Software Engineering

Prof. Dr. Jóakim von Kistowski

TH Aschaffenburg
university of applied sciences

Prof. Dr. Jóakim von Kistowski

Computer Science
Karlsruhe Institute of Technology

PhD Computer Science
University of Würzburg

Research Group Leader
University of Würzburg

Senior Software Architect / DevOps Team Lead
DATEV eG

Member, Chair, Contributor
Standard Performance Evaluation Corporation (SPEC)

Research Topics

Looking for Bachelor/Master Thesis Topics?

GreenIT and Energy Efficiency

- Energy Efficiency of Servers
- Energy Efficiency of Software
- Green Software Operations
- Efficient Management of Server Fleets

Software Architecture and Quality

- Microservice Architectures
- Cloud-native Computing
- Software Resilience and Chaos Testing
- Software Performance Benchmarking and Analysis

Thanks and Acknowledgments

This lecture and the practical exercises are based on the German speaking class by

Prof. Dr. Timea Illes-Seifert

- **Warning**: This class deviates from its German counterpart in some aspects!

Welcome!

TH Aschaffenburg
university of applied sciences

Task for today's lecture

Slip into the role of a sensationalist reporter. What are the headlines that summarize today's lecture? Write along!

What is Software Engineering?

Software and Engineering

Disciplines and Challenges

Project-Based vs. Product-Based SWE

Further Terms

What is Software Engineering?

Software and Engineering

Disciplines and Challenges

Project-Based vs. Product-Based SWE

Further Terms

TH Aschaffenburg
university of applied sciences

Learning Objectives (1/2)

- You can explain what software is.
- They can use examples to explain what characterizes engineering disciplines.
- They can give examples of why the transfer of old engineering principles to software development is difficult and what solutions are available.

Learning Objectives (2/2)

- You know the origins of software engineering. You can explain what is meant by the term "software crisis".
- You can explain what software engineering is.
- You can discuss the extent to which progress in SWE has led to an improvement in the situation 50 years ago.

- What is Software Engineering?
- Software and Engineering
- Disciplines and Challenges
- Project-Based vs. Product-Based SWE
- Further Terms

TH Aschaffenburg
university of applied sciences

What is **software** engineering?

Software

Engineering

SWE?

When did the term software first appear?

1940

1950

1960

1970

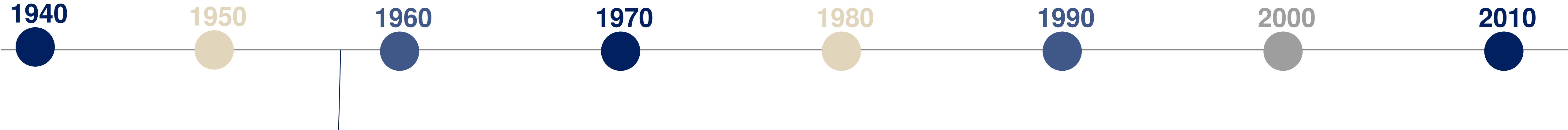
1980

1990

2000

2010

When did the term software first appear?



1957

John. W. Tukey (1915-2000) coined the term "software" as the opposite of hardware

https://de.wikipedia.org/wiki/John_W._Tukey

Hardware dates back to the 16th century and was a generic term for objects made of metal ("ironware").

Question

What is Software?

What is software?
IEEE definition

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

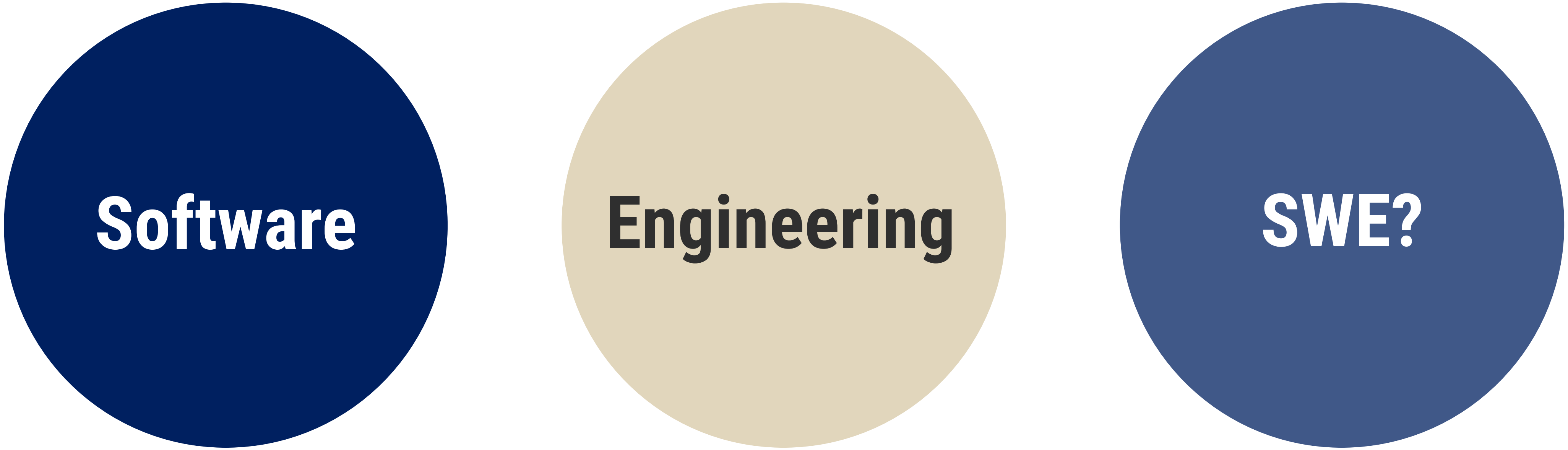
IEEE Std. 610.12 (1990)

http://www.informatik.htw-dresden.de/~hauptman/SEI/IEEE_Standard_Glossary_of_Software_Engineering_Terminology%20.pdf

Software
All programs, processes, rules, documentation and data relating to the operation of computer systems.

IEEE Std. 610.12 (1990)

What is software **engineering**?



Software

Engineering

SWE?

Question

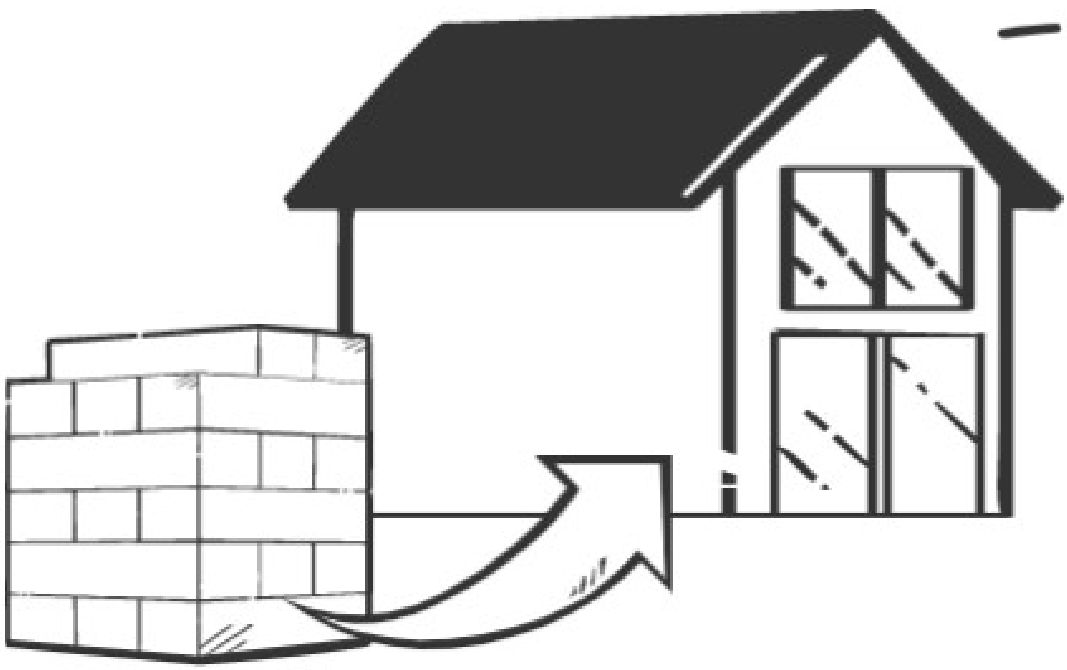
What qualities do you associate with an engineer?

What is engineering?

> *Scientists build to learn, engineers learn to build.*
>
> Fred Brooks [1977]

What is engineering?

Characteristics of engineering disciplines



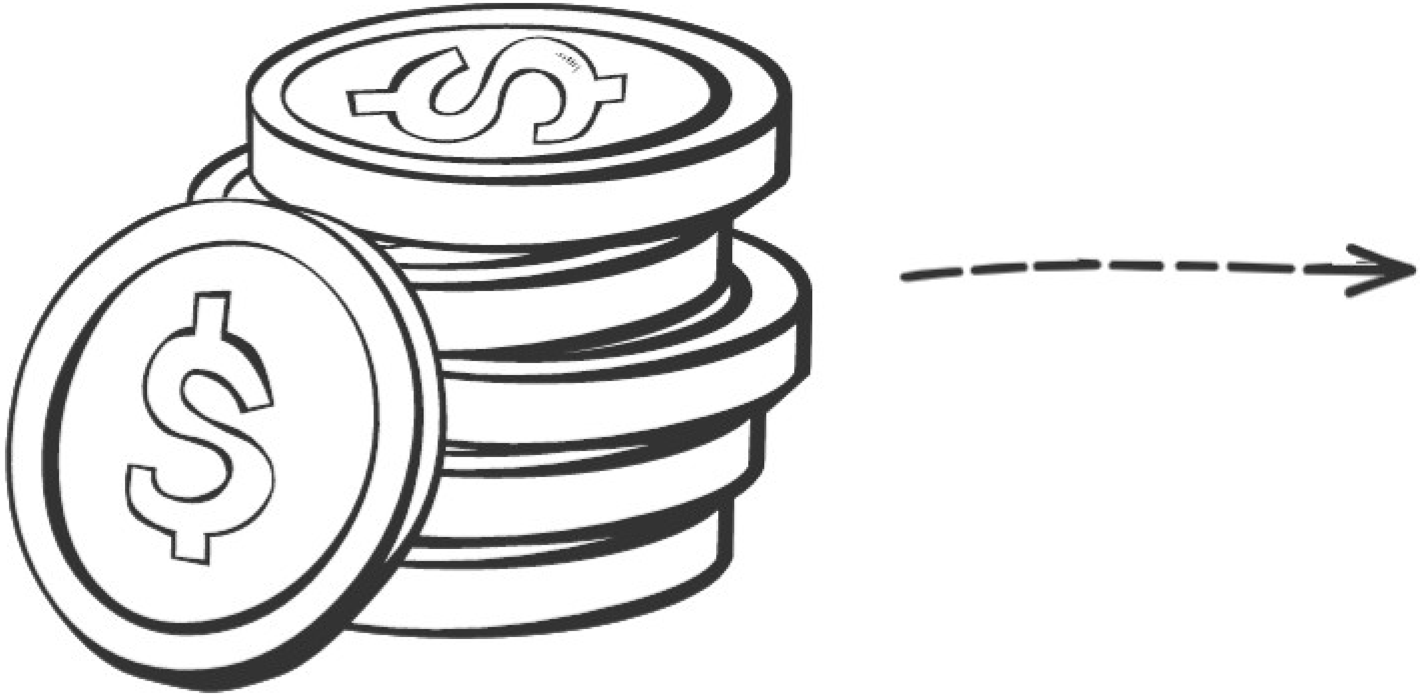
[Ludewig 2003]

Practical Results

- "It works" vs. "It should theoretically work"...
- You can touch the result
- Quality can be measured
- Functionality is only recognized when it has proven itself

What is engineering?

