

Software Engineering Introduction

HUSZERL Gábor
huszerl@mit.bme.hu



Méréstechnika és
Információs Rendszerek
Tanszék




**Critical Systems
Research Group**

Learning Outcomes

At the end of the lecture the students are expected to be able to

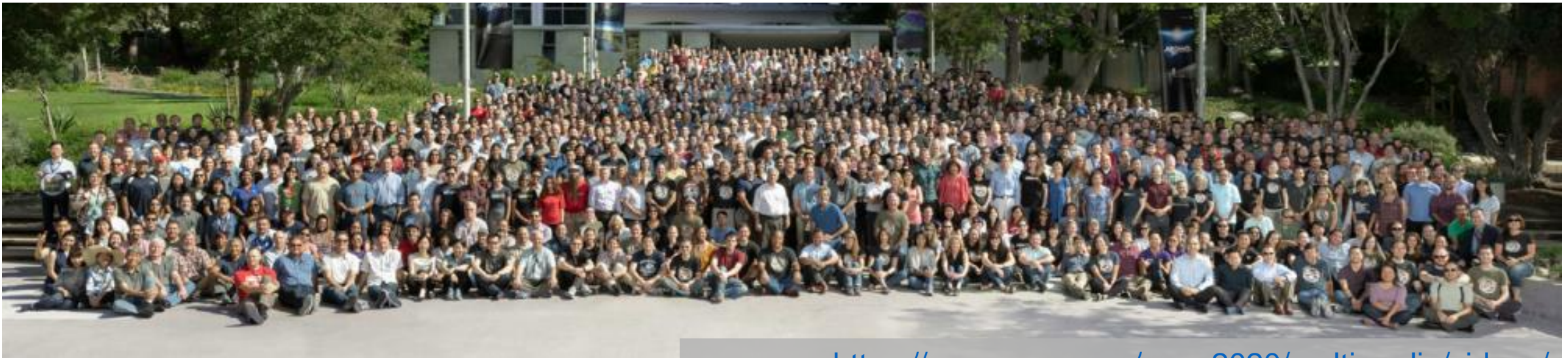
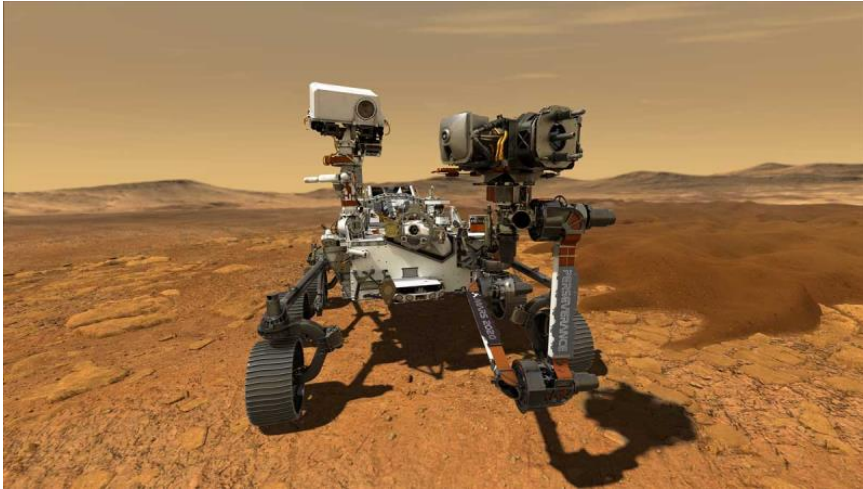
- (K1) present the goals and specific features of soft. eng.,
- (K1) list the specificities of the different application domains,
- (K2) summarize the aspects, tasks and actors involved in software development,
- (K1) recall what career paths exist in the field of IT



Where has software development got to in the last 50 years?

Why is it still so difficult to make good software?

NASA Perseverance Video

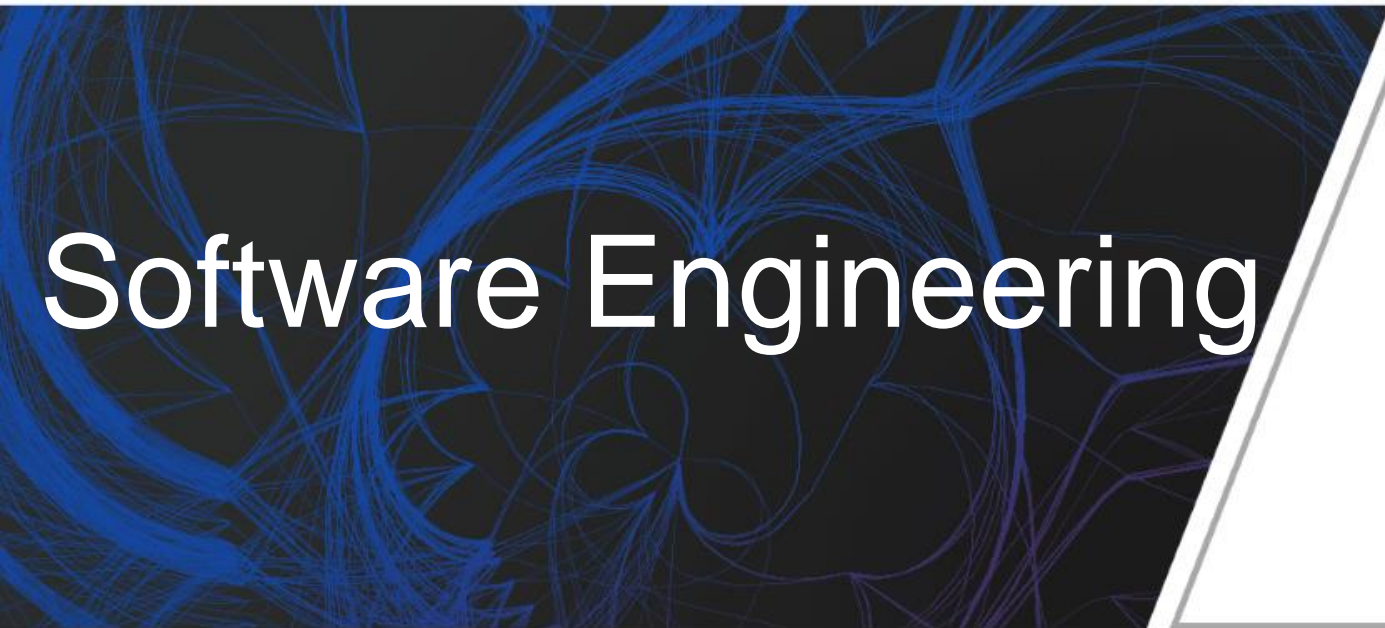


<https://mars.nasa.gov/mars2020/multimedia/videos/>

Where Has Software Development Come From in the Last 50 Years?



Margaret Hamilton beside printouts of the Apollo flight software (1960's)



Software Engineering

Software, Engineering, Software Engineering

What Is Software?

software: computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.

Source: IEEE, "Systems and software engineering — Vocabulary,"
ISO/IEC/IEEE Standard 24765, 2010

Not just the program code!

