Application programming in engineering;

Introduction

10.1.2019

Who am I?

- M.Sc from Aalto university
- Worked with software industry since 2011
  - 5 years in FTA as developer / specialist, since 2016 in private sector as an consultant
- Different roles from developer to scrum master to software architect to GIS consultant

What does a software engineer actually do for work?

- Solve problems
  - It always involves learning something new and applying what you know
  - Usually it involves geographic information and systems
  - Most of the times it involves software engineering
  - Many times it involves putting together multiple existing programs
  - Often it involves designing software systems
  - Sometimes it involves programming

"Every good work of software starts by scratching a developer's personal itch"

Basic lectures of the course:

- Gives you a basic understanding about a software engineering environment
- Explains the different approaches and parts of the software engineering
- After these lectures you are ready to start to work in software engineering project and learn more

Basic lectures:

- Introduction, 10.1.2018
- Process and methods, 15.1.2018
- Delivering quality, 17.1.2018

Reading exercise with questions before every lecture

Reading material for the basic lectures:

- Book: Ian Sommerville, Software Engineering, 10th edition
- Slides of the book: http://iansommerville.com/software-engineering-book/slides/

Content

- What is a software
- Software vs. application
- What software engineering is
- Why software engineering is so important
- Software process
- Software engineering ethics

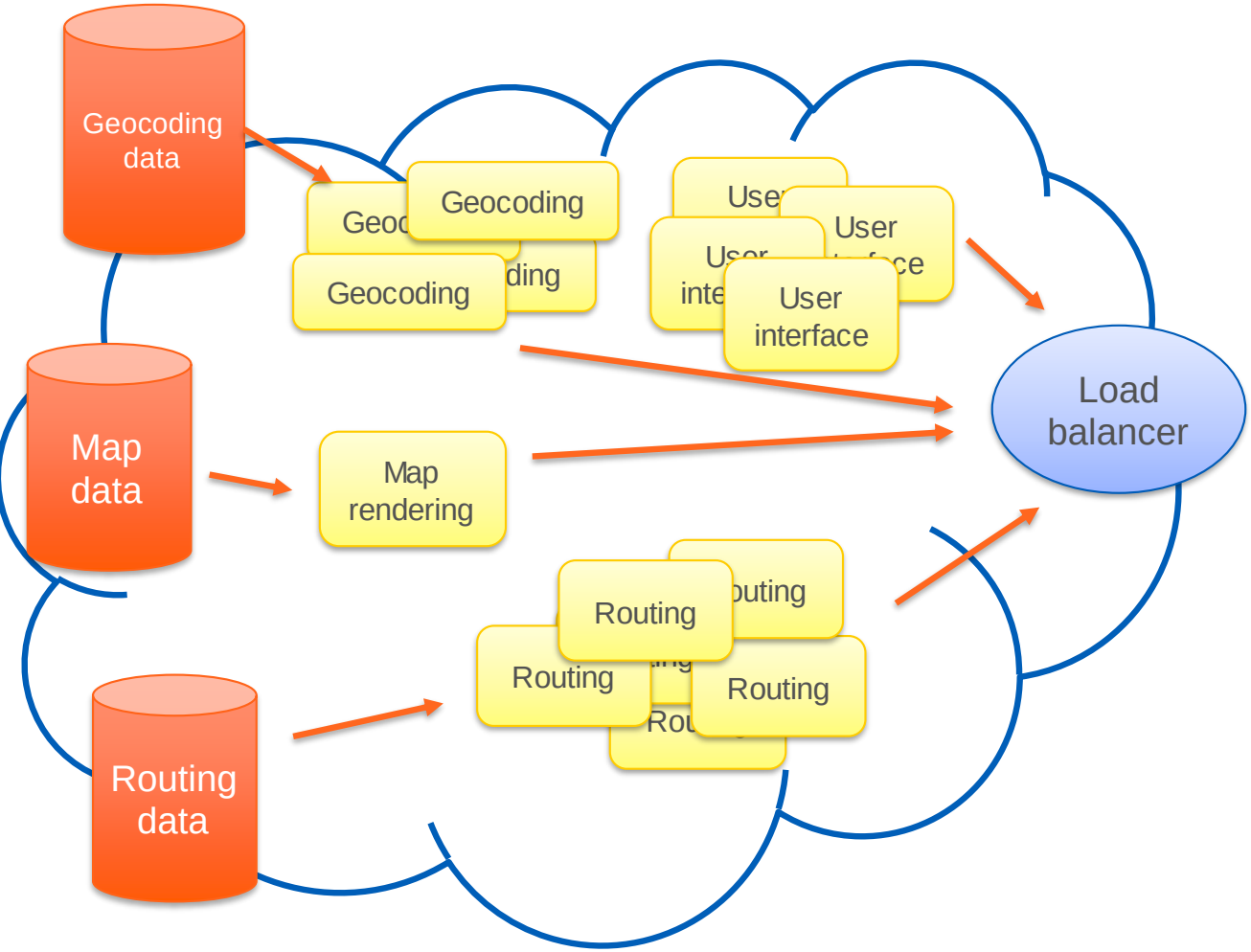
What is a software?