Characteristics of engineering disciplines



[Ludewig 2003]

Cost thinking

- Cost reduction as a goal
- **Does not mean!!!** generate a local optimum → this might have the opposite effect
- Goal: achieve a global cost/benefit ratio

What is engineering?

Characteristics of engineering disciplines

[Ludewig 2003]

Quality awareness

- Basic attitude of an engineer
- Often leads to a reduction in costs

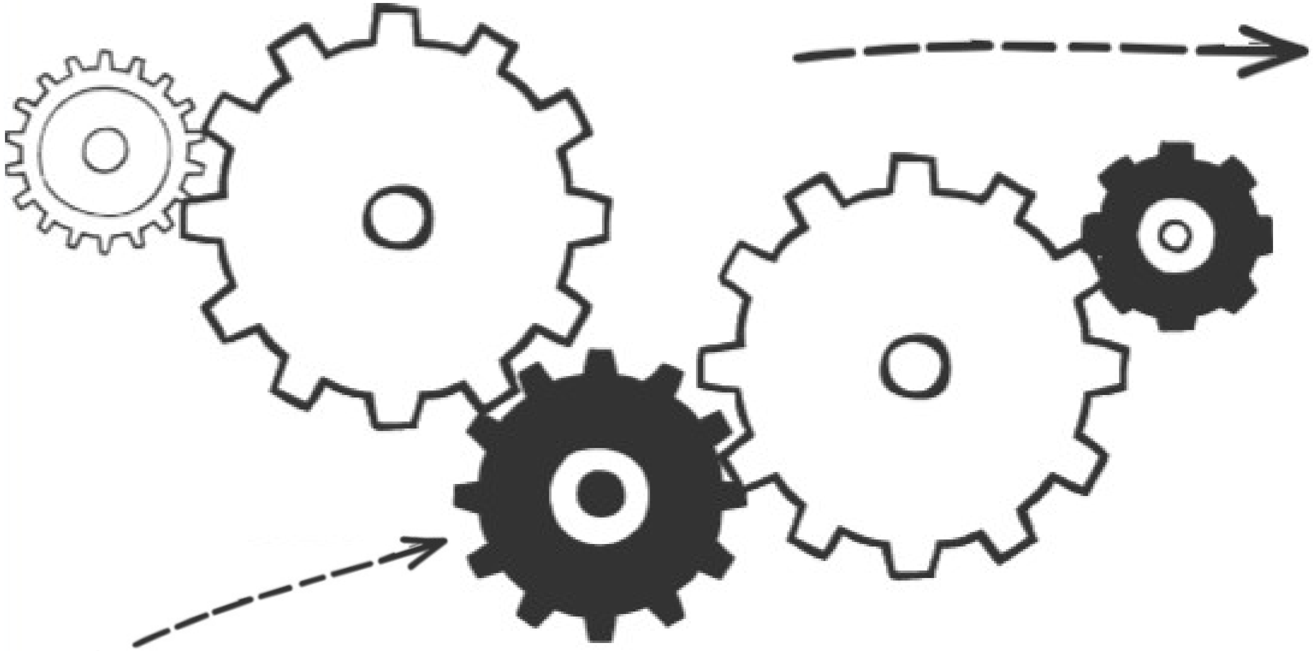
What is engineering?

Characteristics of engineering disciplines

Follow standards with discipline

- Standardized interfaces, terms, communication, etc.
- Dowel pin first standard in Germany (1918)

[Ludewig 2003]



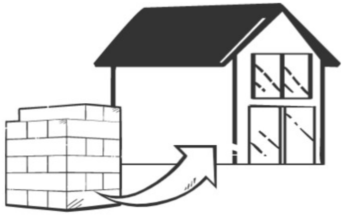
Thinking with assemblies

- Standards as a basis
- "Solved" sub-problems can be incorporated into larger solutions

Known solution, can be integrated

[Ludewig 2003]

What is engineering?
Characteristics of engineering disciplines



Practical Results

- "It works" vs. "It should theoretically work"...
- You can touch the result
- Quality can be measured

Cost thinking

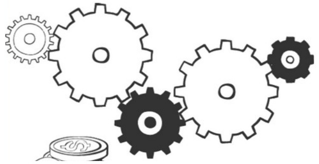
- "Cost reduction as a goal
- **Does not mean!!!** generate a local optimum → this can may the opposite effect
- Goal: achieve a global cost/benefit ratio

Quality awareness

- Basic attitude of an engineer
- Often leads to a reduction in costs

Follow standards with discipline

- Standardized interfaces, terms, communication, etc.



Thinking with assemblies

- Standards as a basis
- "Solved" sub-problems can be incorporated into larger solutions

Task

How can the characteristics be transferred to software engineering? Discuss in groups of two!

Practical Results

- "It works" vs. "It should theoretically work"...
- You can touch the result
- Quality can be measured

Cost thinking

- "Cost reduction as a goal
- Does not mean!!! generate a local optimum → this can may the opposite effect
- Goal: achieve a global cost/benefit ratio

Quality awareness

- Basic attitude of an engineer
- Often leads to a reduction in costs

Follow standards with discipline

- Standardized interfaces, terms, communication, etc.

Thinking with assemblies

- Standards as a basis
- "Solved" sub-problems can be incorporated into larger solutions

What is **software engineering?**

Software

Engineering

SWE?

When did the term software engineering first appear?

1940

1950

1960

1970

1980

1990

2000

2010

When did the term software engineering first appear?

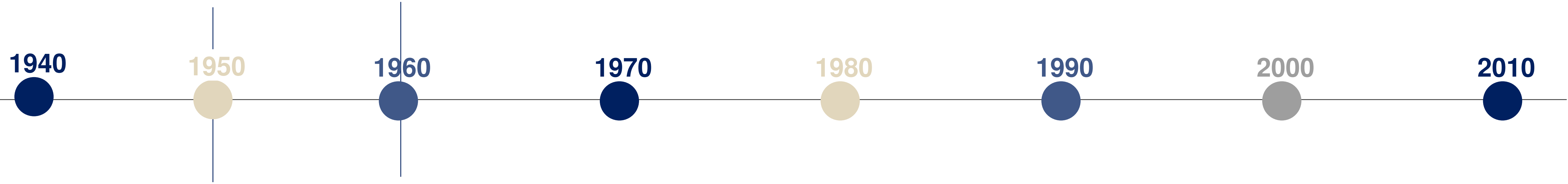
1940
1950
1960
1970
1980
1990
2000
2010

1969

NATO conference in Garmisch Partenkirchen, where F.L. Baur coined the term "software engineering" for the first time.

https://cdn1.vogel.de/unsafe/fit-in/1000x0/images.vogel.de/vogelonline/bdb/1469600/1469671/original.jpg

When did the term software engineering first appear?



1940 1950 1960 1970 1980 1990 2000 2010

- Computers are used more and more in the military sector, software plays a subordinate role.
- Main field of application: Mathematical calculations, FORTRAN is developed for this purpose in 1957.
- With the development of ever faster computers, used by more and more people, more and more complex programs were written.
- Software is becoming increasingly important.

Consequence

More and more errors in increasingly critical areas.

When did the term software engineering first appear?

1940 1950 1960 1970 1980 1990 2000 2010

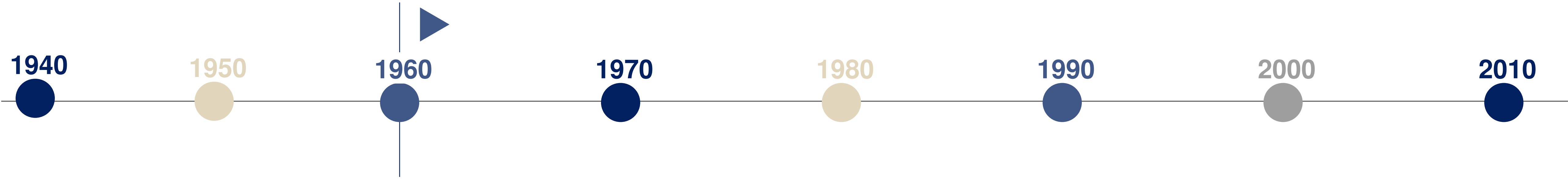
1962

Launch of the Mariner 1 rocket, had to be detonated shortly after launch.

Typo in the program code caused the rocket to veer off course.

https://de.wikipedia.org/wiki/Mariner#/media/Datei:Atlas_Agena_with_Mariner_1.jpg

When did the term software engineering first appear?



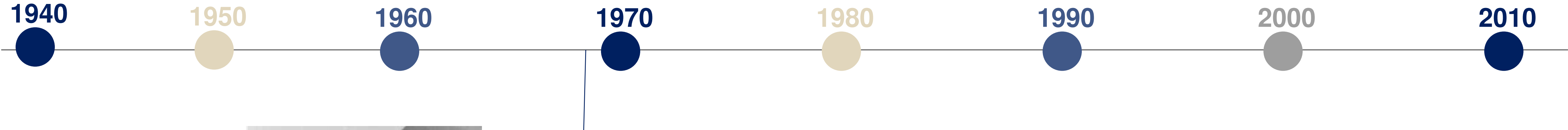
The errors accumulate and lead to considerable damage.

⇨ Software crisis

https://de.wikipedia.org/wiki/Mariner#/media/Datei:Atlas_Agena_with_Mariner_1.jpg

> *As long as there were no machines, programming was not an existing problem; when we had a few weak computers, programming became a minor problem, and now that we have gigantic computers, programming is an equally gigantic problem.*
>
> Edsger W. Dijkstra

When did the term software engineering first appear?



