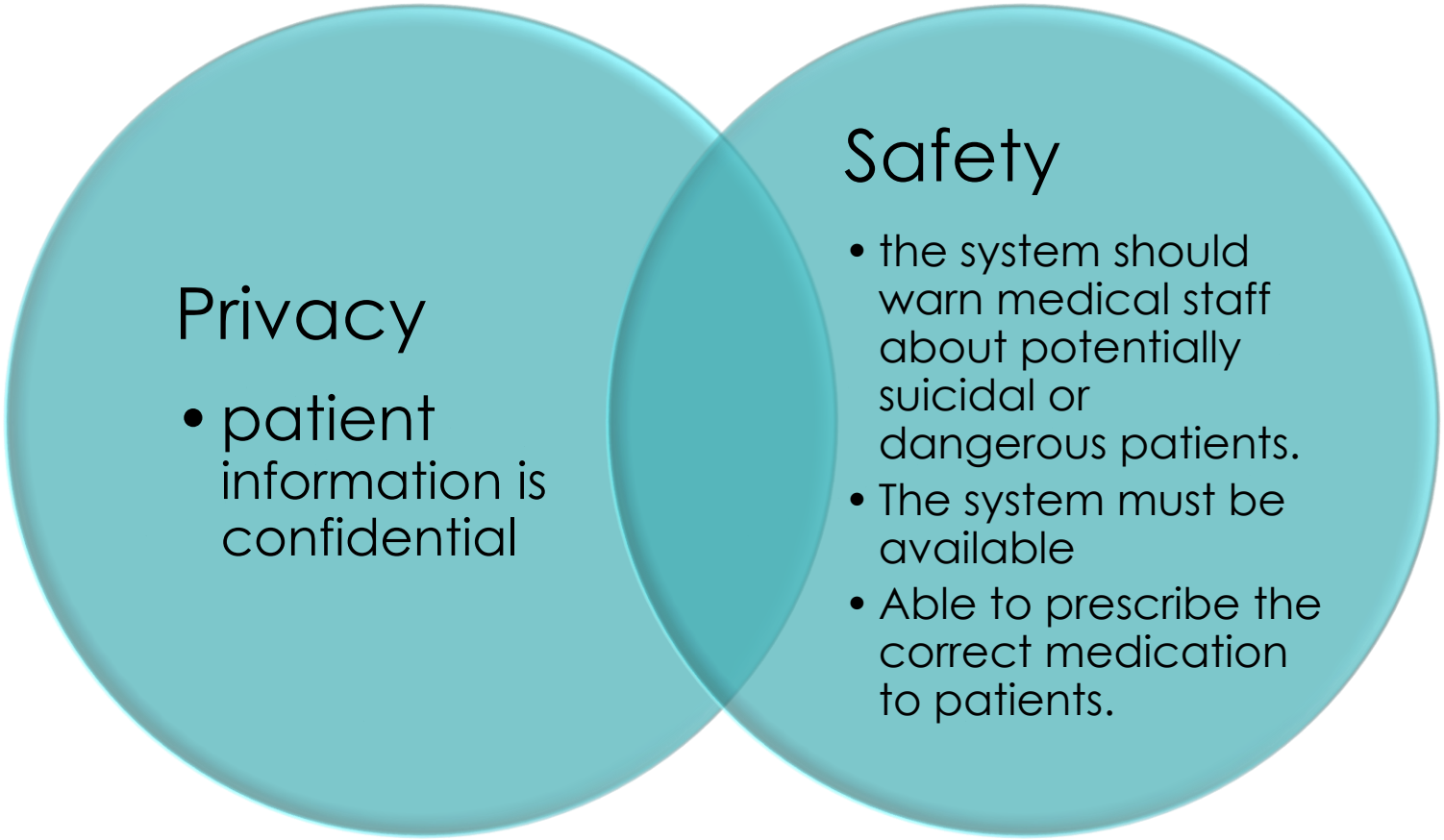
Patient Monitoring

monitors the records of patients that are involved in treatment and issues warnings if possible problems are detected.

Administrative reporting

management reports showing the number of patients treated at each clinic, the number of patients sectioned, the drugs prescribed, etc.

Mentcare system concerns



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.
- There are many different types of system, 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.