Software System

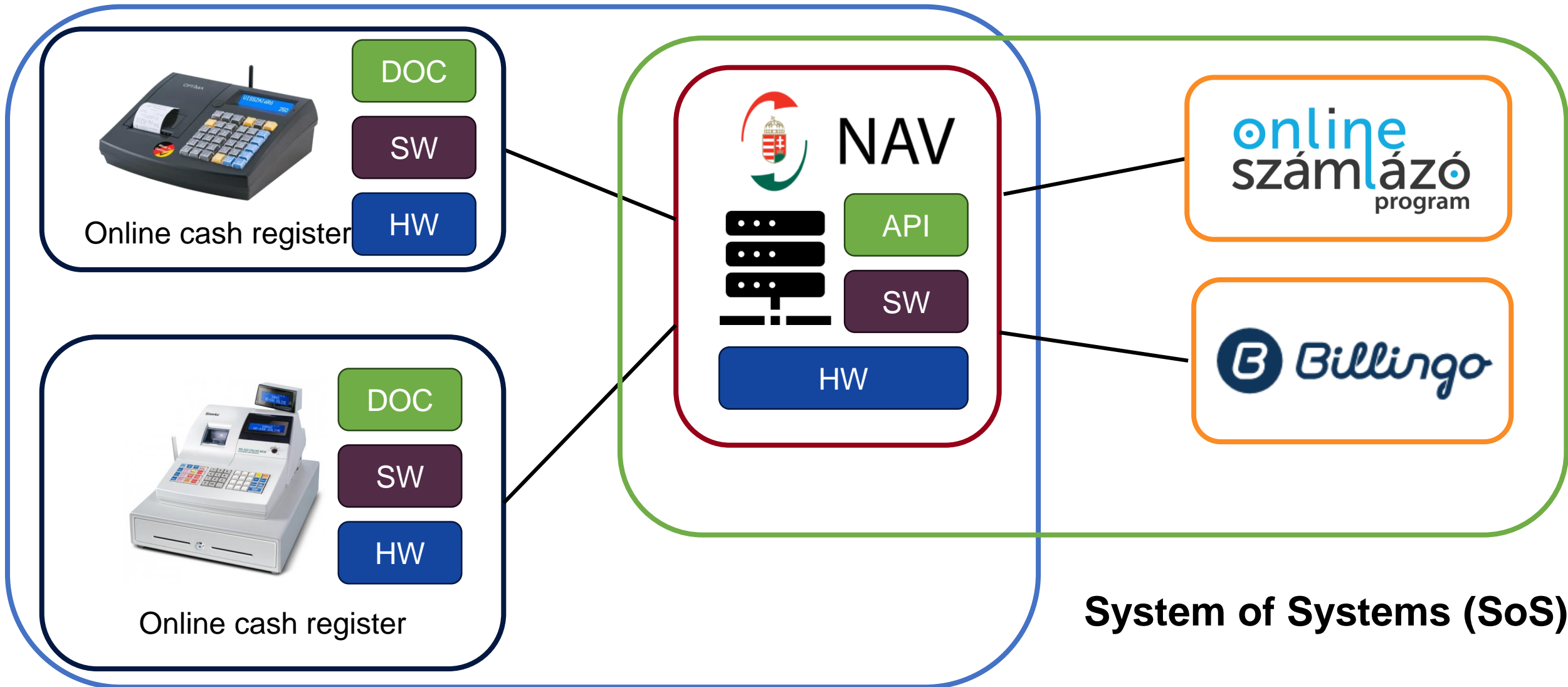
software system: a system made up of software, hardware, and data that provides its primary value by the execution of the software.

Source: OMG, "Essence – Kernel and Language for Software Engineering Methods"
formal/18-10-02, 2018

system: combination of interacting elements organized to achieve one or more stated purposes.

Source: ISO/IEC 15288:2008 (IEEE Std 15288-2008)

Example: Software and System



System of Systems (SoS)

Engineering

engineering: the application of a systematic, disciplined, quantifiable approach to structures, machines, products, systems, or processes.

Source: IEEE, "Systems and software engineering — Vocabulary," ISO/IEC/IEEE Standard 24765, 2010



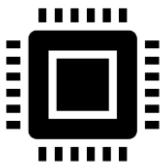
How Does a „Real” Engineer Work?



Customer requirements

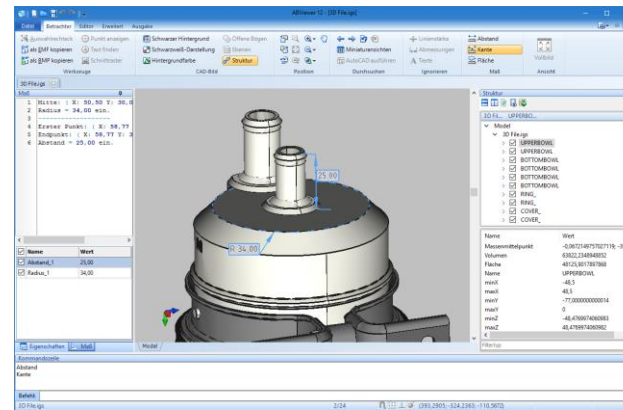


Standards

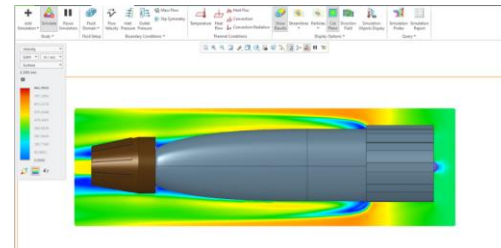


Existing Components

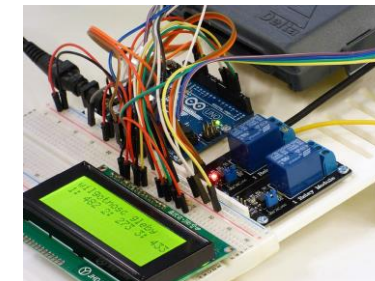
Design, modelling



Production



Simulation, analysis



Prototype, testing

Software Engineering

software engineering: an engineering discipline that focuses on the development and use of rigorous methods for designing and constructing software artifacts that will reliably perform specified tasks.

Source: ACM, "Computing Curricula 2020"