1940 1950 1960 1970 1980 1990 2000 2010

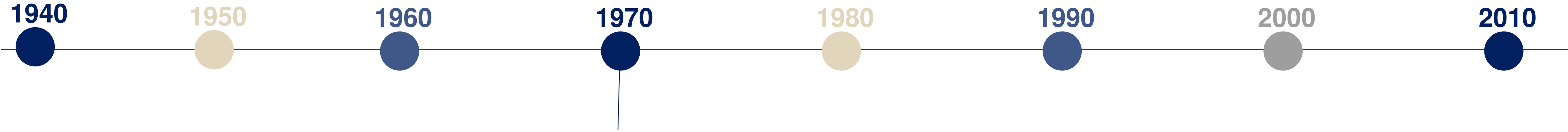
1969

F.L. Baur coined the term "software engineering" for the first time.

https://cdn1.vogel.de/unsafe/fit-in/1000x0/images.vogel.de/vogelonline/bdb/1469600/1469671/original.jpg

> *The whole trouble comes from the fact that there is so much tinkering with software. It is not made in a clean fabrication process, which it should be. What we need is software engineering.*
>
> F. L. Baur

When did the term software engineering first appear?



1940

1950

1960

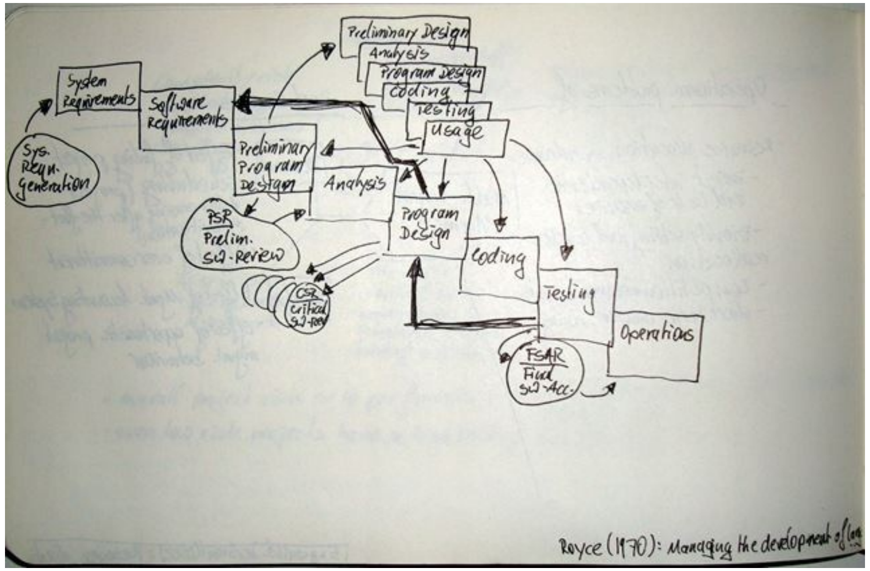
1970

1980

1990

2000

2010



https://alchetron.com/Winston-W-Royce

1970

Winston W. Royce, The waterfall model

Some definitions

Software engineering *is the discovery and application of sound engineering principles with the goal of economically producing software that is reliable and runs on real computers.*
F.L. Bauer

Software engineering *is any activity that involves the creation or modification of software, insofar as the software pursues objectives that go beyond the software itself.*
Ludewig (2001)

IEEE definition

Software Engineering
(1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.
2) The study of approaches as in (1).

IEEE Std. 610.12 (1990)

Software engineering *is the systematic, disciplined and quantifiable approach to the development, operation and maintenance of software, in short: working on software according to engineering principles.*

What is software engineering?
Our definition

Software engineering *comprises the professional, engineering-style (ongoing) development and operation of*

(1) higher quality
(2) large program systems
(3) with many participants
(4) under cost and time constraints.

[based on Paech 2021]

Software Engineering - Basics

- What is Software Engineering?
- Software and Engineering
- Disciplines and Challenges
- Project-Based vs. Product-Based SWE
- Further Terms

Learning Objectives

- You know the most important disciplines in SWE and can explain their main purpose.
- You can assign the activities in the software development process to the associated design decisions, development results and test stages.

Question

Which SWE disciplines do you know?

Disciplines in software engineering

Basic topics

Configuration management | Documentation |
Knowledge management | People in the SWE process and digital ethics | Tools

Development

Requirements
- Context analysis
- Requirements Engineering

Design
- Course granular design (architecture)
- Detailed design

Implementation

QualityMgt.

Quality assurance and testing
- Test, inspection, metrics

Processes and process models
- Improvement, process model, maturity levels

Evolution

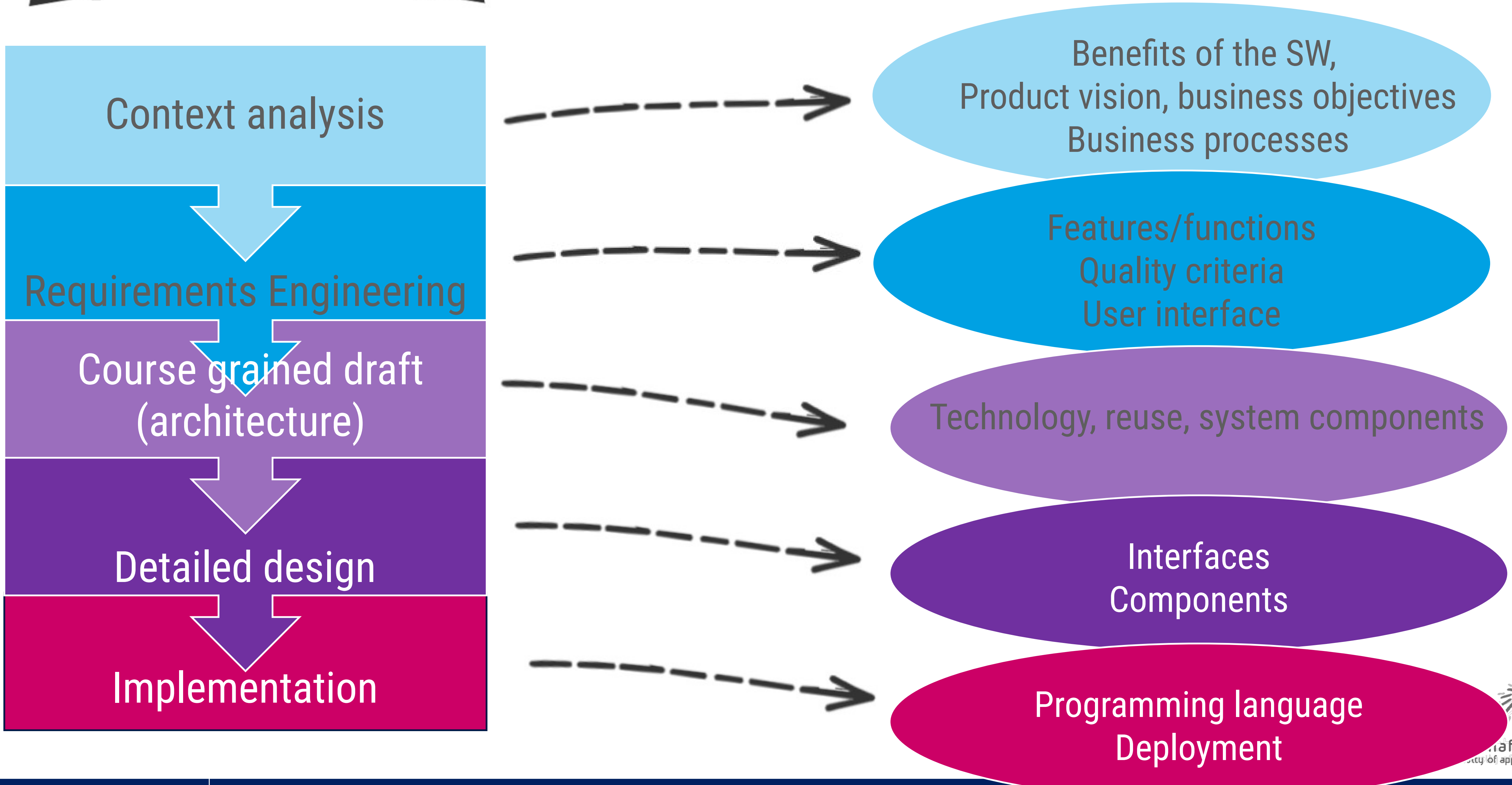
- Roll-Out
- Operation
- Maintenance
- Further development
- Reuse
- Reengineering
- Refactoring
- Change management

Management

- Strategy
- Economy
- Team
- Dates
- Risks
- Customer, client/contractor
- Innovation

Disciplines

Design decisions



Context analysis

Requirements Engineering

Course grained draft (architecture)

