Software Engineering

Requirements Engineering I

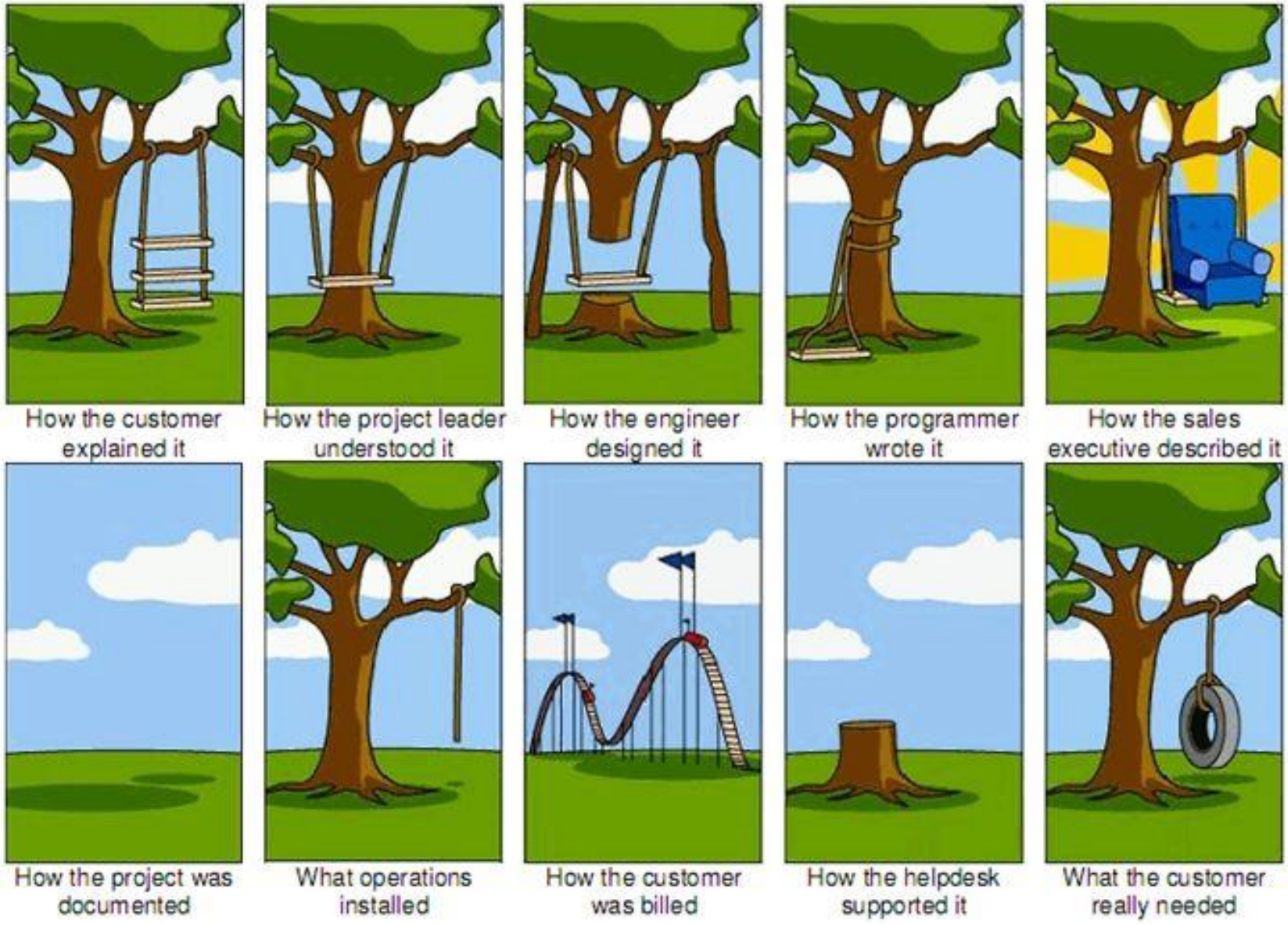
Software & Systems Engineering | Prof. Dr. Andreas Vogelsang | 16.10.2023

@andivogelsang
vogelsang@cs.uni-koeln.de

Learning Goals for Today

- Understand what requirements are and what types exist
- Understand how requirements can be structured in documents
- Understand the most important concepts of RE
- Understand the main challenges of RE in practice

The Requirements Engineering Problem



Universität zu Köln

What are Requirements?

What is a Requirement?

1. A need perceived by a stakeholder.
2. A capability or property that a system shall have.
3. A documented representation of a need, capability or property.

Give some examples of requirements
for a web search engine

M. Glinz: A Glossary of Requirements Engineering Terminology, v1.6, IREB, 2014

User/Stakeholder Requirements

User/Stakeholder Requirement

User or Stakeholder requirements express stakeholders' desires and needs that shall be satisfied by building a system, seen from the stakeholders' perspective.