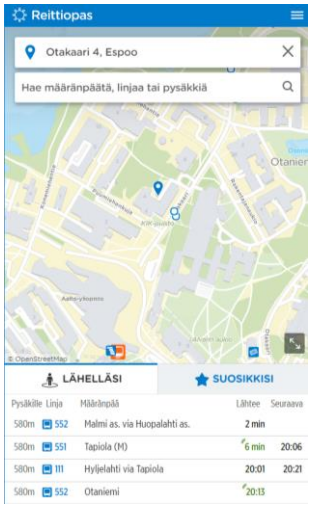
Computer programs and associated documentation. Software product may be developed for a particular customer or may be developed for a general market.

- Generic products
  - Stand-alone systems that are marketed and sold to any customer who wishes to buy them.
  - Examples – desktop and mobile software such as games, graphics software, presentation software; GIS 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, data management systems, nontrivial GIS analysis solutions

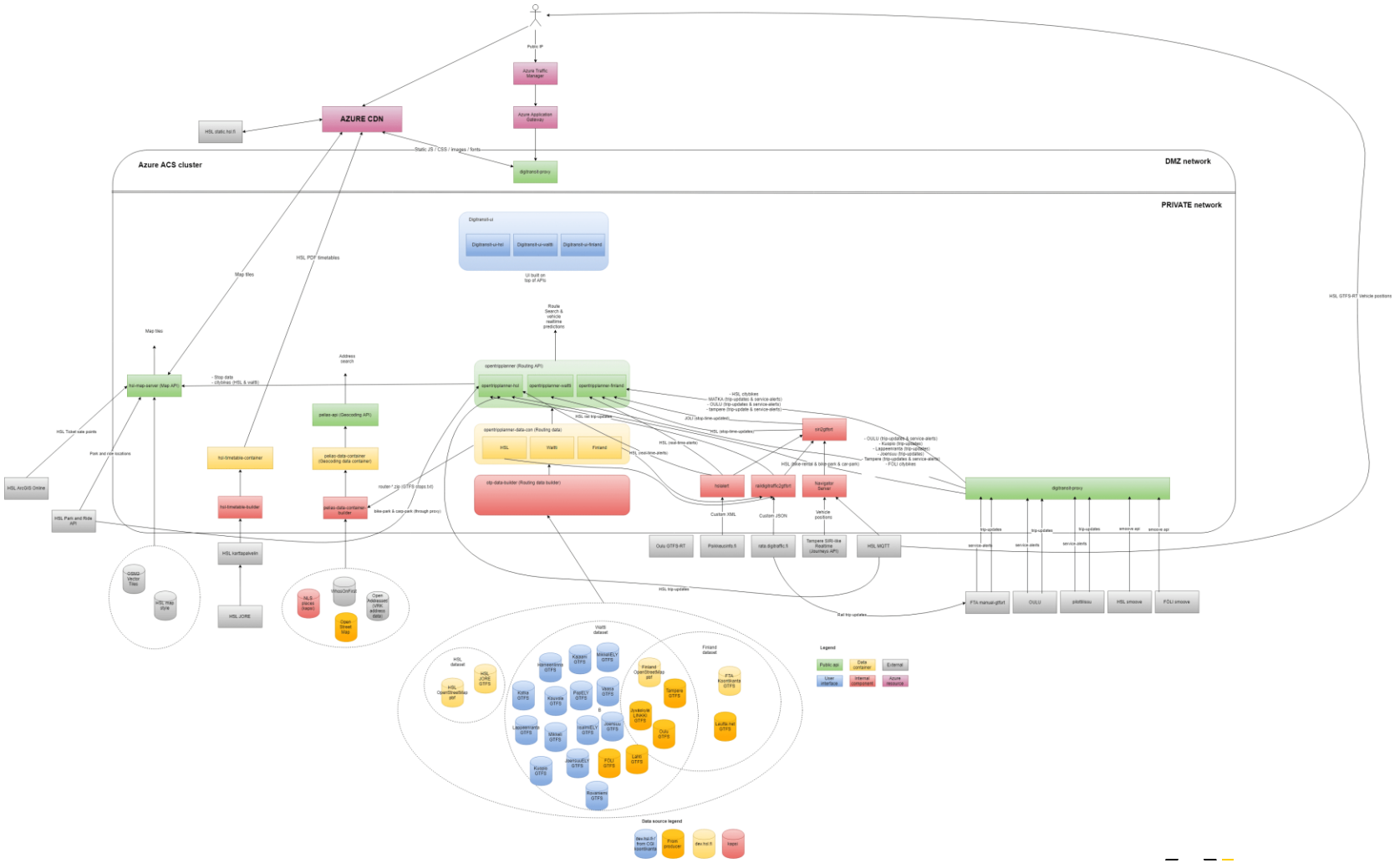
- Generic products
  - The specification of what the software should do is owned by the software developer and decisions on software change are made by the developer.
- Customized products
  - The specification of what the software should do is owned by the customer for the software and they make decisions on software changes that are required based on their own business needs.