Detailed design

Implementation

Benefits of the SW,
Product vision, business objectives
Business processes

Features/functions
Quality criteria
User interface

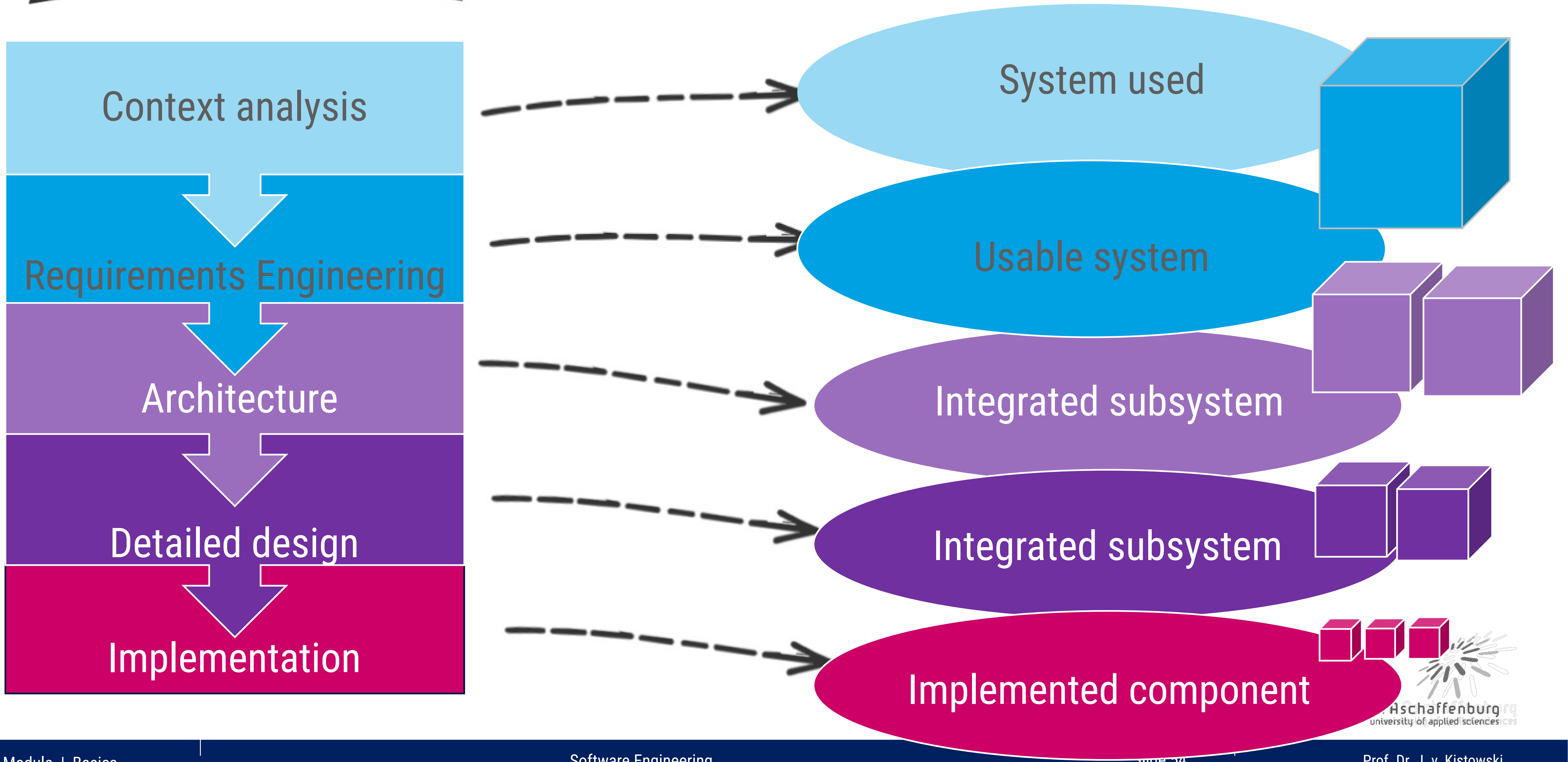
Technology, reuse, system components

Interfaces
Components

Programming language
Deployment

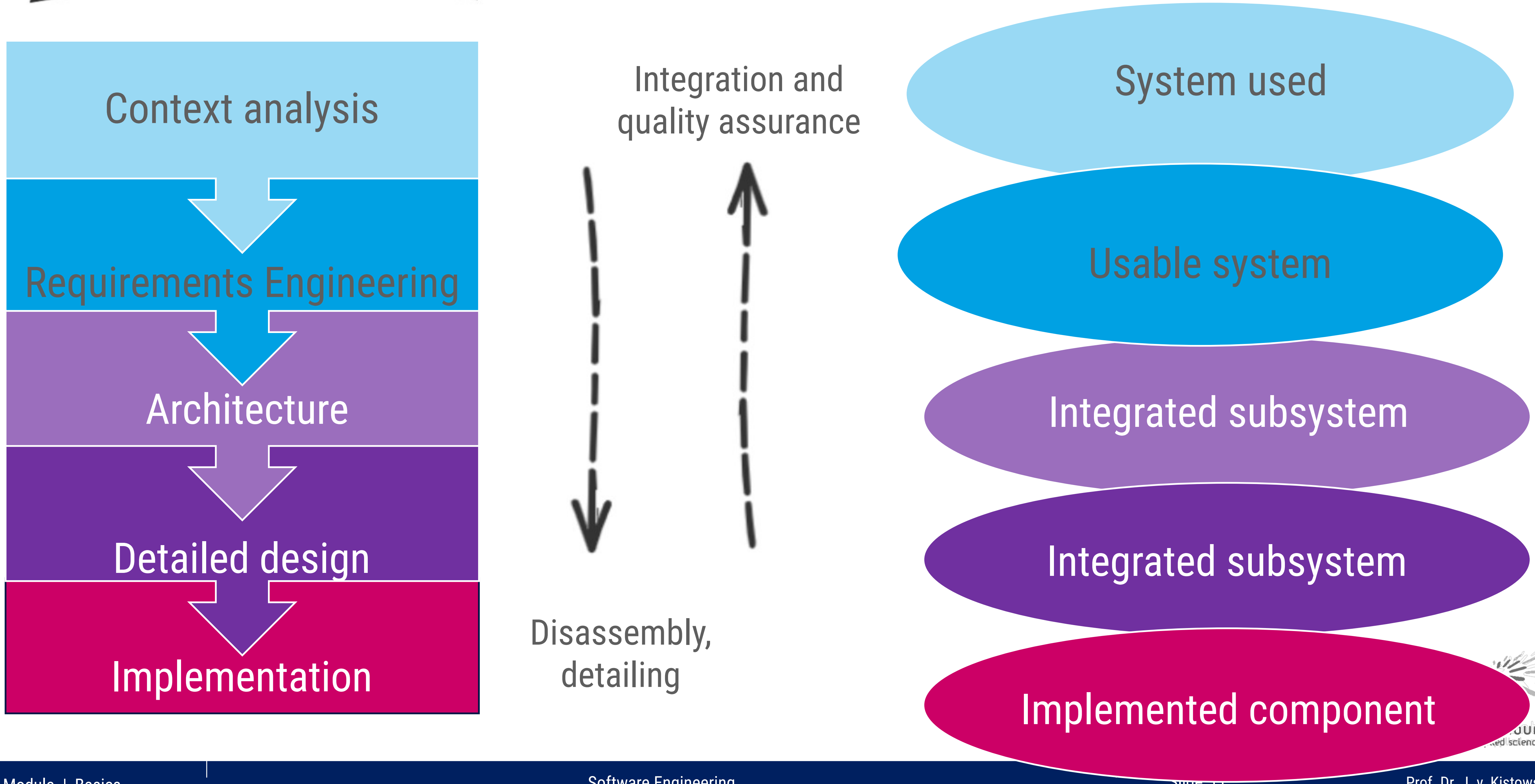
Disciplines

Results of the development



Disciplines

Results of the development



Context analysis

Requirements Engineering

Architecture

Detailed design

Implementation

Integration and quality assurance

Disassembly, detailing

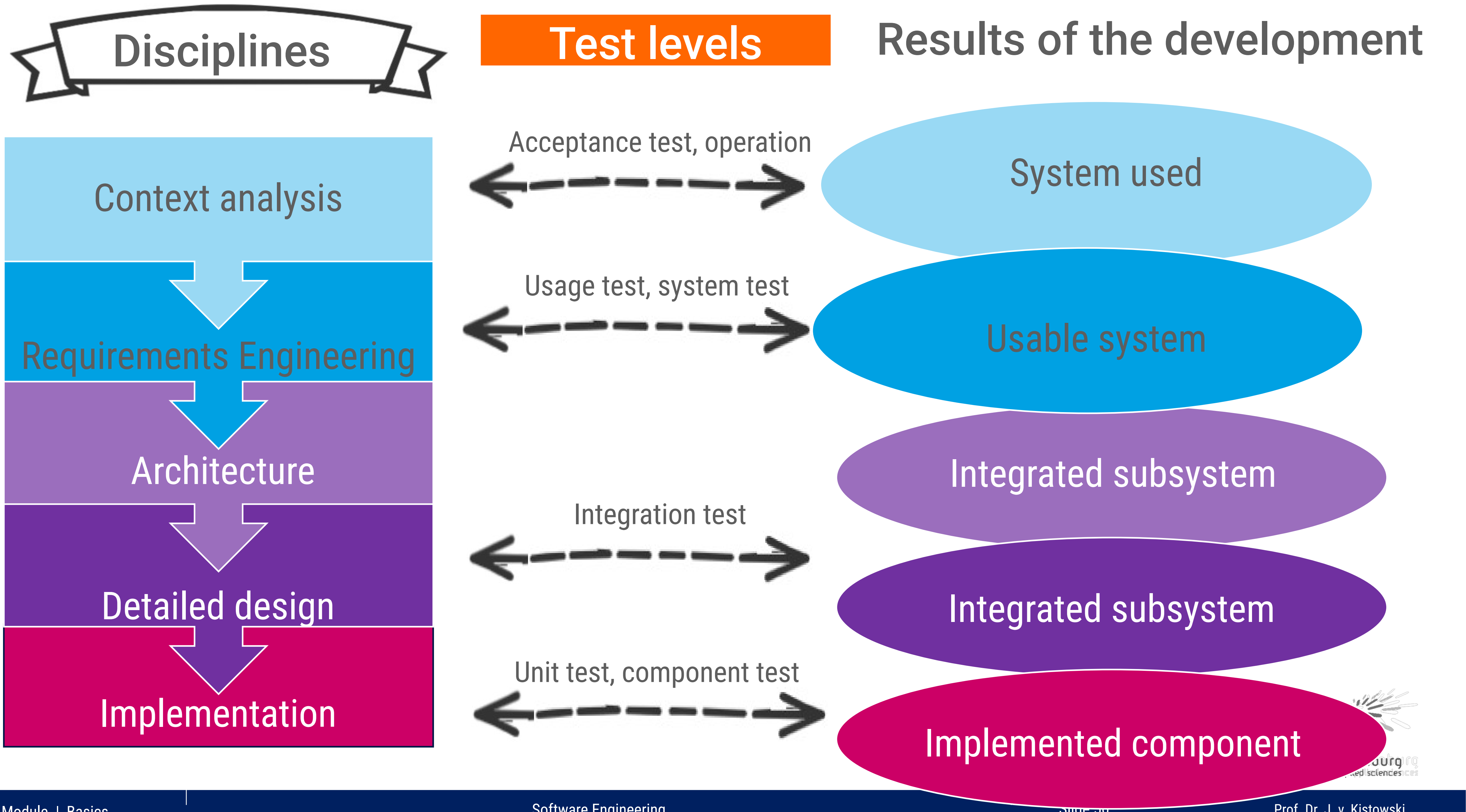
System used

Usable system

Integrated subsystem

Integrated subsystem

Implemented component



Disciplines

Test levels

Results of the development

Context analysis

Requirements Engineering

Architecture

Detailed design

Implementation

Acceptance test, operation

Usage test, system test

Integration test

Unit test, component test

System used

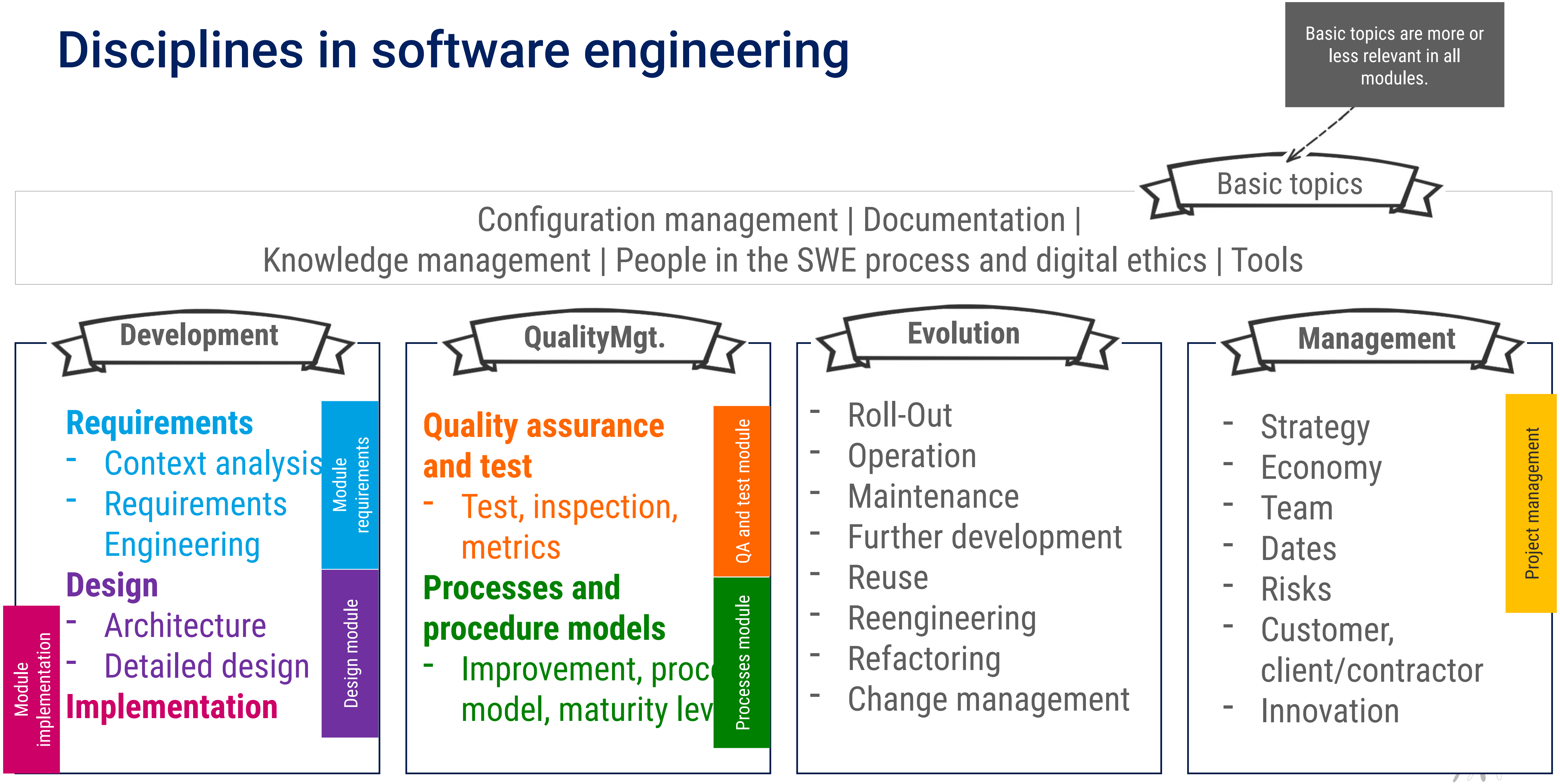
Usable system

Integrated subsystem

Integrated subsystem

Implemented component

Disciplines in software engineering



Basic topics are more or less relevant in all modules.

Basic topics

Configuration management | Documentation |
Knowledge management | People in the SWE process and digital ethics | Tools

Development

Requirements
- Context analysis
- Requirements Engineering

Design
- Architecture
- Detailed design

Implementation

QualityMgt.

Quality assurance and test
- Test, inspection, metrics

Processes and procedure models
- Improvement, proc model, maturity lev