[IREB]

Example

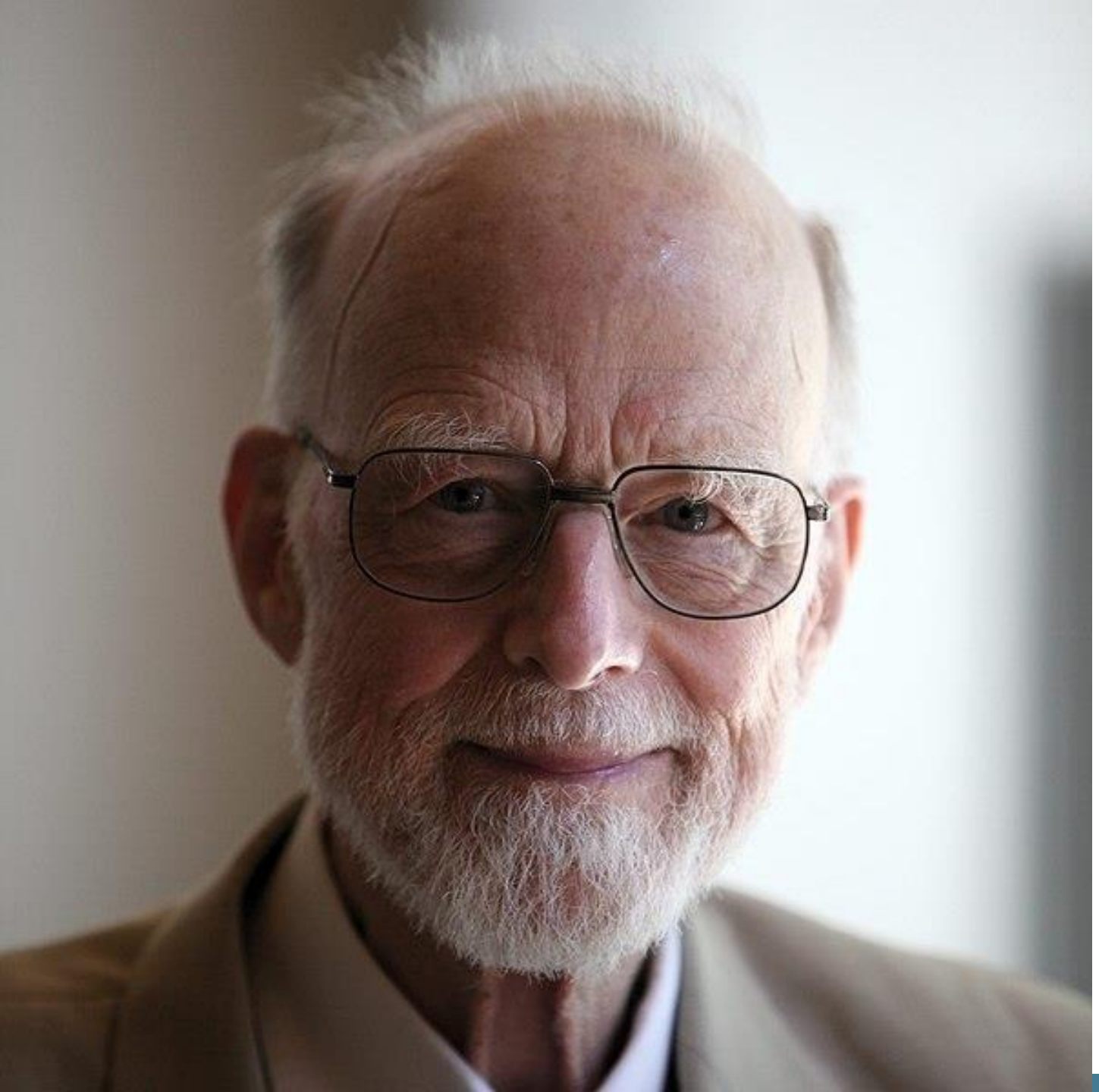
The Corona app shall track contacts by means of random IDs and should store them locally for two weeks.

User/Stakeholder requirements document (Lastenheft)

System description from user's point of view. What? For what? (Was?, Wofür?)

Example

If a user has a positive test result, he or she can inform tracked contacts about their increased risk by means of a QR code.



“The most important property of a program is whether it accomplishes the intention of its user.”

– Tony Hoare

Universität zu Köln

System Requirements

System Requirement

System requirements describe how a system shall work and behave – as observed at the interface between the system and its environment – so that the system satisfies its stakeholders' desires and needs. In the case of pure software systems, we speak of software requirements.

[IREB]

Example

If two devices are within a range of 2m for at least 15 minutes, they exchange their IDs via Bluetooth.

Example

After two devices have exchanged their IDs, the devices shall store the IDs for two weeks.

System requirements document (Pflichtenheft)

System description from technical point of view. How? Whereby? (Wie?, Womit?)

Example

A device generates an ID every 24 hours and stores old IDs for two weeks.

Functional Requirements

Functional Requirement

Functional requirements concern a result or behavior that shall be provided by a function of a system. This includes requirements for data or the interaction of a system with its environment.

[IREB]

Example

If a user has a positive test result, he or she can inform tracked contacts about their increased risk by means of a QR code.

Example

After IDs have been exchanged, the devices store them for two weeks.