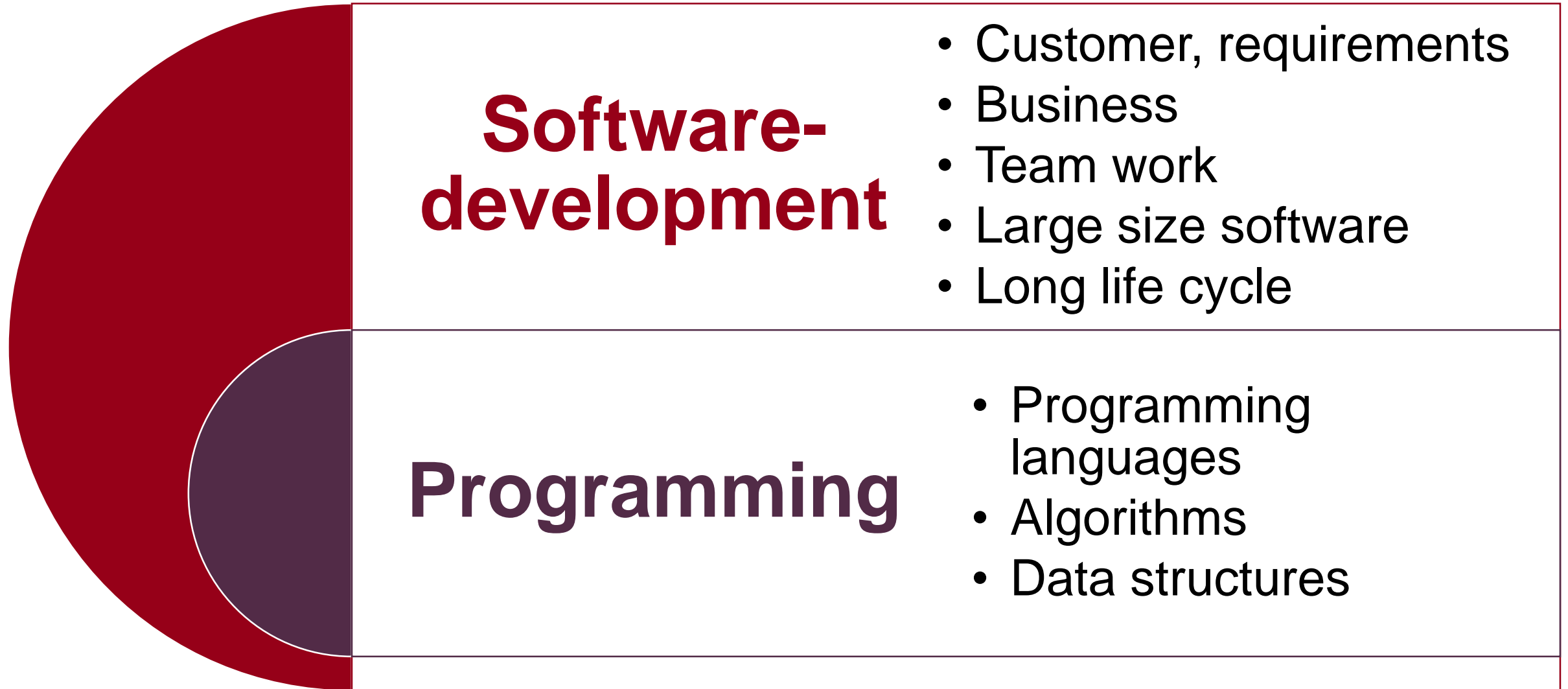
Software engineering is programming integrated over time.

Source: "Software Engineering at Google", 2020

The multiperson development of multiversion programs.

Source: Brian Randell or David Parnas (not clear)

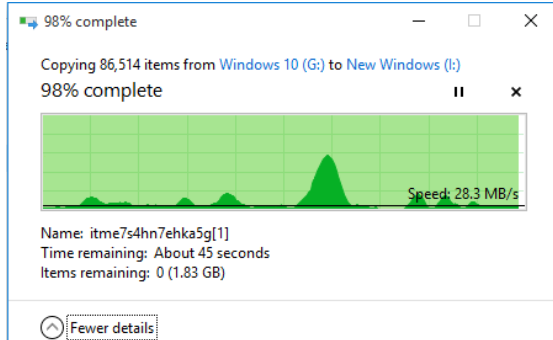
Programming and Software Development



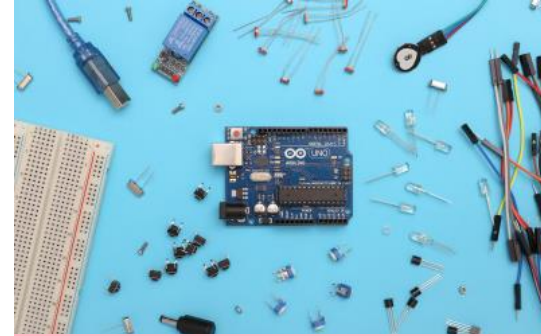
What Are the Specificities of Software



Production



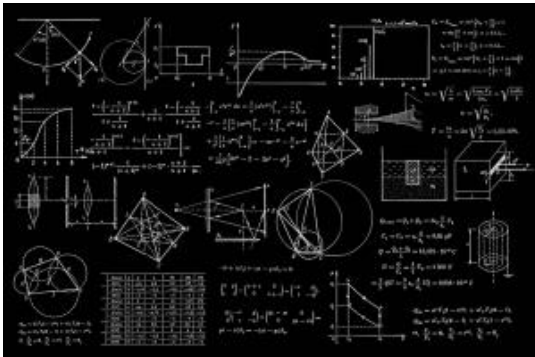
Easy reproduction



Occurances of faults



Consistency



Physical constraints



Logical concepts



Difficult modification



Easy rewrite

Pictures: [Unsplash](https://unsplash.com)

Where Has Software Development Reached?

SOFTWARE ENGINEERING

Report on a conference sponsored by the
NATO SCIENCE COMMITTEE
Garmisch, Germany, 7th to 11th October 1968

Chairman: Professor Dr. F. L. Bauer
Co-chairmen: Professor L. Bolliet, Dr. H. J. Helms

Editors: Peter Naur and Brian Randell

January 1969

1969: software
crisis?

The Cost of Poor Software Quality in the US: A 2020 Report

HERB KRASNER
MEMBER, ADVISORY BOARD
CONSORTIUM FOR INFORMATION & SOFTWARE QUALITY™ (CISQ™)
WWW.IT-CISQ.ORG
HKRASNER@UTEXAS.EDU
DATE: JANUARY 1, 2021

CISQ
Consortium for Information & Software Quality™

2020: yearly “costs” of wrong SW \$2000 billion



Software: Successes and Failures



Linux (1991-)

- „just a hobby, won't be big and professional like gnu”
- ~4 thousand contributors, ~80e commits per year



Mars Orbiter (1998)

- Burnt at entry to Mars
- \$125 million damage
- Imperial → SI conversion



CERN LHC (1998-)

- 25 petabytes per year
- Safety critical control SW
- Several 10,000 researchers



Knight Capital (2012)

- Test code started to trade (bad update)
- \$440 million damage, chaos



Google Maps (2005-)

- Large user base
- Simple user interface
- StreetView



Poly Network (2021)

- Connecting crypto chains
- Erroneous smart contract
- \$620 million (~returned)

Software for Whom?



Specificities of the application domains

Software in every Area of Life



Web Based and Mobile Applications



Large size software and data

- SWs running on several 10 thousand servers
- ! Hyperscaling (SW, company, team, ...)