Evolution

- Roll-Out
- Operation
- Maintenance
- Further development
- Reuse
- Reengineering
- Refactoring
- Change management

Management

- Strategy
- Economy
- Team
- Dates
- Risks
- Customer, client/contractor
- Innovation

Question

What SWE roles/responsibilities do you know?

- **Software** is not "just" code; it also includes documentation and data relating to the operation of computer systems.
- Characteristics of traditional engineering disciplines cannot simply be transferred to software.
- SWE is a comparatively young discipline that originated in the 1970s. It emerged as a response to the **software crisis** that was declared at the time.
- **SWE** comprises the professional, engineering-style (further) development and operation of high-quality, large program systems with many participants under cost and time constraints.
- Different **disciplines** have emerged that focus on human, methodological, economic or even ethical aspects.

TH Aschaffenburg
university of applied sciences

Software Engineering - Basics

What is Software Engineering?

Software and Engineering

Disciplines and **Challenges**

Project-Based vs. Product-Based SWE

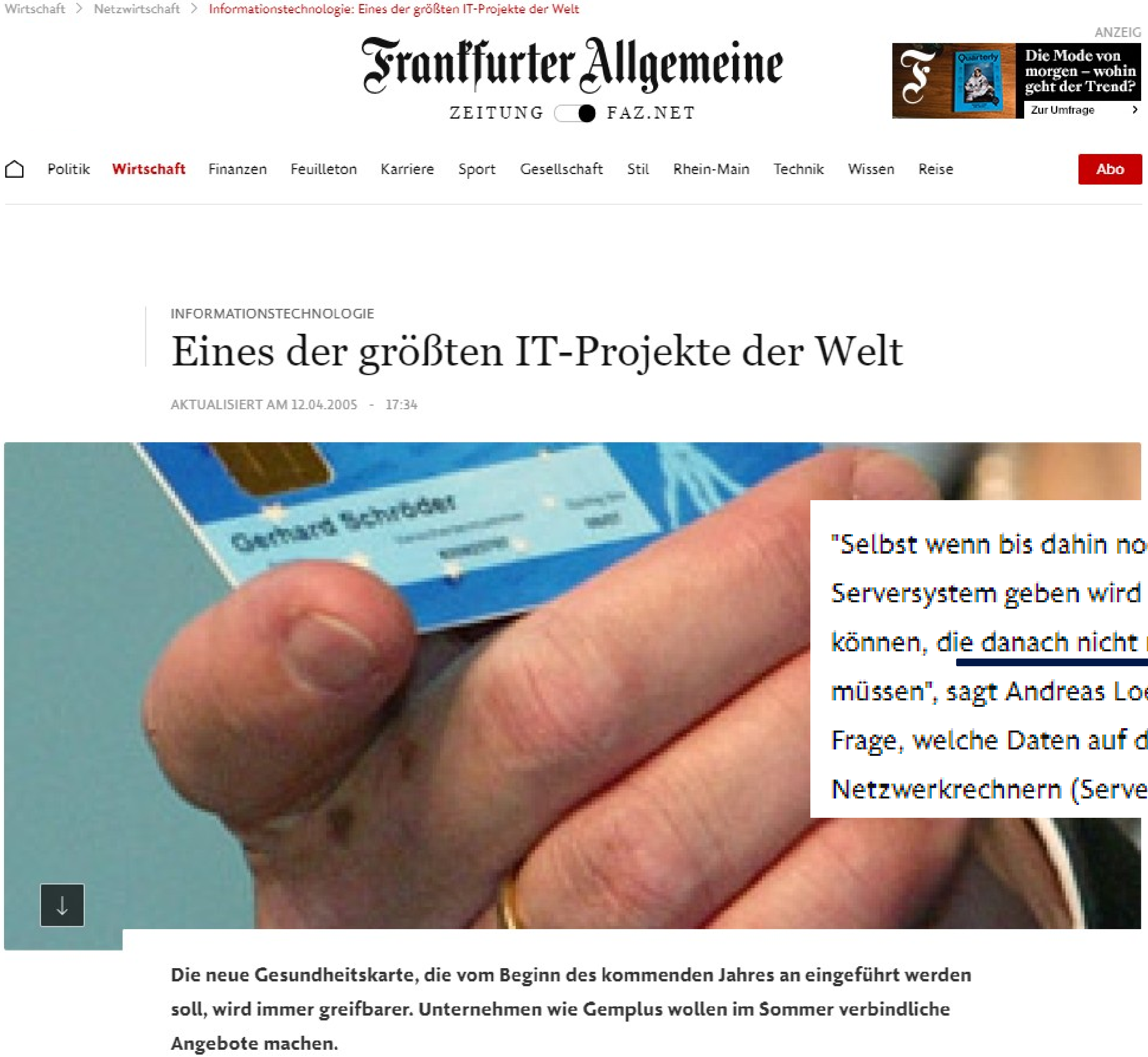
Further Terms

Learning Objectives

- You can use examples to explain why software engineering or software development is **challenging**.
- You can explain why **complexity** is a problem.

Challenges in software engineering:

1. basics and communication



Wirtschaft > Netzwirtschaft > Informationstechnologie: Eines der größten IT-Projekte der Welt