Quality Requirements

Quality Requirement

"Quality requirements pertain to a quality concern that is not covered by functional requirements such as performance, availability, security, or reliability"

[IREB]

Note

often more critical than functional requirements

Example

The delay between pressing a button and showing the next screen shall be < 0.1 sec.

Example

The app shall be easy to use.

Example

The app shall be available 99.999% of the time

Example

The app must not disclose any sensitive data.

Functional vs. Quality Requirements

- Popular distinction:
  - Functional requirements define *what* the system shall do
  - Quality requirements define *how* the system shall do it

The customer entry form shall contain fields for the customer's name and first name, taking up to 32 characters per field, displaying at least 24 characters, left-bound, with a 12 pt. sanserif font.

Functional requirement
→ but contains a lot about the *how*.

The system shall process measurement data produced by the detector of a high-energy particle accelerator in real time.

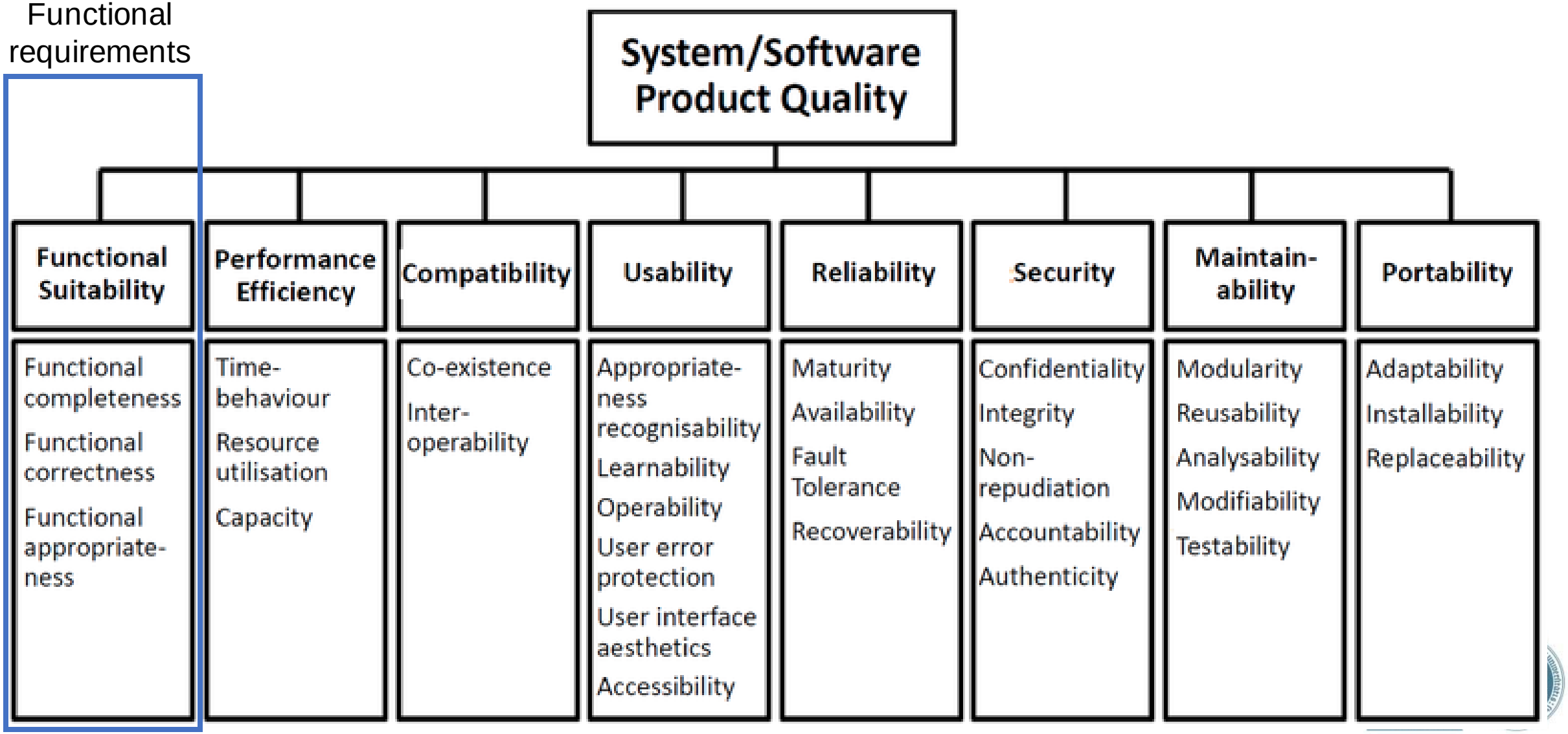
Performance requirement
→ but without, the system would not *function* at all

- Distinction not so obvious especially if requirements are very detailed or quality requirements are essential for the system

Universität zu Köln

ISO 25010 Quality Model

Functional requirements



Constraints

Constraint

"A requirement that limits the solution space beyond what is necessary for meeting the given functional requirements and quality requirements."

[IREB]

(Rahmenbedingungen)

Example

The app must be written in Swift.

Example

The source code of the app must be open source.