Digitransit / reittiopas.hsl.fi





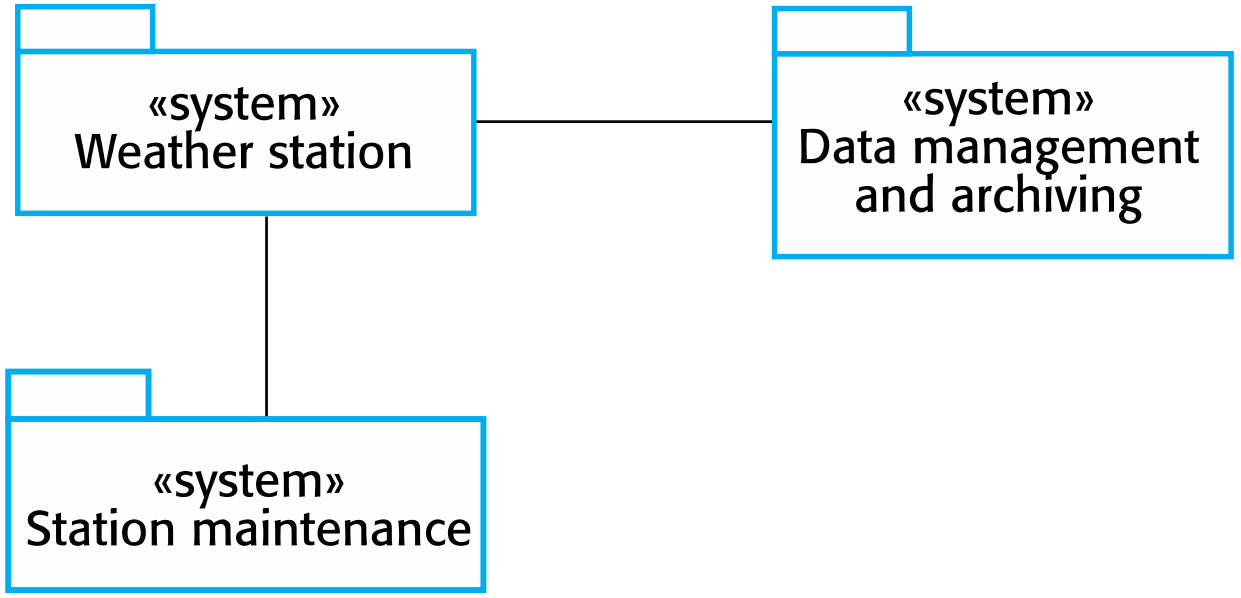
Digitransit / reittiopas.hsl.fi



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.

Wilderness weather station



Wilderness weather station

- 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.

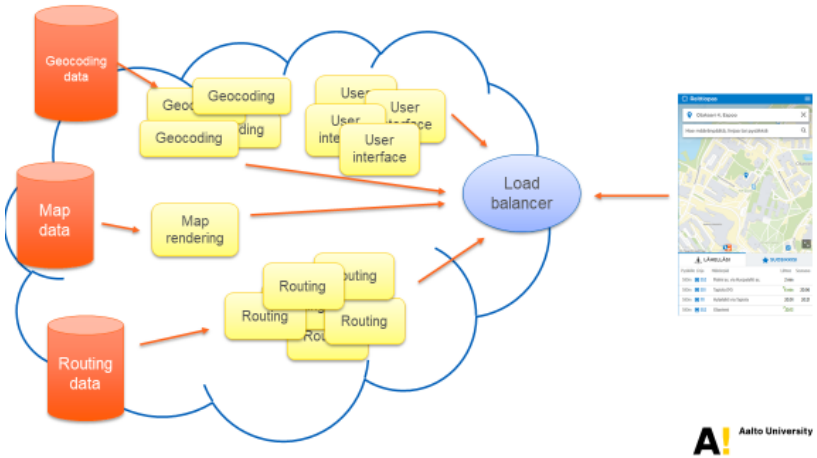
Wilderness weather station – additional software functionality

- 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.