Frankfurter Allgemeine

ZEITUNG FAZ.NET

INFORMATIONSTECHNOLOGIE

Eines der größten IT-Projekte der Welt

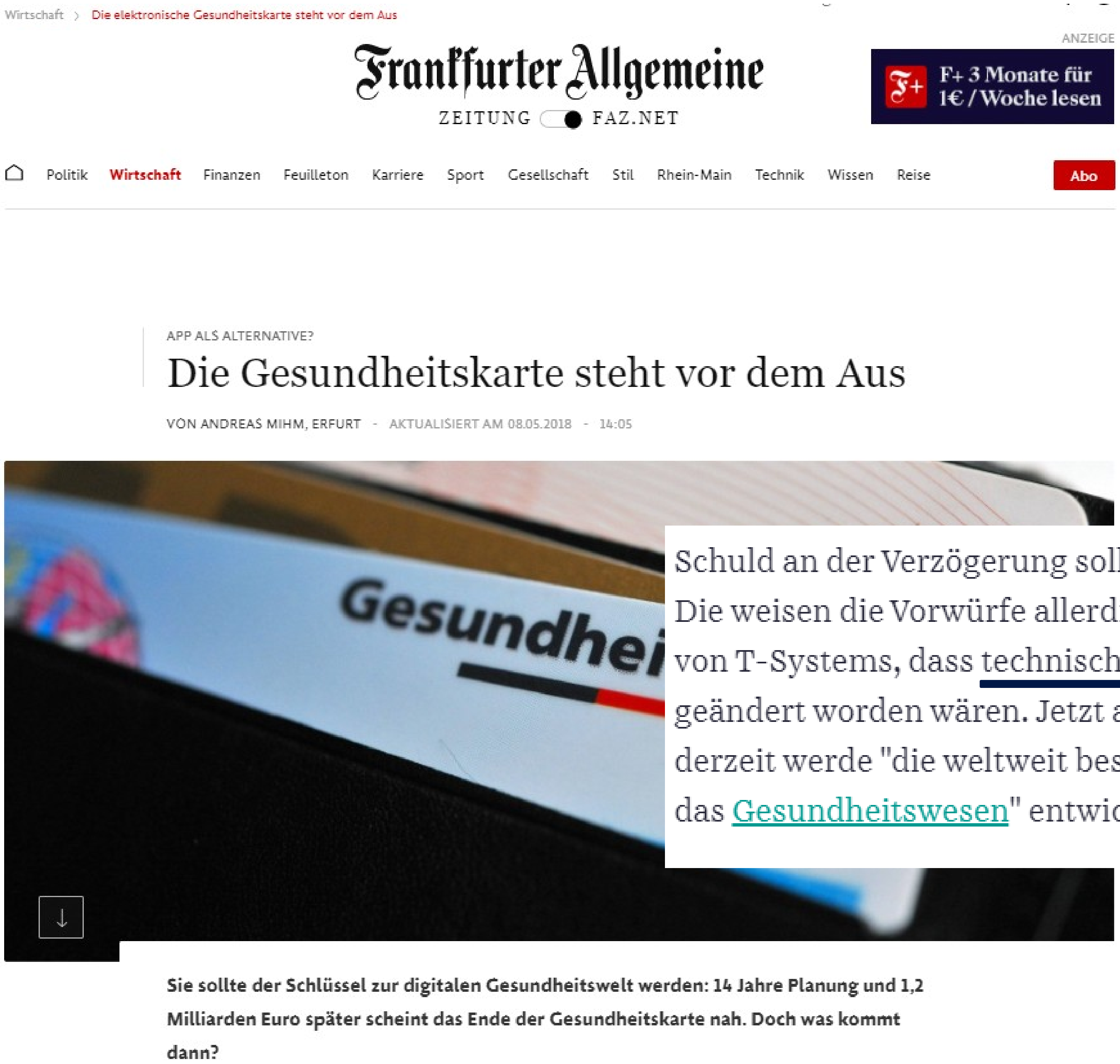
AKTUALISIERT AM 12.04.2005 - 17:34

Die neue Gesundheitskarte, die vom Beginn des kommenden Jahres an eingeführt werden soll, wird immer greifbarer. Unternehmen wie Gemplus wollen im Sommer verbindliche Angebote machen.

"Selbst wenn bis dahin noch immer nicht feststehen sollte, ob es ein Serversystem geben wird oder nicht, werden wir Angebote abgeben können, die danach nicht mehr nennenswert verändert werden müssen", sagt Andreas Loesch von Gemplus. Dabei geht es um die Frage, welche Daten auf den Karten selbst und welche auf zentralen Netzwerkrechnern (Servern) gespeichert werden sollen. Allzu großen

„We will be able to make an offer that won’t have to be changed much afterwards“

https://www.faz.net/aktuell/wirtschaft/netzwirtschaft/informationstechnologie-eines-der-groessten-it-projekte-der-welt-1228267.html



APP ALS ALTERNATIVE?

Die Gesundheitskarte steht vor dem Aus

VON ANDREAS MIHM, ERFURT - AKTUALISIERT AM 08.05.2018 - 14:05

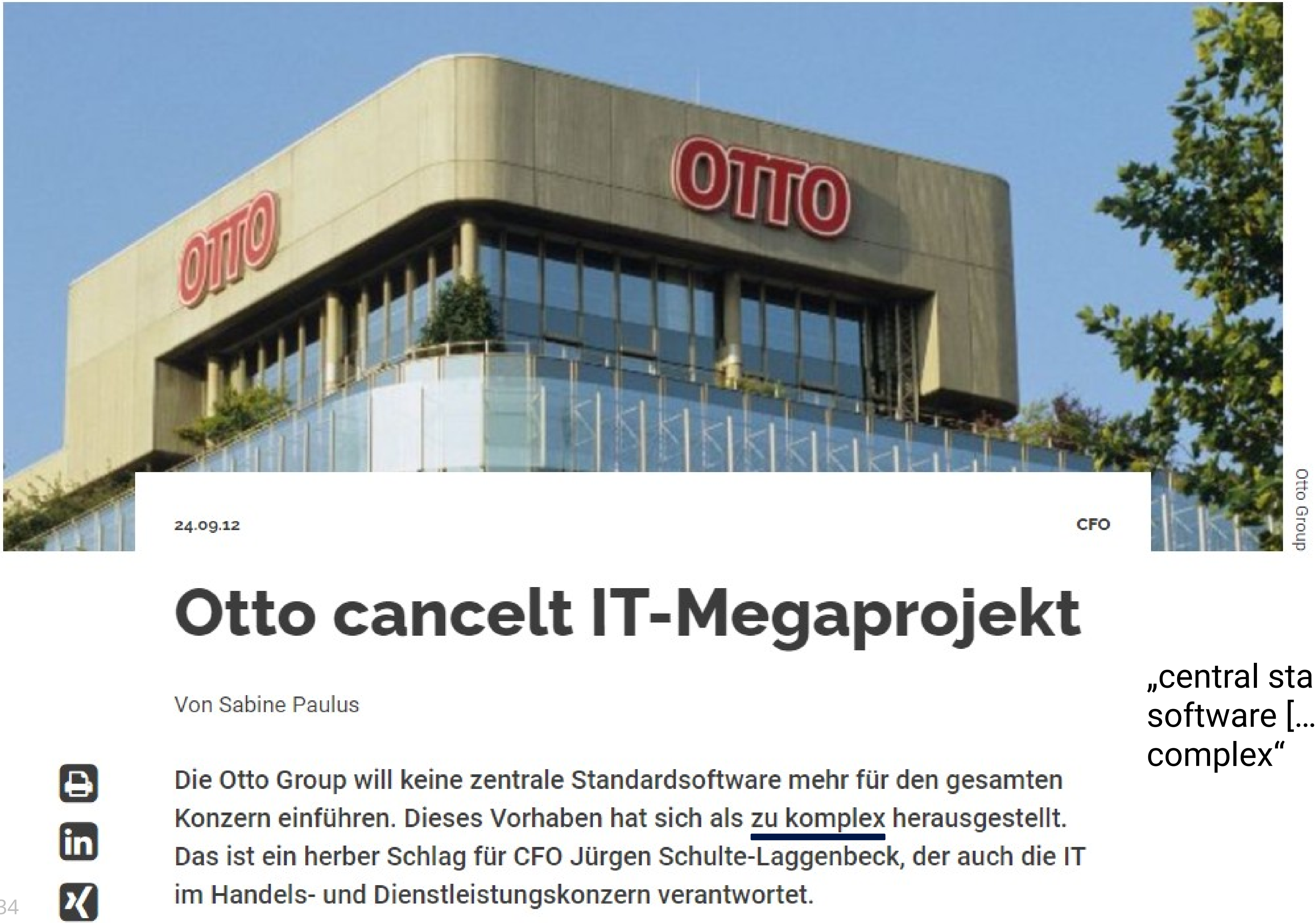
Sie sollte der Schlüssel zur digitalen Gesundheitswelt werden: 14 Jahre Planung und 1,2 Milliarden Euro später scheint das Ende der Gesundheitskarte nah. Doch was kommt dann?

Schuld an der Verzögerung sollen die beteiligten Industriefirmen sein. Die weisen die Vorwürfe allerdings zurück. So sagt etwa ein Sprecher von T-Systems, dass technische Anforderungen mehr als 150 Mal geändert worden wären. Jetzt aber sei die Industrie "auf der Zielgeraden, derzeit werde "die weltweit bestgeschützte öffentliche Infrastruktur für das Gesundheitswesen" entwickelt.

„techical requirements changed more than 150 times“

https://www.faz.net/aktuell/wirtschaft/die-elektronische-gesundheitskarte-steht-vor-dem-aus-15578934.html

https://www.sueddeutsche.de/wirtschaft/e-card-elektronische-gesundheitskarte-offenbar-vor-dem-aus-1.3617842



24.09.12

CFO

Otto Group

Otto cancelt IT-Megaprojekt

Von Sabine Paulus

Die Otto Group will keine zentrale Standardsoftware mehr für den gesamten Konzern einführen. Dieses Vorhaben hat sich als zu komplex herausgestellt. Das ist ein herber Schlag für CFO Jürgen Schulte-Laggenbeck, der auch die IT im Handels- und Dienstleistungskonzern verantwortet.

„central standard software […] too complex“

https://www.finance-magazin.de/cfo/cfo-digital/otto-cancelt-it-megaprojekt-1234901/

Challenges in software engineering:

2. complexity

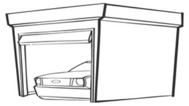
The challenges
2. complexity

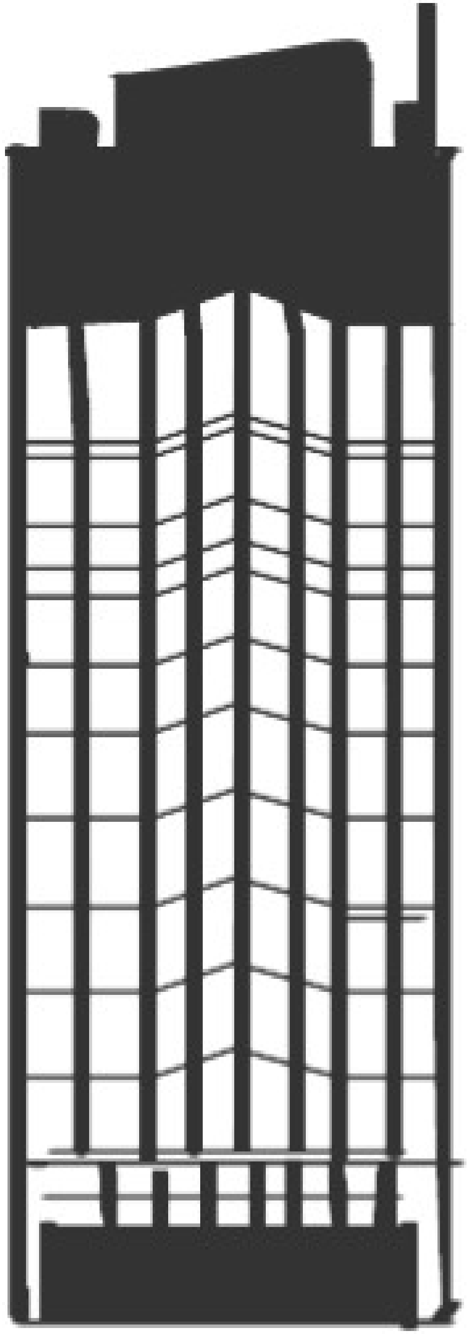
> *The big is not a multiple of something small, but something completely different.*
>
> [Ludewig, 2013]

The challenges
2. complexity

"The big is not a multiple of something small but something completely different." [Ludewig, 2013]

Building a high-rise building is not the same as building a carport 100 times over. Instead it is something completely different. The requirements, effort and construction are completely different.