Note

Constraints come directly from a stakeholder (not under control of the system designers)

Example

Data collection must conform to GDPR (DSGVO).

Constraints vs. other system requirements

The systems *must*… **vs.** the system *shall*…

Process and Project Requirements

Process Requirement

Process requirements prescribe activities to be performed by the developing organization. For instance, process requirements could specify the methodologies that must be followed, and constraints that the organization must obey.

Example

Every code change must be reviewed by at least one member of the team.

Example

Every change request must be approved by the change control board.

Project Requirement

Project requirements are conditions or tasks that must be completed to ensure the success or completion of the project. They provide a clear picture of the work that needs to be done. They're meant to align the project's resources with the objectives of the organization.

Example

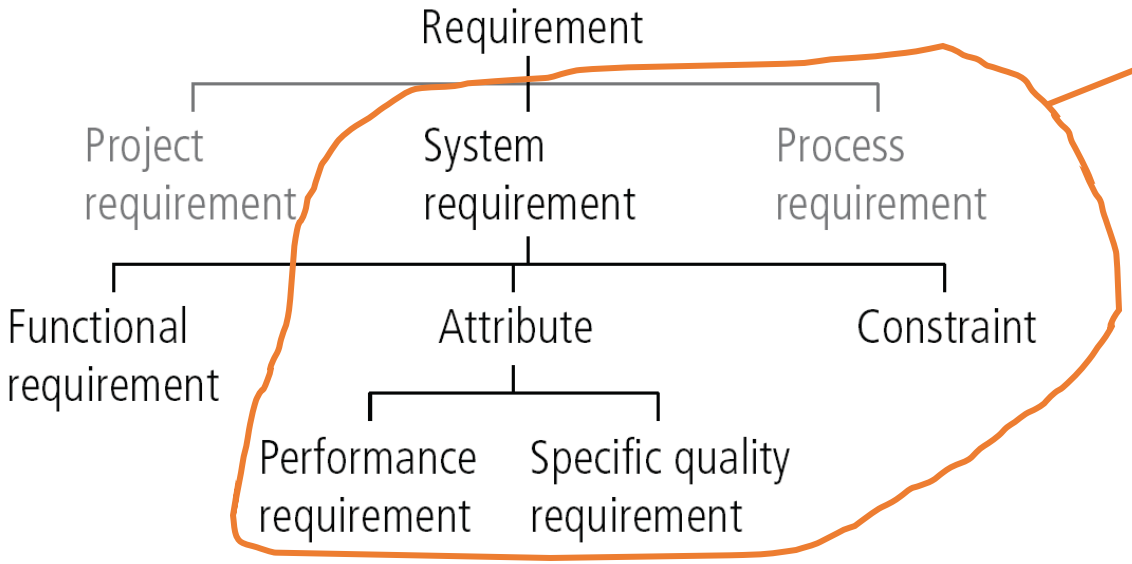
A first version of the app shall be released on May 01, 2024.

Example

The costs for development shall not exceed 0.5 Mio. Euro.

Universität zu Köln

Requirements Types



Functionality and behavior:
Functions
Data
Stimuli
Reactions
Behavior

Time and space bounds:
Timing
Speed
Volume
Throughput

"-ilities":
Reliability
Usability
Security
Availability
Portability
Maintainability
...

Physical
Legal
Cultural
Environmental
Design&Implementation
Interface
...

Sometimes called "non-functional requirements"

No[1]	**Question** Was this requirement stated because we need to specify...	**Result**
1	... some of the system's behavior, data, input, or reaction to input stimuli – regardless of the way how this is done?	Functional
2	... restrictions about timing, processing or reaction speed, data volume, or throughput?	Performance
3	... a specific quality that the system or a component shall have?	Specific quality
4	... any other restriction about what the system shall do, how it shall do it, or any prescribed solution or solution element?	Constraint

[1] Questions must be applied in this order

M. Glinz: On Non-functional Requirements, RE, 2007

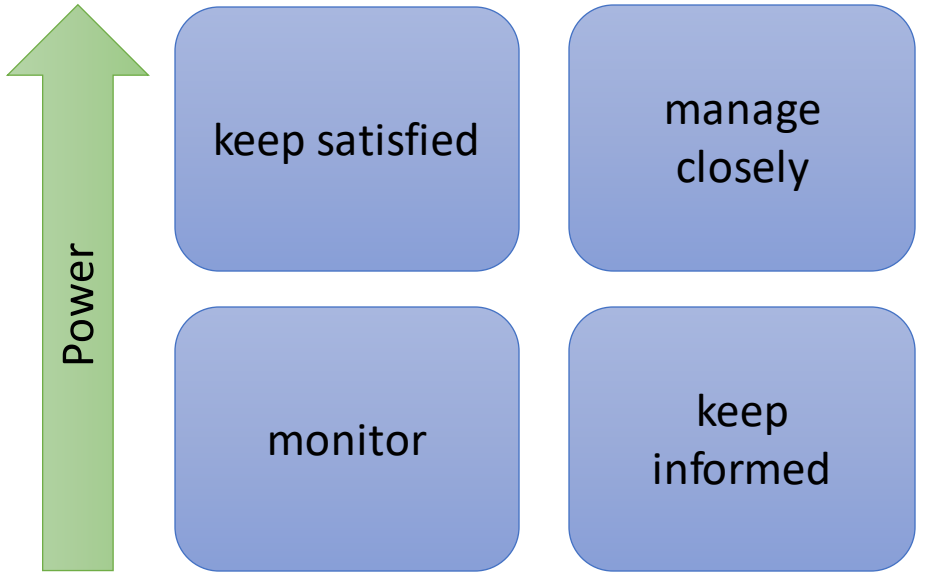
Stakeholders, Vision, and Scope

Stakeholder

- A person or organization who influences a system's requirements or who is impacted by that system.