User experience (UX)

- Experimentation, rapidly changing needs
- ! Fast feedback and adaptation

See „Mobile and Web Based Software” subject

Business and Enterprise Applications



Governmental and enterprise SW

- Tailored SW, involving external suppliers
- ! Detailed specifications and contracts



Commercial (boxed) products

- Diverse target group: from SME to multi
- ! Customizability and consultancy

See „Management of Information Systems” course

Critical and Embedded Systems



Embedded systems

- Physical components, long life cycle, costs
- ! Collaboration with other engineers

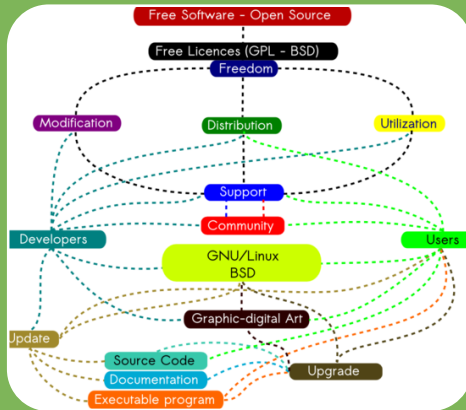


Safety critical systems

- Danger of damaging people or values
- ! Guaranteeing expected correct operation

See „System Modelling” and „Embedded computer systems” courses

Open Source Libraries and Tools



Open source development

- Brought radically new methods
- ! Asynchronous, distributed projects

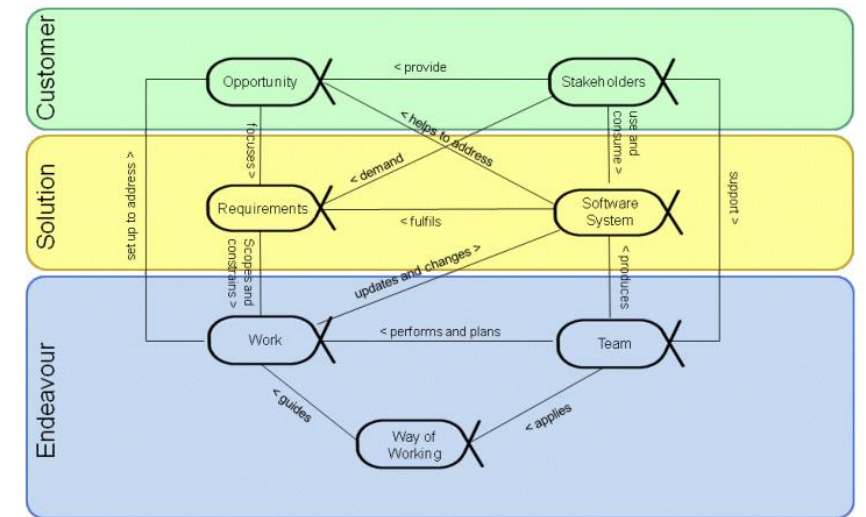


SW tools and libraries

- OS, compilers, network SW, ...
- ! Maintenance costs, community

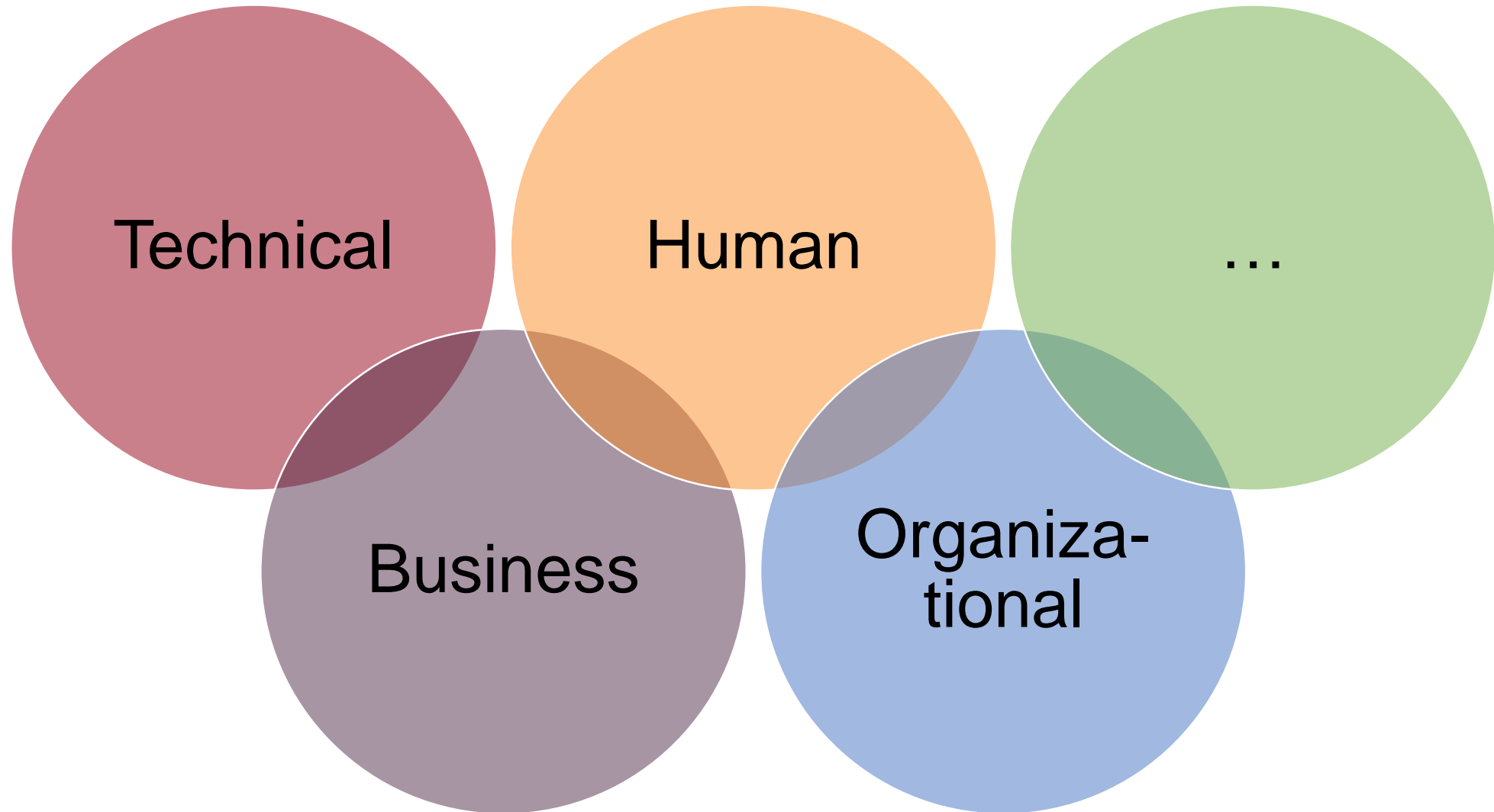
See „Open source and free software” elective subject

How?