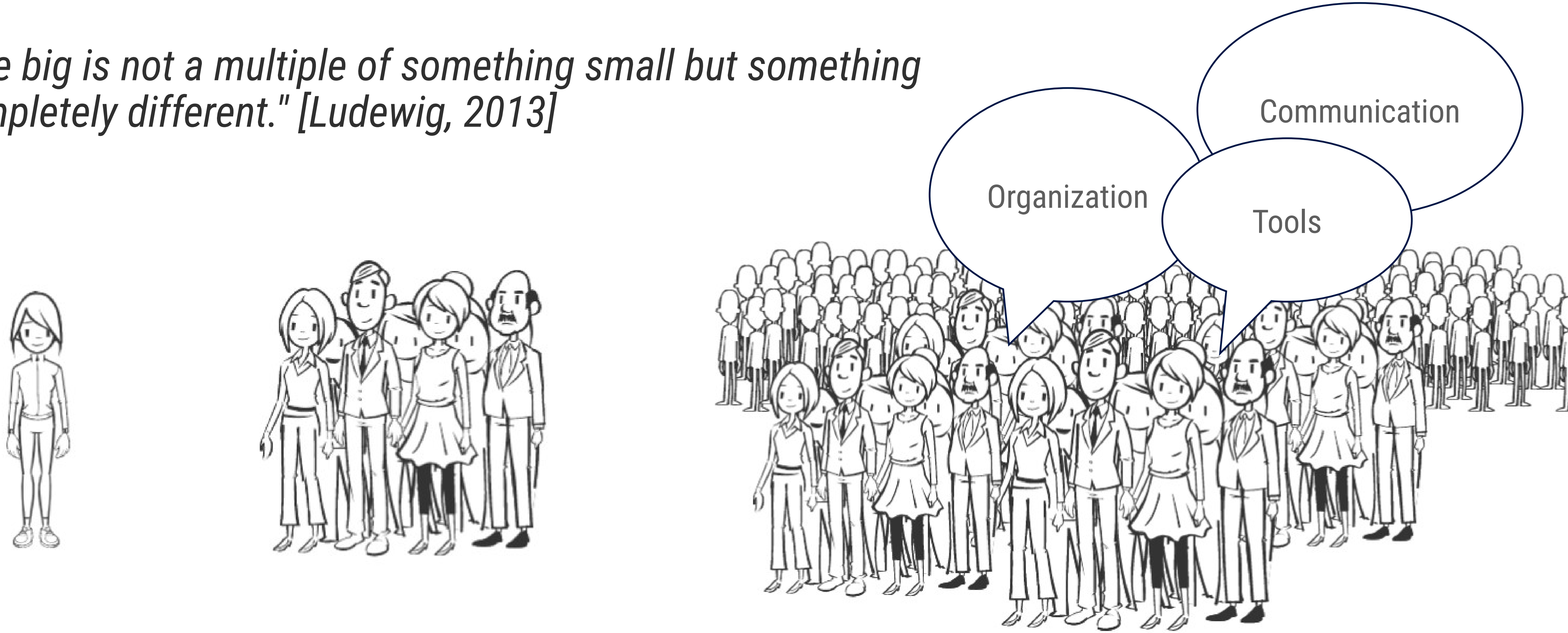




TH Aschaffenburg
university of applied sciences

The challenges
2. complexity

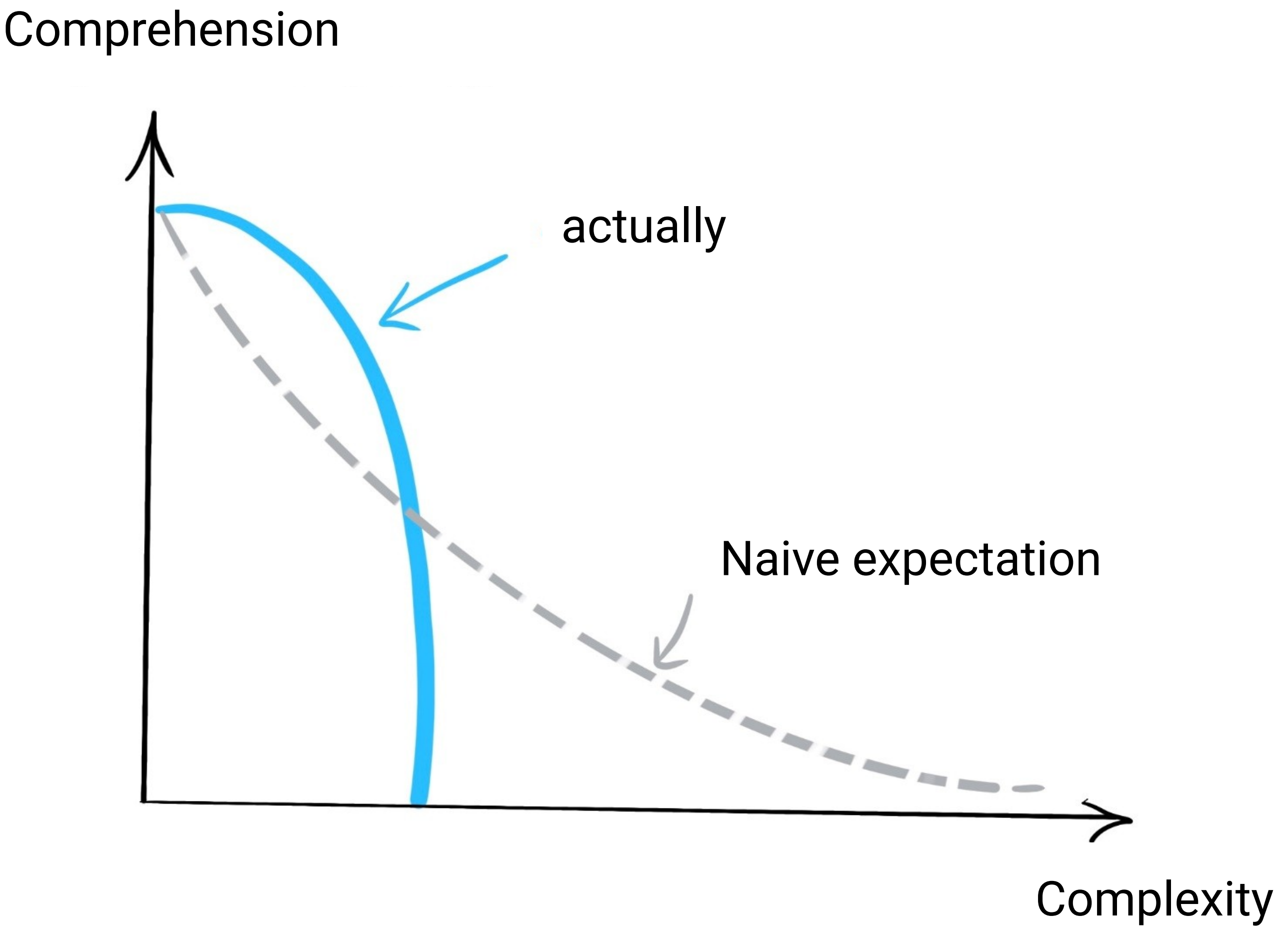
"The big is not a multiple of something small but something completely different." [Ludewig, 2013]



The challenges
2. complexity

> *Solutions for small problems cannot be scaled for large problems.*
>
> [Ludewig, 2013]

The challenges
2. complexity



According to [Ludewig, 2013]

Challenges in software engineering:

3. global optimum

TH Aschaffenburg
university of applied sciences

Challenges
3. global optimum

- The goal is to achieve a global optimum
- Does not mean that it is sufficient to achieve optimization in the individual areas.
  - Example: We want to optimize costs and omit the test. A quality problem will probably arise, not a global optimum.
  - Example: We want to optimize performance. Maintainability will probably suffer as a result, etc.
- A global optimum means looking at the context from a holistic perspective. And this is neither easy nor "done quickly". But it is necessary in order to avoid unfavorable local optima.

Challenges
3. global optimum

> *The chain is only as strong as its weakest link.*
>
> If a single discipline is neglected, the software engineering chain breaks.
>
> Anonymous

- Given that software engineering is a young discipline, the importance of software is growing rapidly.
- The challenges of manufacturing high-quality products are high.
- Communication and the quality of the foundations are important success factors for software development projects.
- Further challenges are
  - **Complexity** of the domain to be mapped (solutions do not scale with size)
  - finding a **global optimum**.

What is Software Engineering?

Software and Engineering

Disciplines and **Challenges**

Project-Based vs. Product-Based SWE

Further Terms

Question

Project-based vs. product-based SWE
What could be the difference?

Learning Objective

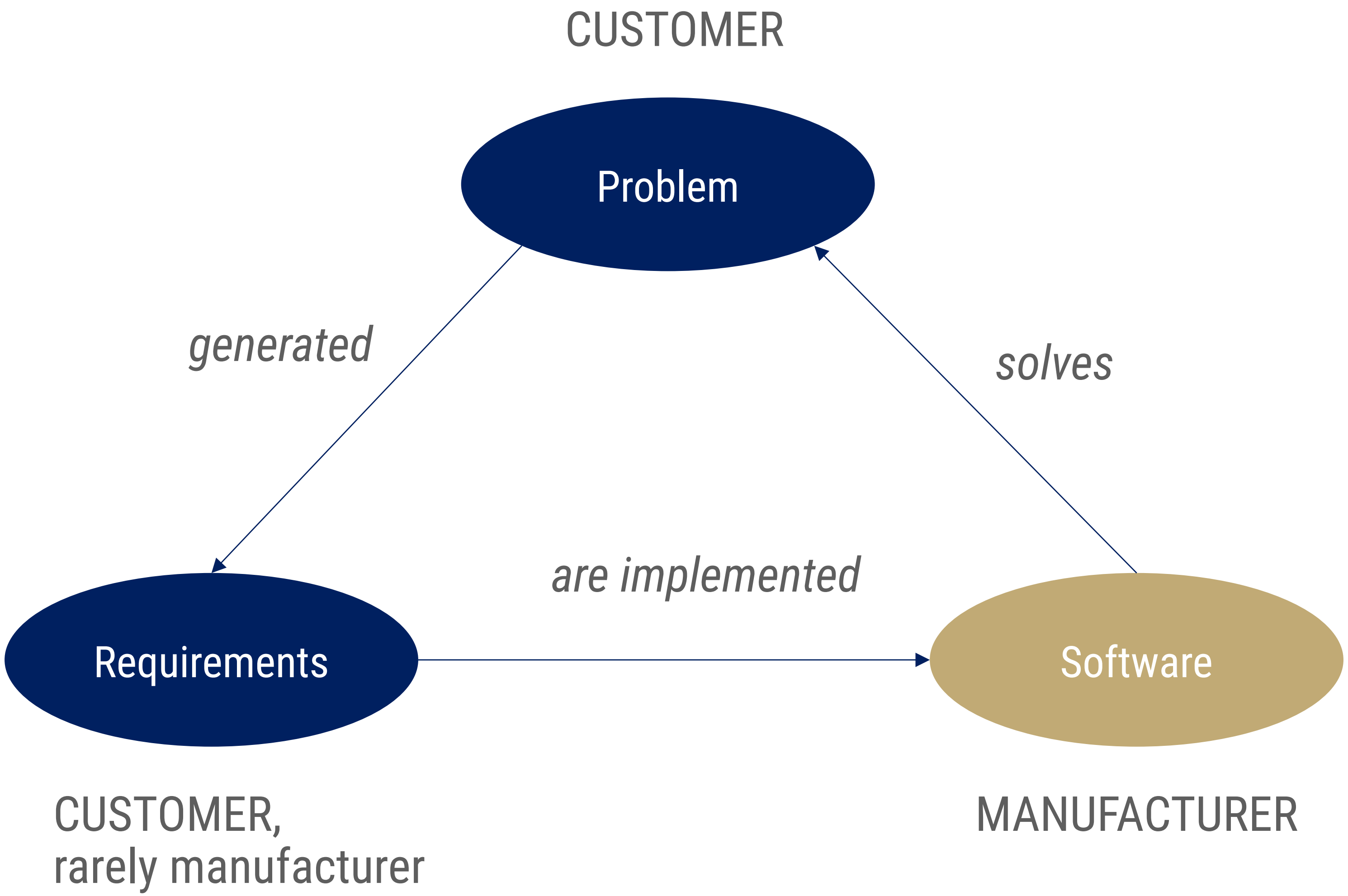
- You can explain the differences between product-based and project-based SWE.