- Characteristics
  - Power/Influence
  - Interest
  - Possibility to influence
  - Attitude (pos. vs. neg.)
  - Internal, partner, external

- The user is always a stakeholder



System Vision: What?

- The **system vision** is a joint vision of the system agreed upon all active stakeholders
- Characteristics
  - Big Picture
  - Abstract
  - Rich context
  - Focus on impact
- Purpose
  - Agreement on what is the project about
  - Easy communication with stakeholders (incl. developers)
  - Validation and elicitation of new aspects

System Vision: How?

- A system vision text should contain the following ingredients:
  - **Problem:** What are problems in a certain **context**. What are the reasons? How severe are the problems?
  - **Solution:** What is the **core idea** to solve the problem
  - **Value:** What is the added value of the solution? What is different with the solution?

Short!

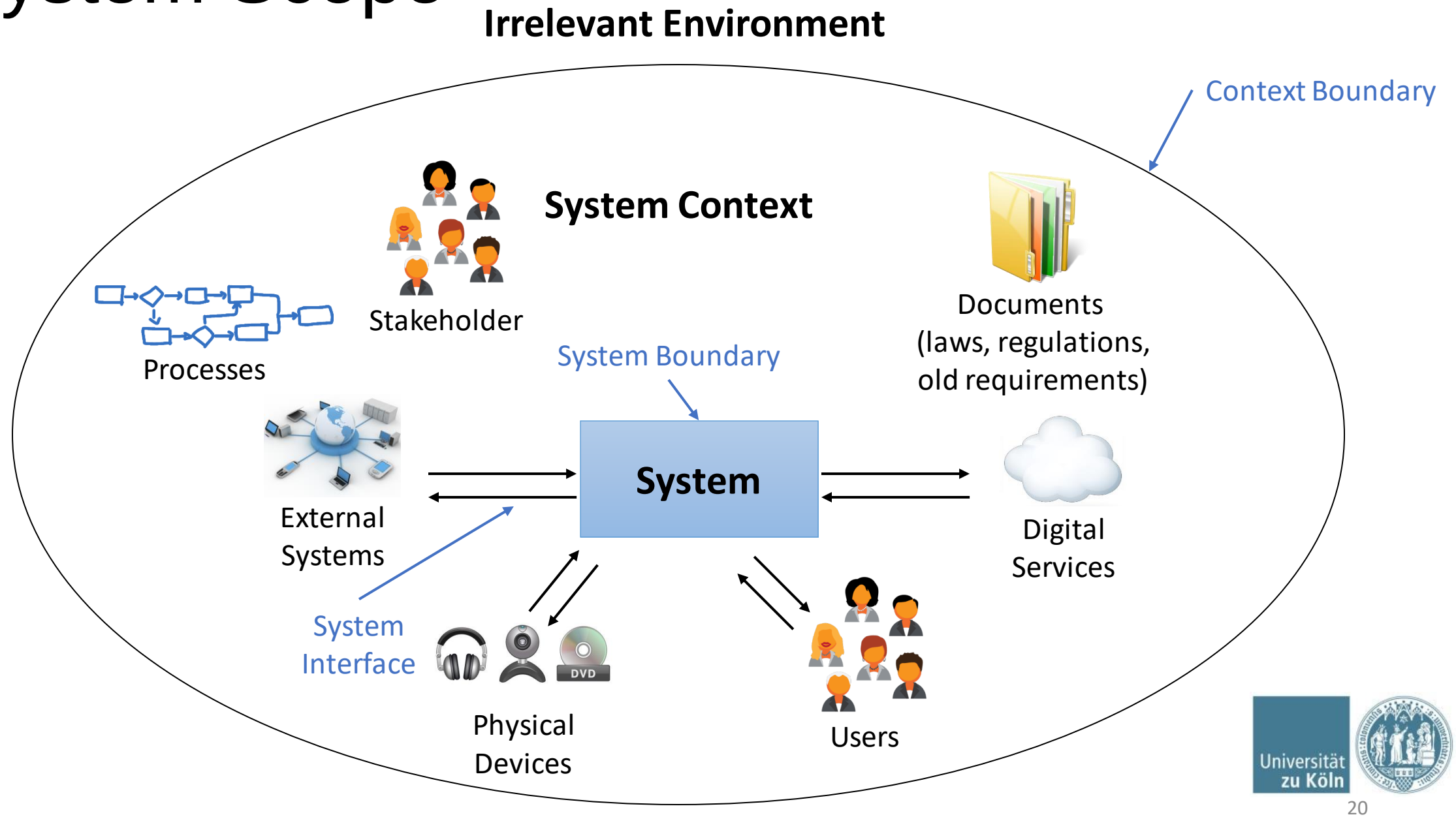
A few sentences to one page max

System vision of the Corona Warn App:

The aim of the Corona-Warn-App is to identify SARS-CoV-2 chains of infection as quickly as possible and to break them. It does this by rapidly and reliably notifying people that they have had contact with infected persons and therefore may have been exposed to the virus. These individuals can then self-isolate, to help stop the SARS-CoV-2 pandemic.

https://github.com/corona-warn-app/cwa-documentation/blob/master/scoping_document.md

System Scope



System Scope

- Elements within the context of a system are sources of requirements
- Each requirement must be justified by a context element
  - The justification is called the requirement's *rationale*

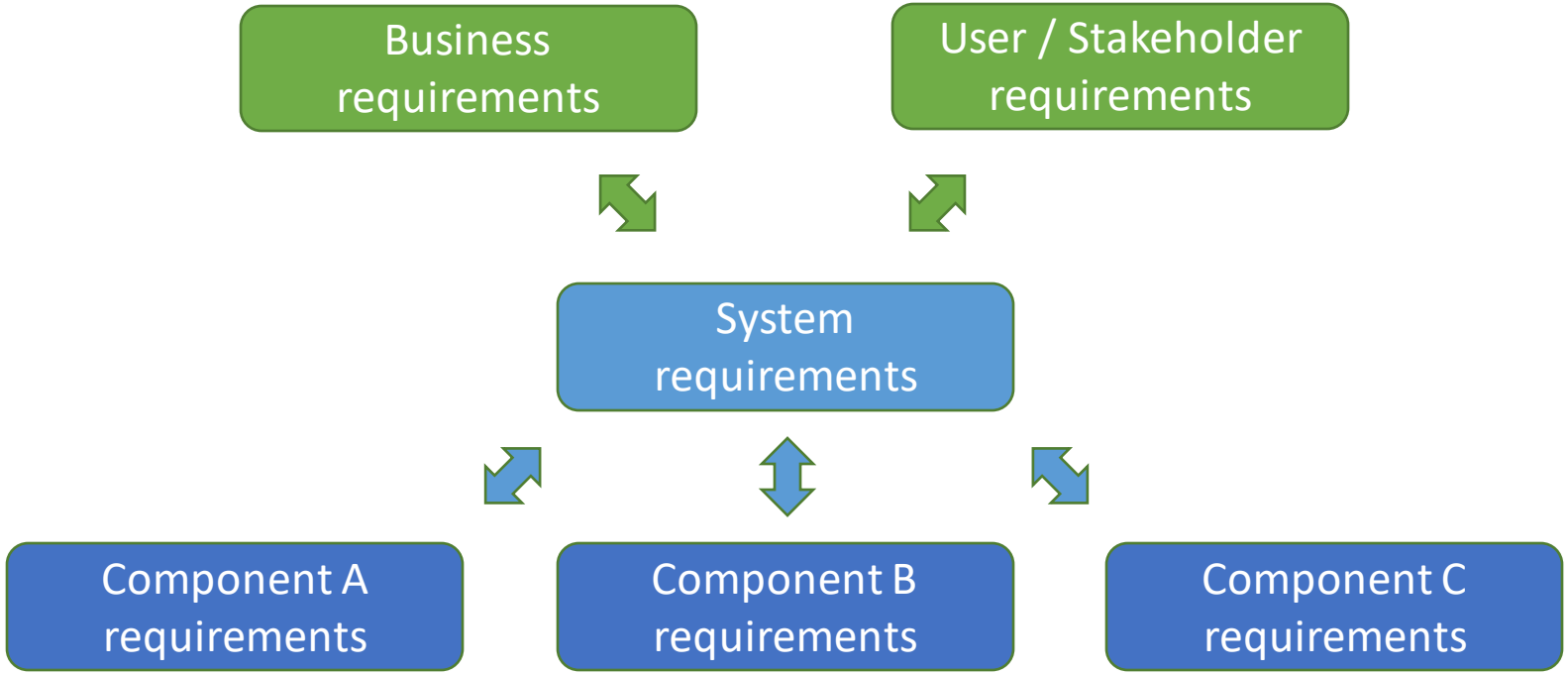
- The **system boundary** determines what is part of the system-under-development and what is part of the environment that has a relation to the system
- The **context boundary** distinguishes the relevant from the irrelevant parts of the environment
- The **system interface** determines the interaction possibilities with elements in the context at runtime

- System boundary, context boundary, and interface may change over time

How to structure requirements?

