Moodle-Kurs zur Vorlesung und Übung

URL: https://moodle.uni-due.de/course/view.php?id=832

Einschreibeschlüssel: REM2122!

Hinweise zur Präsenzlehre in der Corona-Pandemie

- **UDE-Boardingsystem**:
  - 3G-Regelung (getestet – geimpft – genesen) ist Zugangsvoraussetzung für alle Veranstaltungen der UDE
  - Keine Teilnahme ohne Vorlage eines 3G-Nachweises
  - UDE-Boarding-System: Kontrolle der 3G-Regelung und Zugang zu den Lehrveranstaltungen
  - Nähere Informationen: https://www.uni-due.de/de/covid-19/ude-boarding.php
- **Während Vorlesung und Übung**:
  - ***Maskenpflicht***: Tragen eines medizinischen Mundnasenschutzes (OP-Maske, FFP2-Mund-Nasenschutz) im gesamten Gebäude und an den Sitzplätzen
    - Ausnahme: Vortragende/Vortragender
- ***Bitte den Aufzug im SH-Gebäude nur mit Abstand und mit max. 2 Personen gleichzeitig nutzen***

Stand: 12.10.2021

Topics of this Lecture

1. Introduction and Fundamentals
2. Framework for Requirements Engineering
3. Context
4. Documentation
5. Conceptual Modeling
6. Goals
7. Scenarios
8. Solution-Oriented Requirements
9. Negotiation & Validation
10. Management

Book for this Lecture

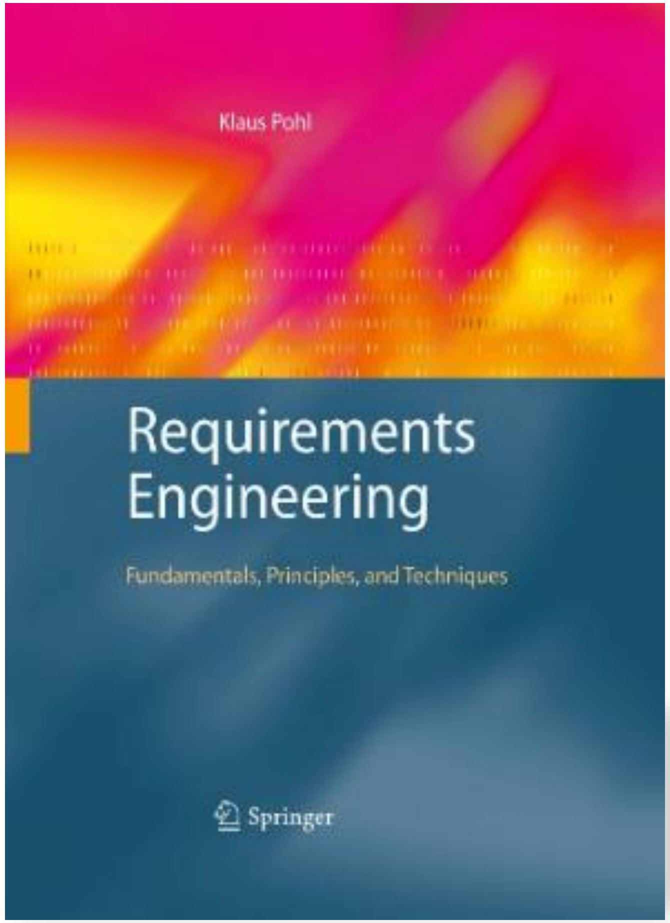
Klaus Pohl

Requirements Engineering

Fundamentals, Principles and Techniques

1st edition, Springer, 2010.

http://www.requirements-book.com/



Klaus Pohl

Requirements Engineering

Fundamentals, Principles, and Techniques

Springer

Requirements Engineering & Management

Introduction and Fundamentals I

Prof. Dr. Klaus Pohl

PALUNO
The Ruhr Institute for Software Technology

Agenda

1. Importance of Requirements Engineering
2. Three Dimensions of Requirements Engineering
3. Three Types of Requirements

1. Importance of Requirements Engineering

No Silver Bullet

"The hardest single part of building a software system is **deciding precisely what to build**.