YYT-C3002
Application Programming in Engineering
Spring 2019

YYT-C3002
Application Programming in Engineering
Spring 2018

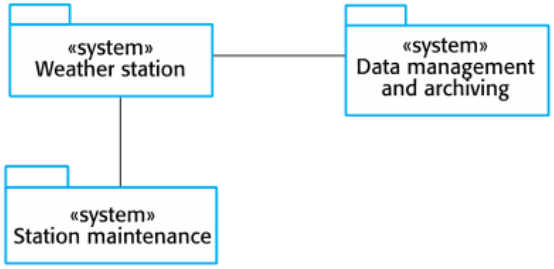
Reittiopas.fi



Aalto University

YYT-C3002
Application Programming in Engineering
Spring 2018

Wilderness weather station



Aalto University

Aalto University

Software vs. Application

- All applications are software but not all software are applications
- Application is executable
  - Application needs to perform a specific task
- Applications are operating system specific but software is not
- Applications needs more often user interaction than software

Software 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.

Software types

- 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.

Software types

- 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.

Essential attributes of good software

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.*
- Surrounding realities, hardware, environment and business requirements are always changing
- No software can foresee all scenarios that it will face

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.*
- Software controls systems that our society and lives literally depends on
- The black swan theory & failure of imagination

Efficiency

- *Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilization, etc.*
- Moore's law vs. Wirth's law
- “You can always buy more memory” is not viable strategy
- Efficiency = money

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 is (usually) designed to be used by a user
- Users are human, normally specific focus group
- Understanding the underlying problem is crucial
- Software should make work easier, not harder
  - Most of the gains can be archived by process and method changes

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.

What software engineering is?

- Software engineering ≠ Computer science
- Software engineering is about solving problems with the help of software.
- 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.

Software engineering fundamentals

- Some fundamental engineering principles apply to all types of software system, irrespective of the development techniques used:
  - Systems should be developed using a managed and understood development process. Of course, different processes are used for different types of software.
  - Dependability and performance are important for all types of system.
  - 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.
- Web technologies offers universal unified platform for solving many problems
- 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 use.

Web-based software engineering

- Web-based systems are complex distributed systems but the fundamental principles of software engineering discussed previously are as applicable to them as they are to any other types of system.
- The fundamental ideas of software engineering apply to web-based software in the same way that they apply to other types of software system.

Web software engineering

- Software reuse
  - Software reuse is the dominant approach for constructing web-based systems. When building these systems, you think about how you can assemble them from pre-existing software components and systems.
- Incremental and agile development
  - Web-based systems should be developed and delivered incrementally. It is now generally recognized that it is impractical to specify all the requirements for such systems in advance.

Web software engineering

- Service-oriented systems
  - Software may be implemented using service-oriented software engineering, where the software components are stand-alone web services.
- Rich interfaces
  - Interface development technologies such as Javascript and HTML5 have emerged that support the creation of rich interfaces within a web browser.

Why software engineering is so important?

- Almost every company is a software company
- 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.

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*
  - As new software engineering techniques help us to build larger, more complex systems, the demands change. Systems have to be built and delivered more quickly; larger, even more complex systems are required; systems have to have new capabilities that were previously thought to be impossible.
- *Failure to use software engineering methods*
  - It is fairly 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.

Software Process

Software process activities

- Software specification, where customers and engineers define the software that is to be produced and the constraints on its operation.
- Software development, where the software is designed and programmed.
- Software validation, where the software is checked to ensure that it is what the customer requires.
- 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.

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.

Software engineering code of ethics (ieee)

- PUBLIC
  - Software engineers shall act consistently with the public interest.
- 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.
- PRODUCT
  - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
- JUDGMENT
  - Software engineers shall maintain integrity and independence in their professional judgment.

https://www.computer.org/web/education/code-of-ethics

Software engineering code of ethics (ieee)

- MANAGEMENT
  - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
- PROFESSION
  - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
- COLLEAGUES
  - Software engineers shall be fair to and supportive of their colleagues.
- 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.

https://www.computer.org/web/education/code-of-ethics

Rationale for the code of ethic

- *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.*