Requirements Work Products

- Requirements are documented in *requirements specifications*
- There may be requirements specifications on different requirements level



Requirements Work Products

- Requirements documents summarize and structure requirements from different levels of abstraction
- Requirements document templates provide guidelines how to structure requirements documents
  - A unified structure eases readability
  - Predefined sections ensure that you don’t forget anything
- Requirements document templates
  - Generic (e.g., ISO 29148)
  - Domain-specific
  - Company-specific
  - Project-specific

ISO 29148: Requirements Documents Templates

Stakeholder Requirements Specification

1. Introduction
- 1.1 Business purpose
- 1.2 Business scope
- 1.3 Business overview
- 1.4 Definitions
- 1.5 Stakeholders

2. References

3. Business management requirements
- 3.1 Business environment
- 3.2 Goal and objective
- 3.3 Business model
- 3.4 Information environment

4. Business operational requirements
- 4.1 Business processes
- 4.2 Business operational policies and rules
- 4.3 Business operational constraints
- 4.4 Business operational modes
- 4.5 Business operational quality
- 4.6 Business structure

5. User requirements

6. Concept of proposed system
- 6.1 Operational concept
- 6.2 Operational scenario

7 Project Constraints

8. Appendix
- 8.1 Acronyms and abbreviations

System Requirements Specification

1. Introduction
- 1.1 System purpose
- 1.2 System scope
- 1.3 System overview
  - 1.3.1 System context
  - 1.3.2 System functions
  - 1.3.3 User characteristics
- 1.4 Definitions

2. References

3. System requirements
- 3.1 Functional requirements
- 3.2 Usability requirements
- 3.3 Performance requirements
- 3.4 System interface
- 3.5 System operations
- 3.6 System modes and states
- 3.7 Physical characteristics
- 3.8 Environmental conditions
- 3.9 System security
- 3.10 Information management
- 3.11 Policies and regulations
- 3.12 System life cycle sustainment
- 3.13Packaging, handling, shipping and transportation

4. Verification
(parallel to subsections in Section 3)

5. Appendices
- Assumptions and dependencies
- Acronyms and abbreviations

Software Requirements Specification

1. Introduction
- 1.1 Purpose
- 1.2 Scope
- 1.3 Product overview
  - 1.3.1 Product perspective
  - 1.3.2 Product functions
  - 1.3.3 User characteristics
  - 1.3.4 Limitations
- 1.4 Definitions

2. References

3. Specific requirements
- 3.1 External interfaces
- 3.2 Functions
- 3.3 Usability Requirements
- 3.4 Performance requirements
- 3.5 Logical database requirements
- 3.6 Design constraints
- 3.7 Software system attributes
- 3.8 Supporting information

4. **Verification**
(parallel to subsections in Section 3)

5. Appendices
- 5.1 Assumptions and dependencies
- 5.2 Acronyms and abbreviations

What is Requirements Engineering?

What is Requirements Engineering?

A systematic and disciplined approach to the specification and management of requirements with the following goals:

- Knowing all relevant requirements
- Achieving a consensus among the stakeholders about these requirements
- Documenting requirements appropriately
- Managing requirements systematically

- *Synonyms:* Requirements Analysis, Requirements Management, Business Analysis, Business Process Analysis, Requirements Specification,…

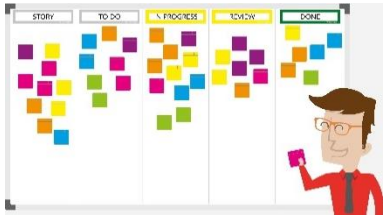
M. Glinz: A Glossary of Requirements Engineering Terminology, v1.6, IREB, 2014

Requirements Engineering
is where the
informal meets the formal
– Michael Jackson

Universität zu Köln

There is not only one RE method!

- “Classical” RE
  - Result: A requirements specification (document)
  - Basis for contracting and implementation
  - Typical in customer/supplier situations
- “Agile” RE
  - Result: Product backlog, user stories, epics, …
  - Basis for sprint planning
  - Typical in most “in-house” SW development projects
- “Change-based” RE
  - Result: Issues, Tickets, Change Requests,…
  - Basis for product maintenance and evolution
  - Typical in many “long-lived systems”
- “Code-based” RE
  - Results: Code comments, manpages, mailing lists,…
  - Basis for understanding and changing code
  - Typical in open source projects



JIRA

<code>

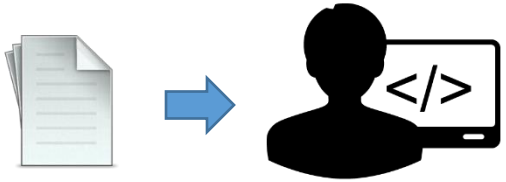
RE Impact

Why to write down requirements?