No other part of the conceptual work is as difficult as establishing the **detailed technical requirements**, including all the interfaces to people, to machines, and to other software systems.

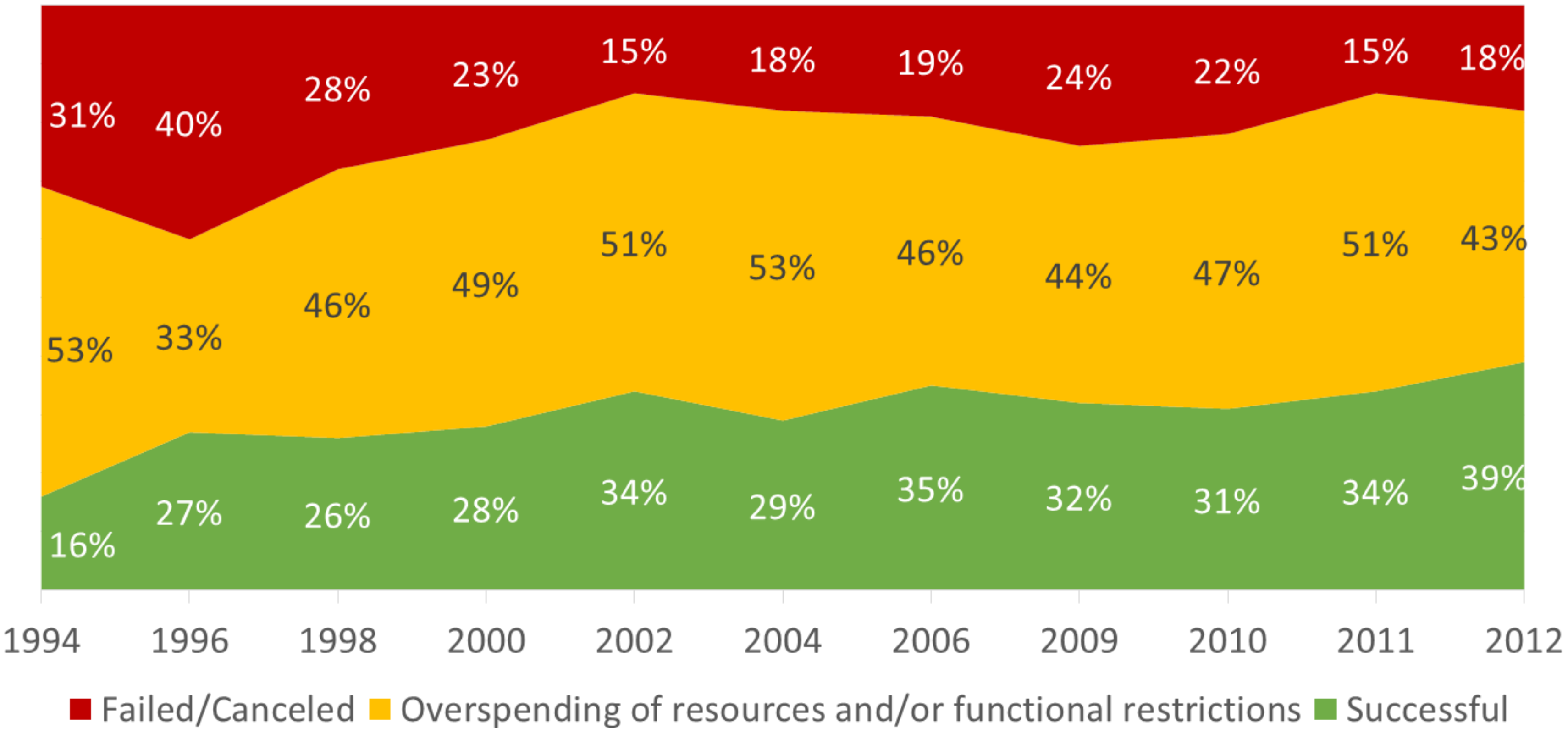
No other part of the work so cripples the resulting system if done wrong. No other part is more **difficult to rectify later**."