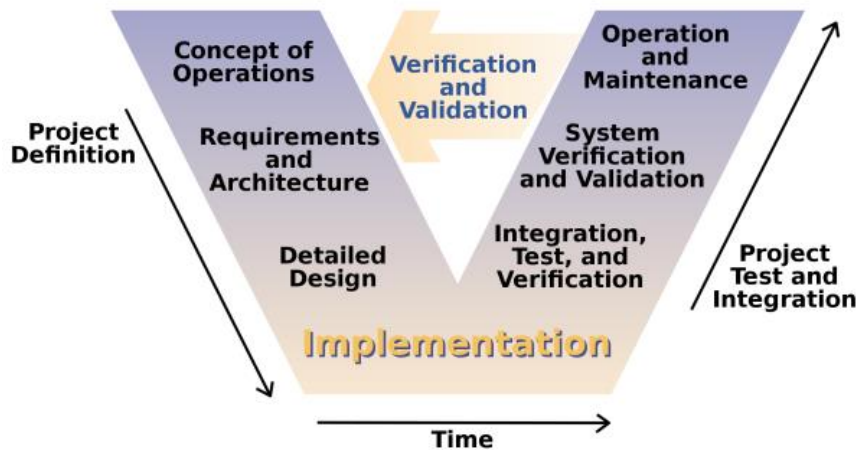
Methods and best practices

Aspects of Software Engineering

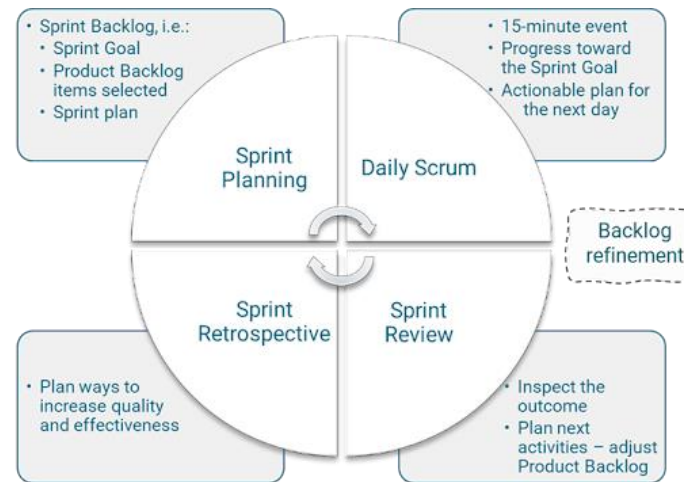


Software Engineering Methods

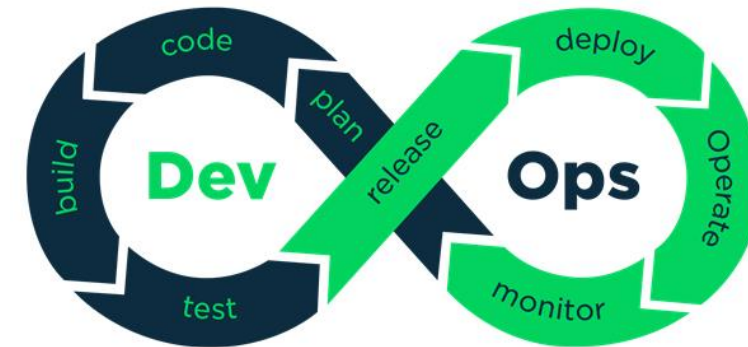
How to make (good) software?



V-model



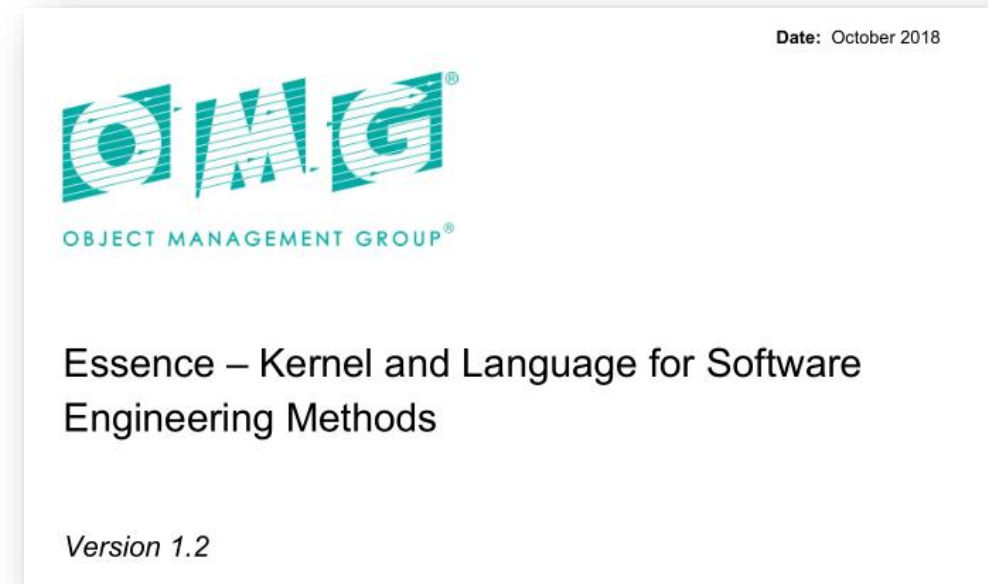
Scrum (agile)



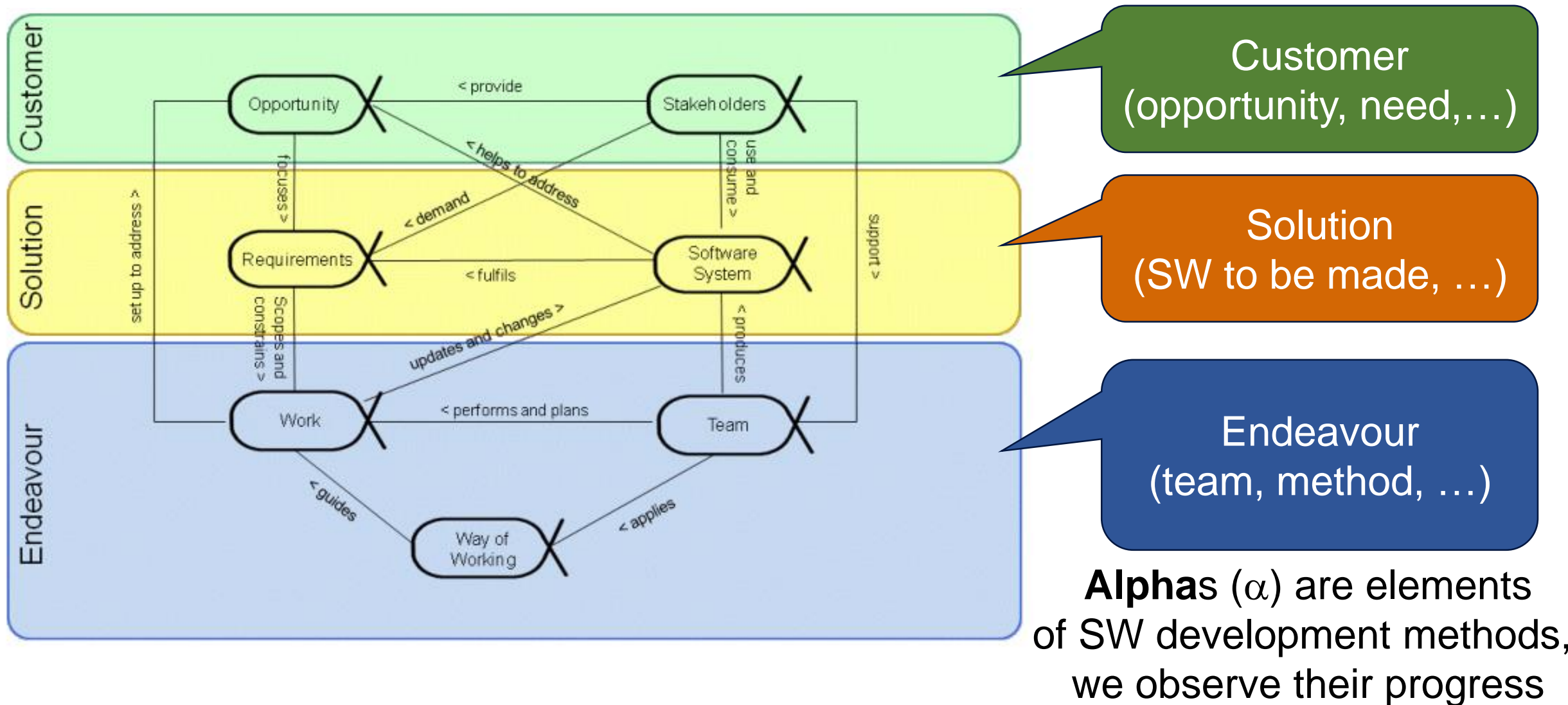
DevOps

What do the Methodologies Have in Common?