Consolidation

Contracting

Development



Cost Estimation

Testing

Cross-project/product

Challenges of RE (in practice)

Naming the Pain in RE – Top 10 RE Problems

1. Incomplete and/or hidden requirements
2. Communication flaws between project team and customer
3. Moving targets (changing goals, business processes and/or requirements)
4. Underspecified requirements that are too abstract
5. Time boxing / Not enough time in general
6. Communication flaws within the project team
7. Stakeholders with difficulties in separating requirements from known solution designs
8. Insufficient support by customer
9. Inconsistent requirements
10. Weak access to customer needs and/or business information

Universität zu Köln

D. Mendez Fernandez et al.: Naming the Pain in Requirements Engineering: Contemporary Problems, Causes, and Effects in Practice, 2016

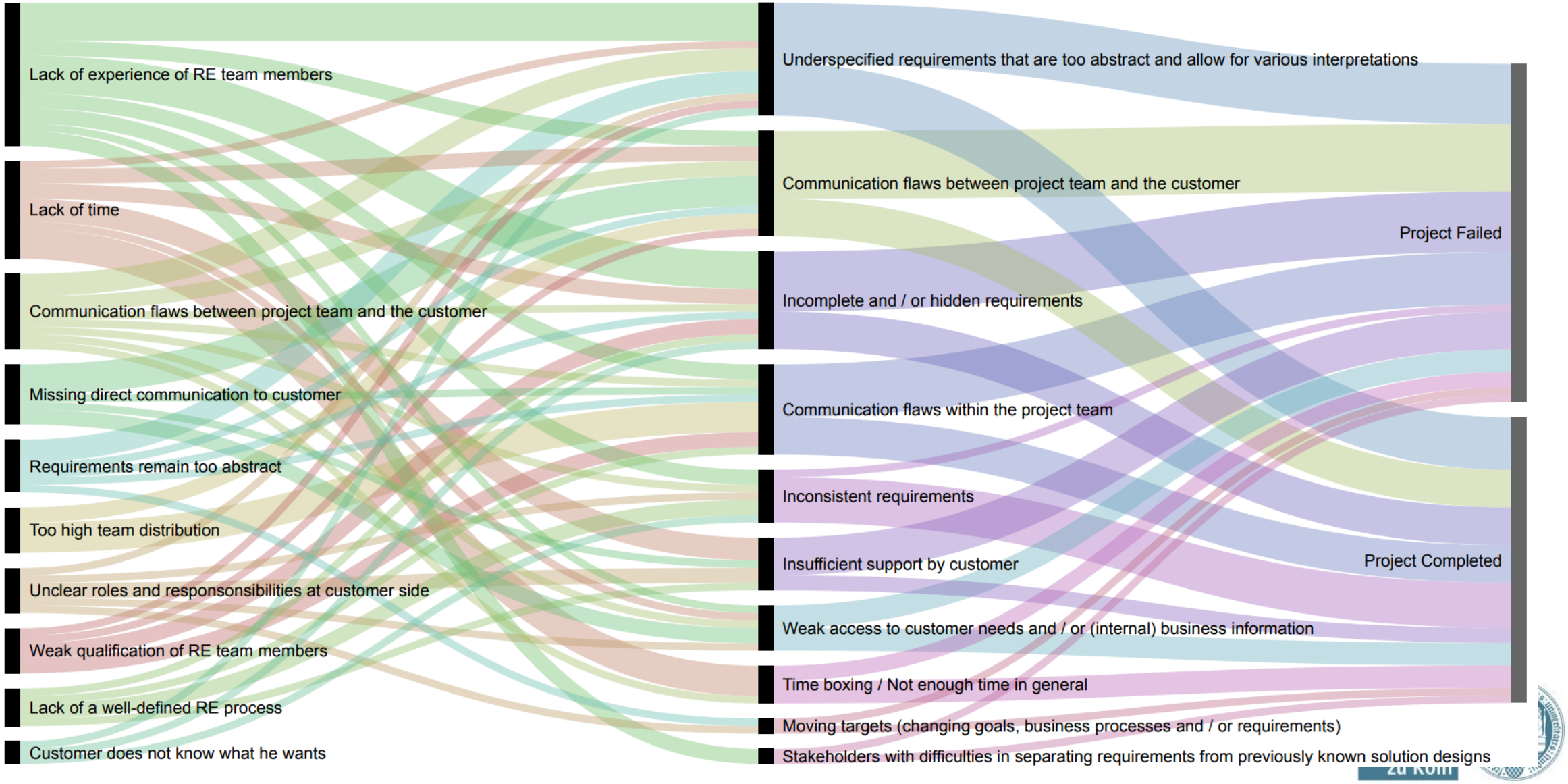
Top 10 Causes for RE Problems

1. Lack of time
2. Lack of experience of RE team members
3. Weak qualification of RE team members
4. Communication flaws between project team and the customer
5. Requirements remain too abstract
6. Changing business needs
7. Customer does not know what he wants
8. Missing direct communication to customer
9. Language barriers
10. Strict time schedule by customer

Universität zu Köln

D. Mendez Fernandez et al.: Naming the Pain in Requirements Engineering: Contemporary Problems, Causes, and Effects in Practice, 2016

Causes, Effects and Project Success



D. Mendez Fernandez et al.: Naming the Pain in Requirements Engineering: Contemporary Problems, Causes, and Effects in Practice, 2016

Why is RE so Challenging?

- Much RE has no value per se
- It is not the requirements engineer who directly profits from good RE work
- Complete and unambiguous requirements specifications are an illusion
- RE is not just about SW but about humans, systems, and processes

→ As little as possible, as much as needed

→ Work value-oriented and risk-driven

→ Create and ensure common understanding

→ Cooperation over confrontation and contracts.