[Brooks 1987]

Importance of Requirements Engineering

CHAOS Report – Project Success and Failure

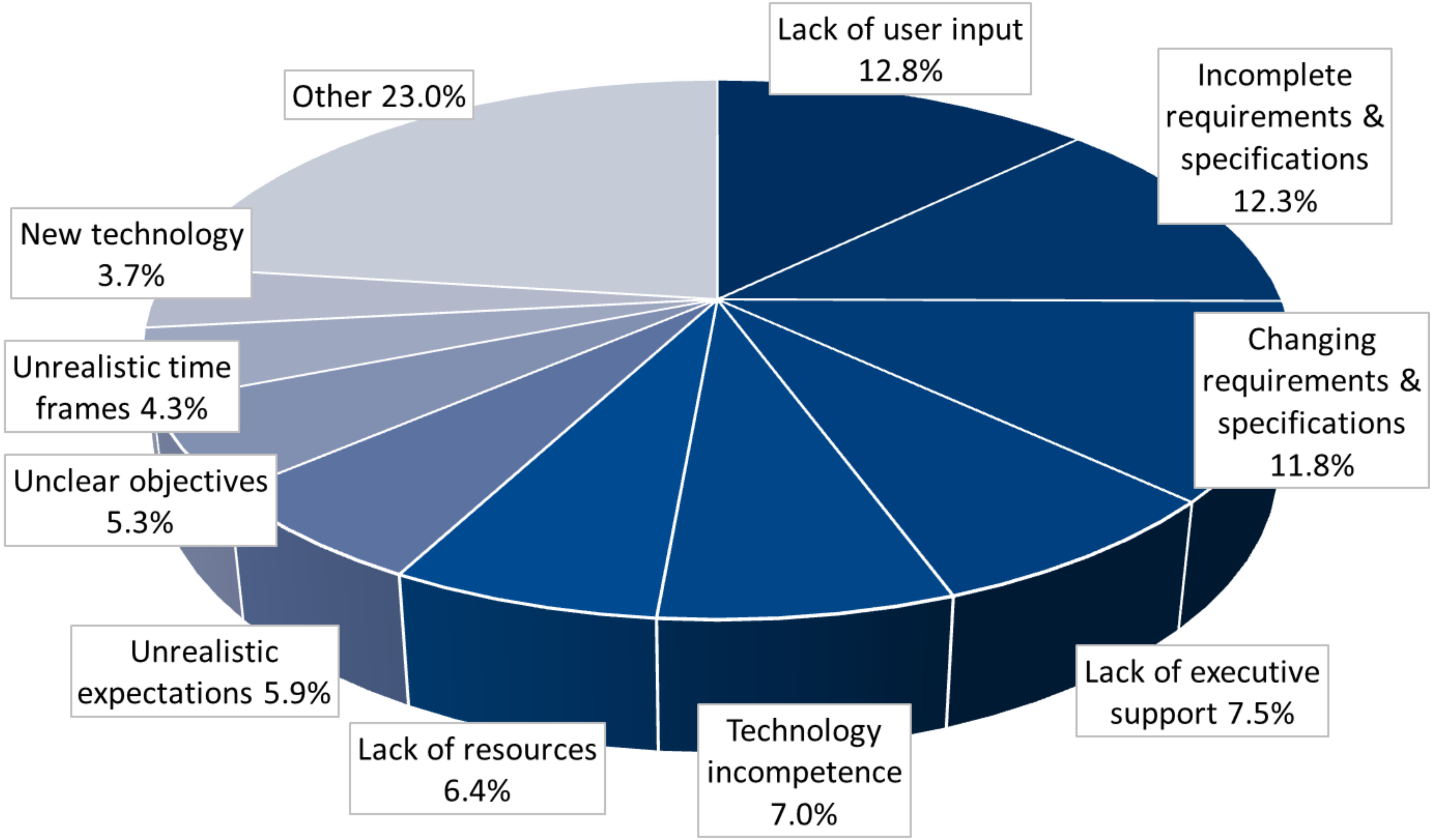
In average only **30%** of the projects were successful, **23%** were cancelled, and **47%** were completed with resource overspend and functional restrictions.