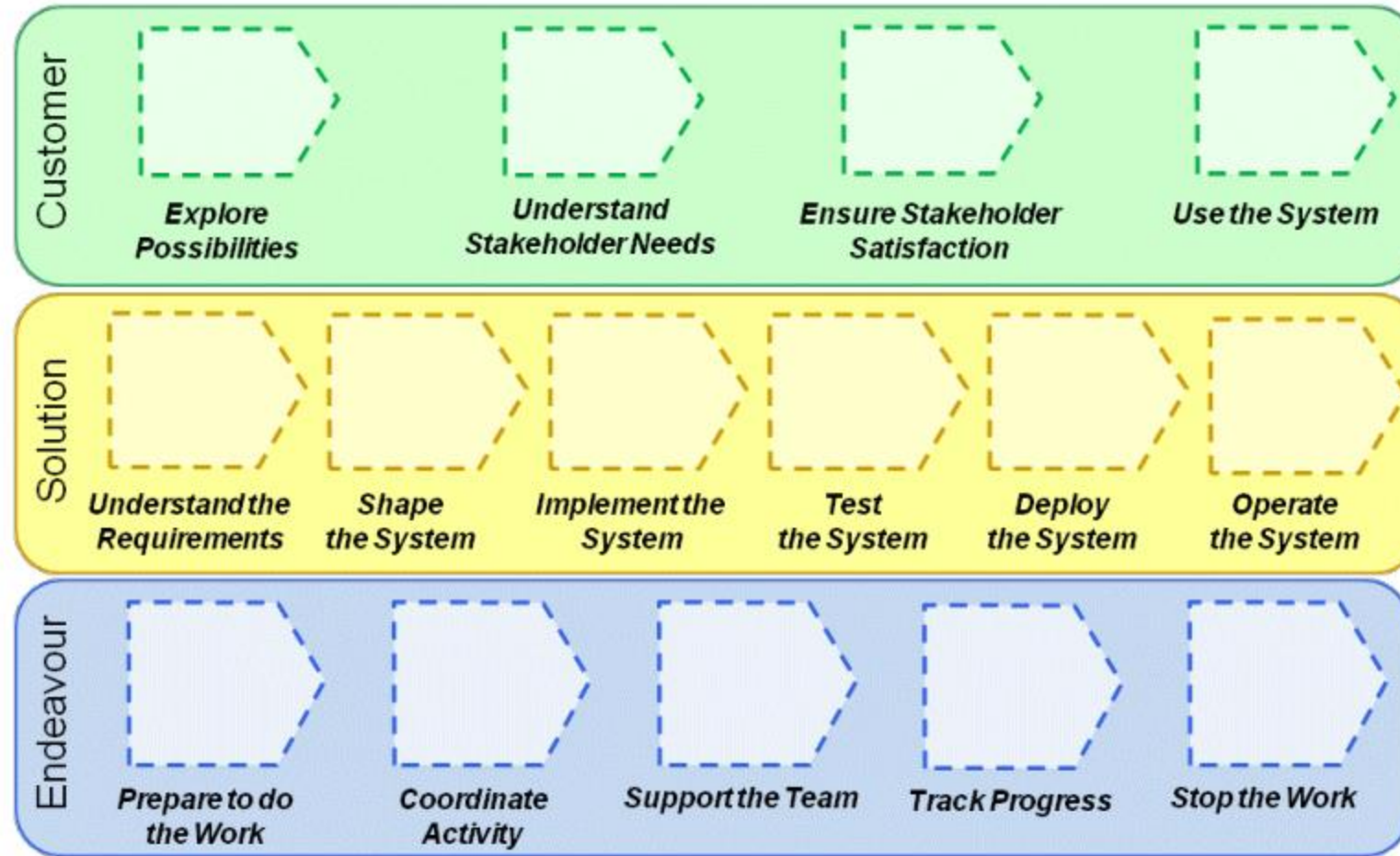
- Object Management Group (OMG)
- **Essence** standard
- Identifying a common „kernel”
- The best practices and different methods can be built on it



Essence: What Is There to Deal with?

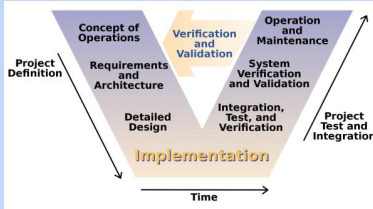


Essence: What Can Be Done on Them?



Methods and Practices (Examples)

Methods



Practices

Pair programming

Domain-driven Design (DDD)

Continuous integration (CI)

Cross-functional teams

Test-driven development (TDD)

User stories

Hardware-in-the-loop testing (HiL)

There Is No Single Redeeming ...

- There is no best practice
- There is no best method
- Everything depends on the circumstances
- One must always choose, decide

Different Environments – Different Goals

New service of a start-up

- „Move fast and break things”
- Little documentation, simple processes

Kanban?
Testing in a production environment?

Car braking system

- Compliance with safety standards
- Multi-level checking

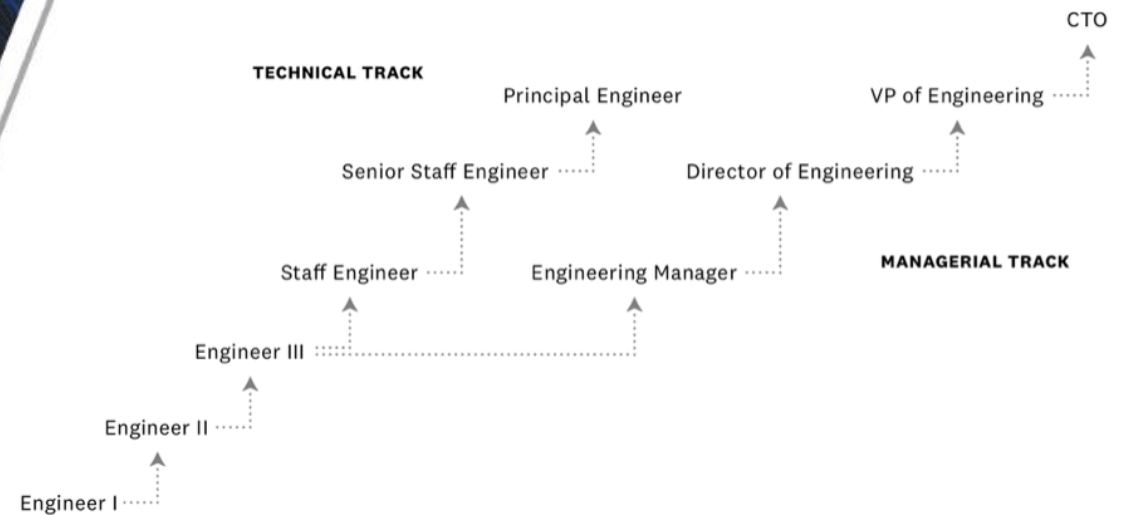
Modell based development?
MiL / SiL / HiL testing?