Project-based software engineering

Dominated software engineering since the 70s

- Origin: Development of **customized software**
- There is a **client/customer** who decides on the functionality of the system. For an external customer: legally binding contract.
- **Requirements** are implemented on the basis of the customer's problem. If "the problem" changes, the software must be adapted.
- Customer pays for the software and its further development.
- In most cases, the software has a long service life.

Project-based software engineering



CUSTOMER

Problem

generated

solves

are implemented

Requirements

Software

CUSTOMER,
rarely manufacturer

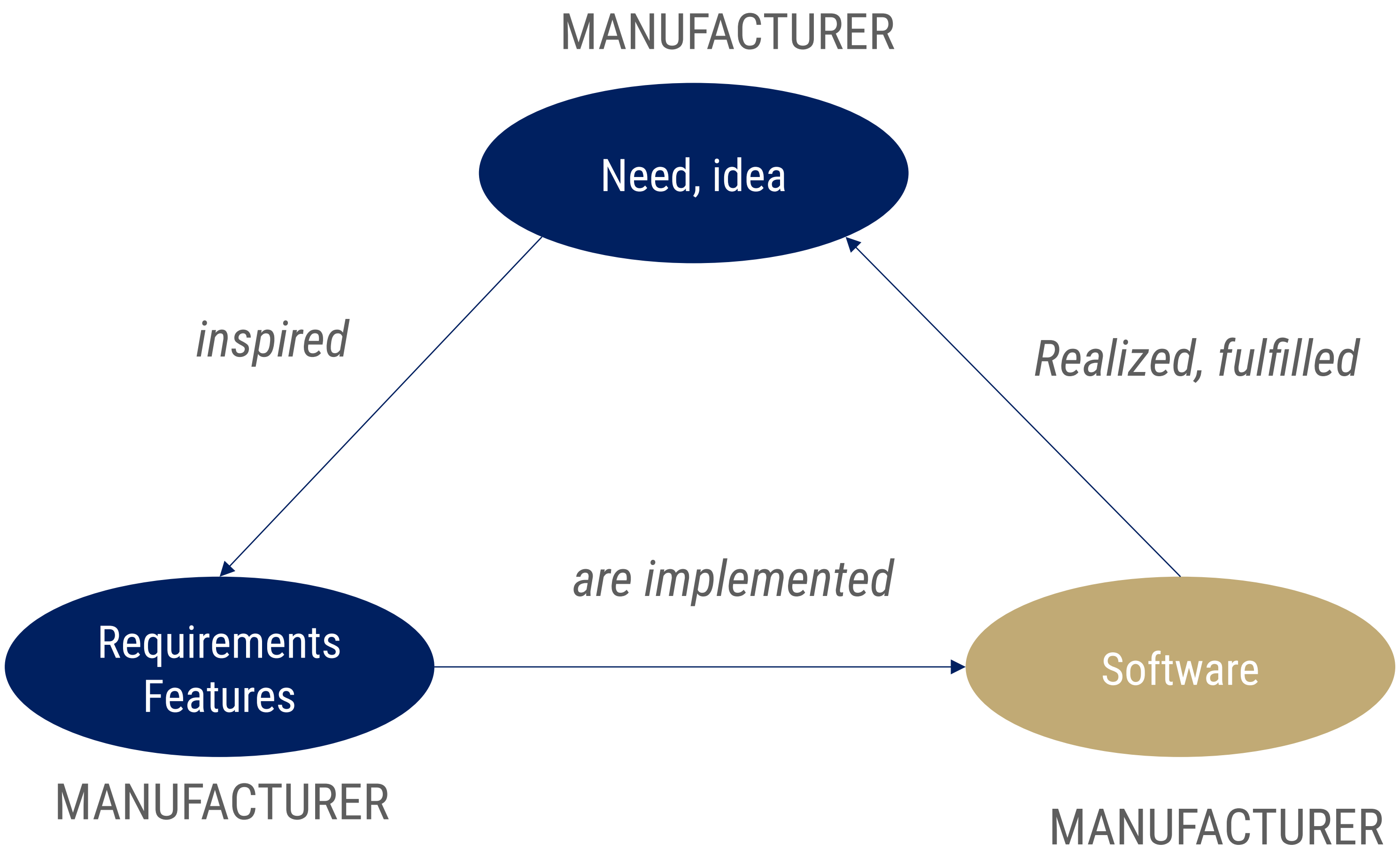
MANUFACTURER

Product-based software engineering

More and more standard software leads to new approaches in software engineering

- Origin: Development of **standard software, e.g. apps**
- There are many users, but no **client/customer**. Software manufacturer decides on product strategy, features, platforms, etc.
- **Requirements** arise from the potential needs of future users.
- **Development costs** are spread across a large group of users. Customization costs may arise because the standard solution does not fit.
- The manufacturer decides on **service life and further development** (when which content is added or even when product development is discontinued).
- Rapid market launch can be a success factor. (Excellent products can fail if a lower quality product is launched earlier). Users are less likely to switch if they have invested time and/or money in the first product.

Product-based software engineering



What is Software Engineering?

Software and Engineering

Disciplines and **Challenges**

Project-Based vs. Product-Based SWE

Further Terms

Further terms relating to software

Who am I?

I represent a section of the software. Depending on the context, I abstract from certain details. Examples: Specifications, diagrams, also metrics (key figures).

I am a tangible piece of information that is created, modified and used by people when they perform activities. I can be a model, a model element or a document. Examples Document, e.g. requirements specification, model, e.g. object-oriented analysis model, source code, e.g. C++ program.

I am there to carry out work that - at least in principle - could be done without me. However, I am no longer included in the end result.

Further terms relating to software

I represent a section of the software. Depending on the context, I abstract from certain details. Examples: Specifications, diagrams, also metrics (key figures).
Model

I am a tangible piece of information that is created, modified and used by people when they perform activities. I can be a model, a model element or a document. Examples Document, e.g. requirements specification, model, e.g. object-oriented analysis model, source code, e.g. C++ program. ***Artifact***

I am there to carry out work that - at least in principle - could be done without me. However, I am no longer included in the end result. ***Tool***

Taxonomy

A scheme that <u>partitions a body of knowledge</u> and <u>defines the relationships among the pieces</u>. It is used for classifying and understanding the body of knowledge.

IEEE Std. 610.12 (1990)

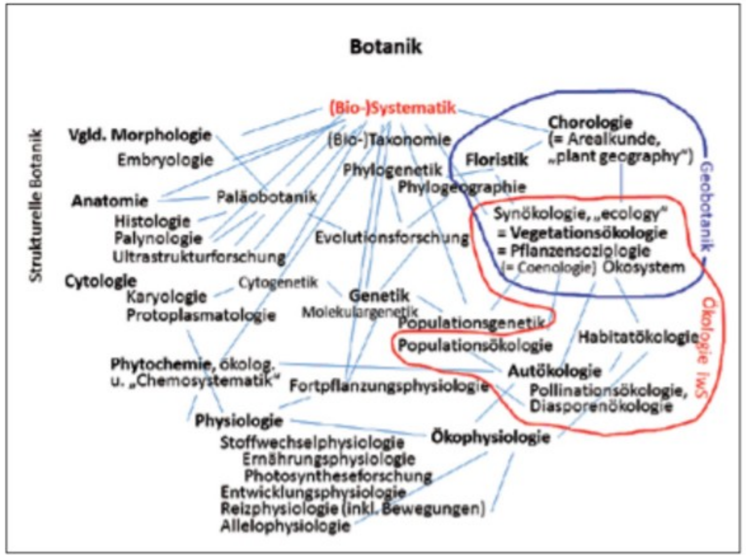
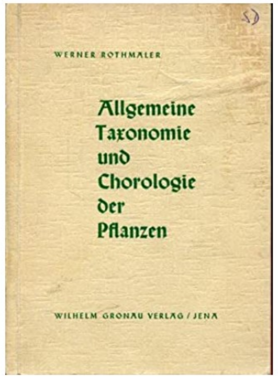


Abb. 3:
Überblick über die Teildisziplinen der wissenschaftlichen Botanik.

https://www.zobodat.at/pdf/CAR_203_123_0349-0428.pdf



WERNER ROTHMALER

Allgemeine
Taxonomie
und
Chorologie
der
Pflanzen

WILHELM GRONAU VERLAG / JENA

General taxonomy and chorology of plants. Basics of special botany.

<u>**Rothmaler, Werner**</u>

Publisher: Jena. Published by Wilhelm Gronau. 1950., 1950

Question

Why are taxonomies and definitions of terms particularly important in SWE?

Why are taxonomies and definitions of terms particularly important in SWE?

- Tool and end product are "LANGUAGE".
- Language is ambiguous by definition.
- This makes it all the more important to define terms and use them consistently!

TH Aschaffenburg
university of applied sciences

> *ALWAYS create a glossary to define terms and develop a common understanding within the team.*

Core principle of „Domain Driven Design”

TH Aschaffenburg
university of applied sciences

Questions?

TH Aschaffenburg
university of applied sciences

What were the headlines of the lecture?

Slip into the role of a sensationalist reporter. What are the headlines that summarize today's lecture? Write along!

Literature

- [Ludewig 2013] Ludewig, Lichter: Software Engineering. Grundlagen, Menschen, Prozesse, Techniken, dpunkt.verlag.
- [Paech 2021] Barbara Paech: Lecture Software Engineering, Uni-Heidelberg.

Thank you for your attention!

Software Engineering

Prof. Dr. J. v. Kistowski

TH Aschaffenburg
university of applied sciences