Bank supplier

- Certifications (ISO, CMMI...)
- Following well defined processes

Scrum / SAFe?
Acceptance tests?

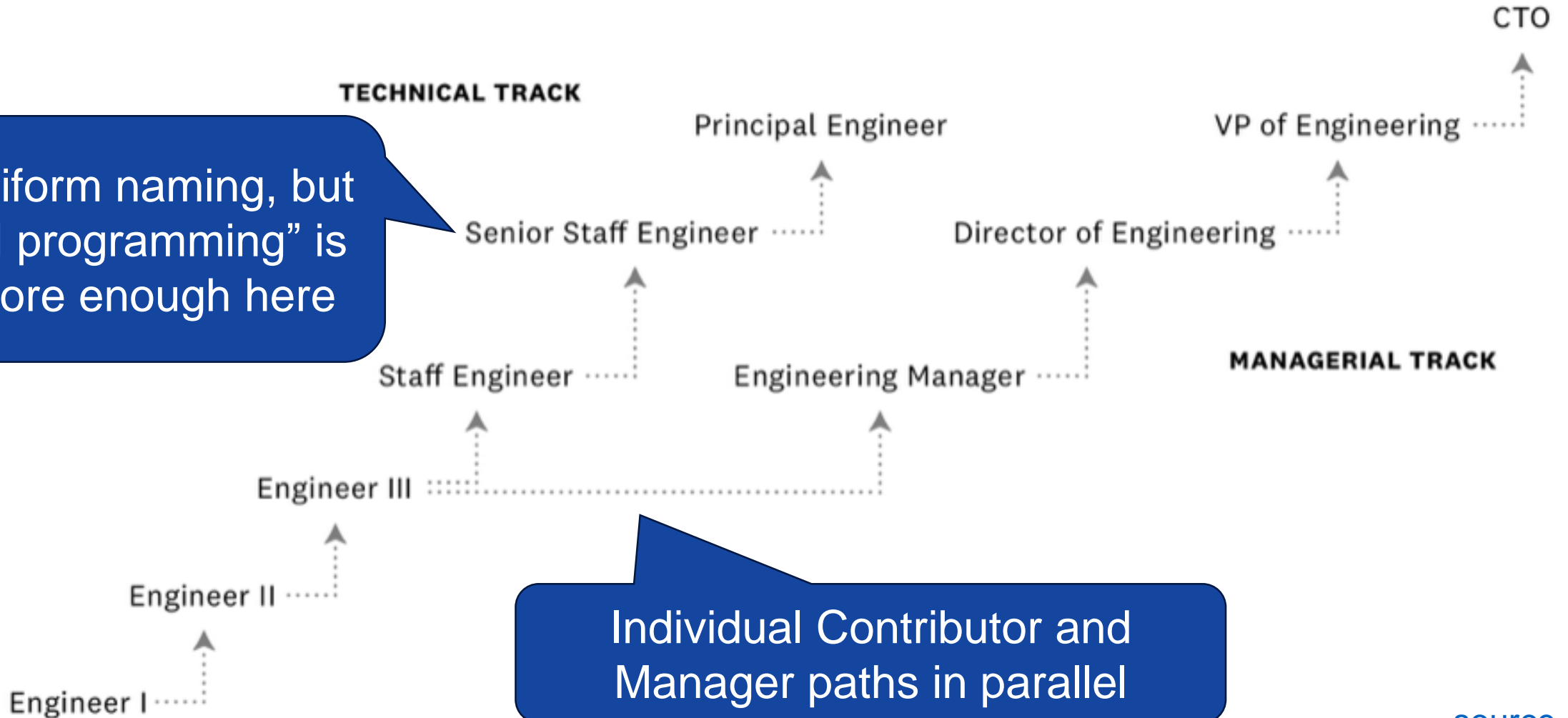
Professional development



Career paths, organizations, ethics

Career Paths for Software Developer

No uniform naming, but
“good programming” is
no more enough here



Individual Contributor and
Manager paths in parallel

[source](#)

What do I need to improve? (Example)

Technical skills	Quality & testing	Writing code		Delivery	Incremental value delivery	Work breakdown
		Testing				Prioritisation, dependencies
	Debugging & observability	Debugging			Self-organization	Dealing with ambiguity
		Observability				Reliability, delivery accountability
	Software design & architecture	Understanding Code		Economic thinking		
		Software Architecture				
	Security	Security				
Feedback, Communication , Collaboration	Feedback	Delivering Feedback		Leadership	Decision making	Decision making
		Seeking and receiving feedback			Driving alignment	Driving alignment
	Communication	Effective communication			Process thinking	Process thinking
		Knowledge Sharing			Facilitation	Facilitation
	Collaboration	Teamwork		Mentoring	Mentoring	
		Relationship building				
		Handling disagreement				
				Strategic Impact	Business Acumen & Strategy	Business acumen
			Strategic work			
			Product Thinking			

CircleCI career matrix

Many aspects, not only technical ones!

At higher levels, more and more people are affected more and more by our own work

Many detailed examples: <https://progression.fyi/>

Different Positions in Software Engineering

Test engineer

Network / system
administrator

Data scientist

Software engineer

....

Product manager

Software architect

Technical writer / Evangelist

Professional Organizations



Association for
Computing Machinery



Standards
Development
Organization®



IEEE

*Advancing Technology
for Humanity*



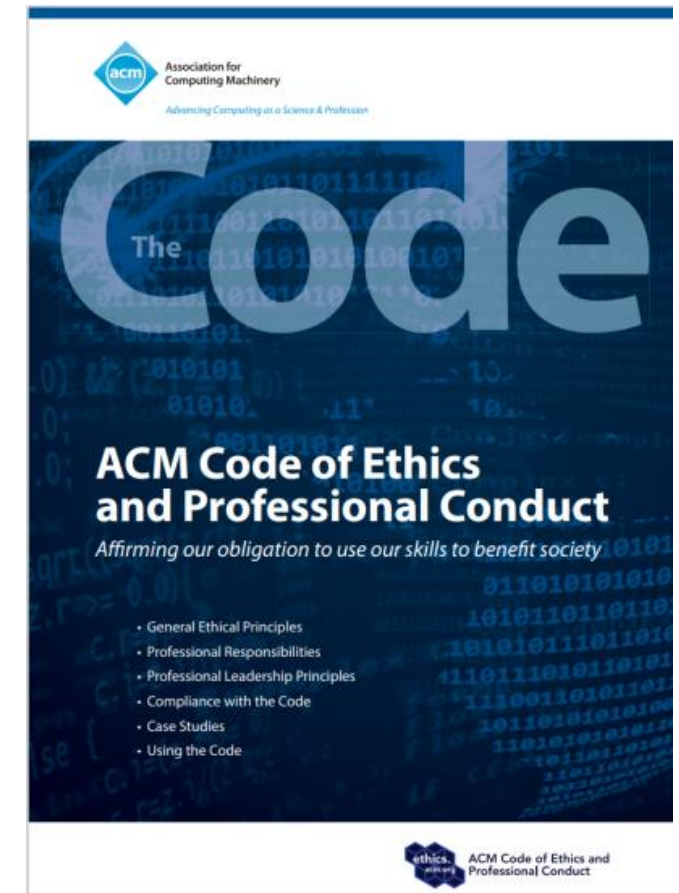
International Software
Testing Qualifications Board



Ethical and Professional Working

ACM Code of Ethics and Professional Conduct

- Contribute to society and to human well-being
- Avoid harm
- Be honest and trustworthy
- Be fair and take action not to discriminate
- Respect privacy
- Honor confidentiality
- ...



<https://www.acm.org/code-of-ethics>

Ethical and Professional Working?



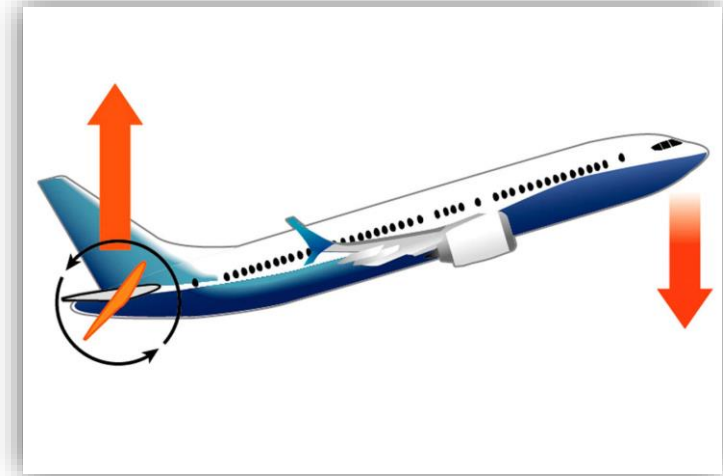
[source](#)

Dieselgate



[source](#)

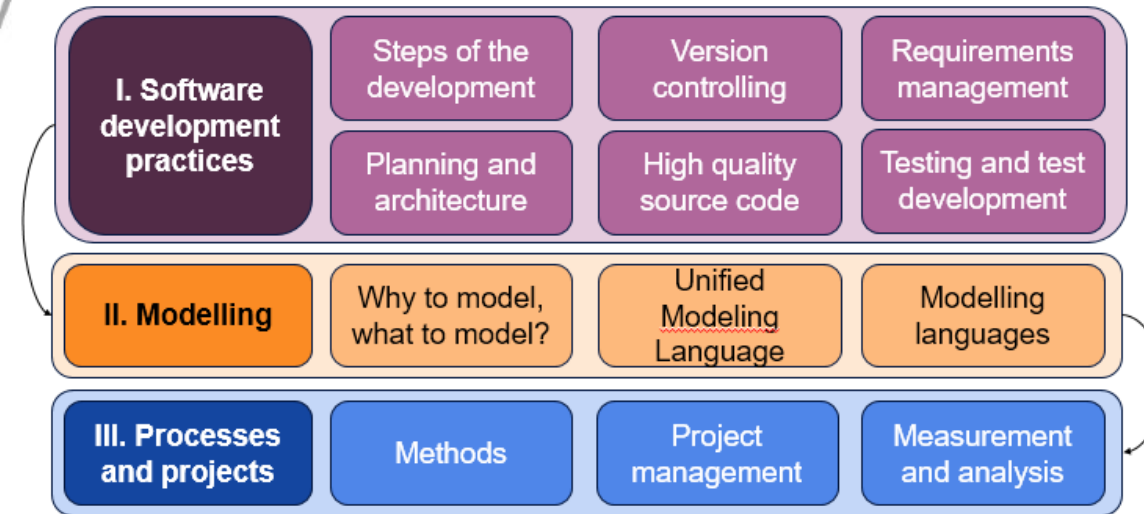
Cambridge Analytica



[source](#)

737 MAX MCAS

What next?



Summary

Further Topics of the Subject

I. Software development practices

Steps of the development

Version controlling

Requirements management

Planning and architecture

High quality source code

Testing and test development

II. Modelling

Why to model, what to model?

Unified Modeling Language

Modelling languages

III. Processes and projects

Methods

Project management

Measurement and analysis

Useful Resources (worth downloading!)

- **IEEE standards** (the PDF is currently not available ☹)
 - [24765-2010](#) Systems and SW engineering – Vocabulary
 - [SE VOCAB](#) – online searchable format available (!)
- **Software Engineering Book of Knowledge (SWEBOK)**
 - [Version V3](#) available and downloadable
- **International Software Testing Qualifications Board (ISTQB)**
 - [Foundation Level Syllabus](#) (v4.0)
 - [Glossary of Testing Terms](#) (EN / HU)



Summary

Software Engineering

software engineering: an engineering discipline that focuses on the development and use of rigorous methods for designing and constructing software artifacts that will reliably perform specified tasks.

Source: ACM, "Computing Curricula 2020"

Software engineering is programming integrated over time.

Source: "Software Engineering at Google", 2020

The multiperson development of multiversion programs.

Source: Brian Randell or David Parnas (not clear)

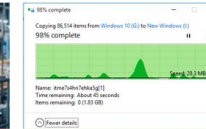
Software Engineering (VIMIAB04)



What Are the Specificities of Software



Production



Easy reproduction



Occurrences of faults



Consistency



Physical constraints



Logical concepts



Difficult modification



Easy rewrite

Pictures: [Unsplash](#)

Software Engineering (VIMIAB04)



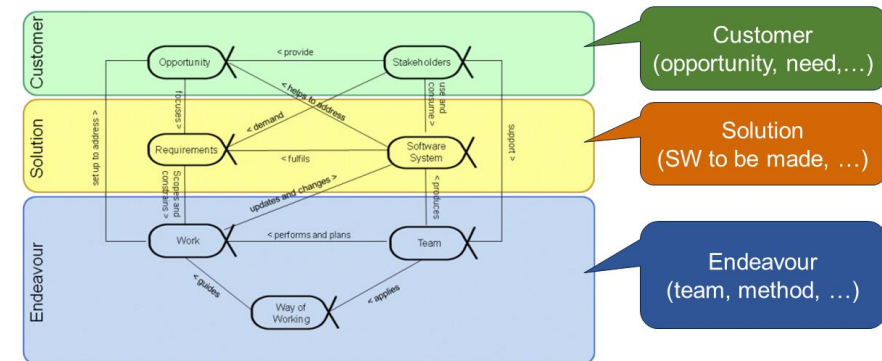
Software in every Area of Life



Software Engineering (VIMIAB04)



Essence: What Is There to Deal with?



Software Engineering (VIMIAB04)