Based on [Wikipedia 2014]

Importance of Requirements Engineering

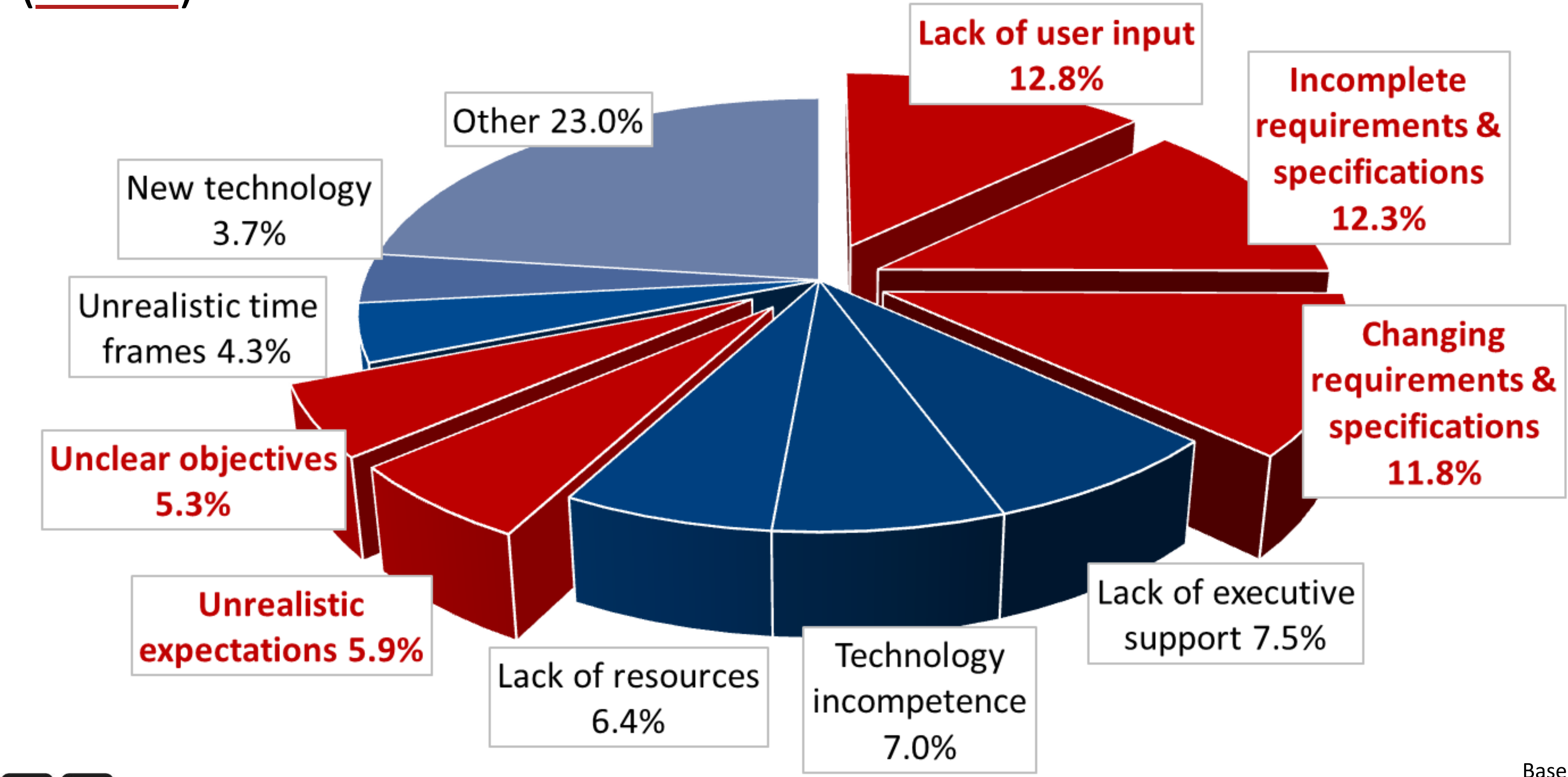
CHAOS Report – Reasons for Resource Overspend and Functional Restrictions



Based on [The Standish Group 1995]

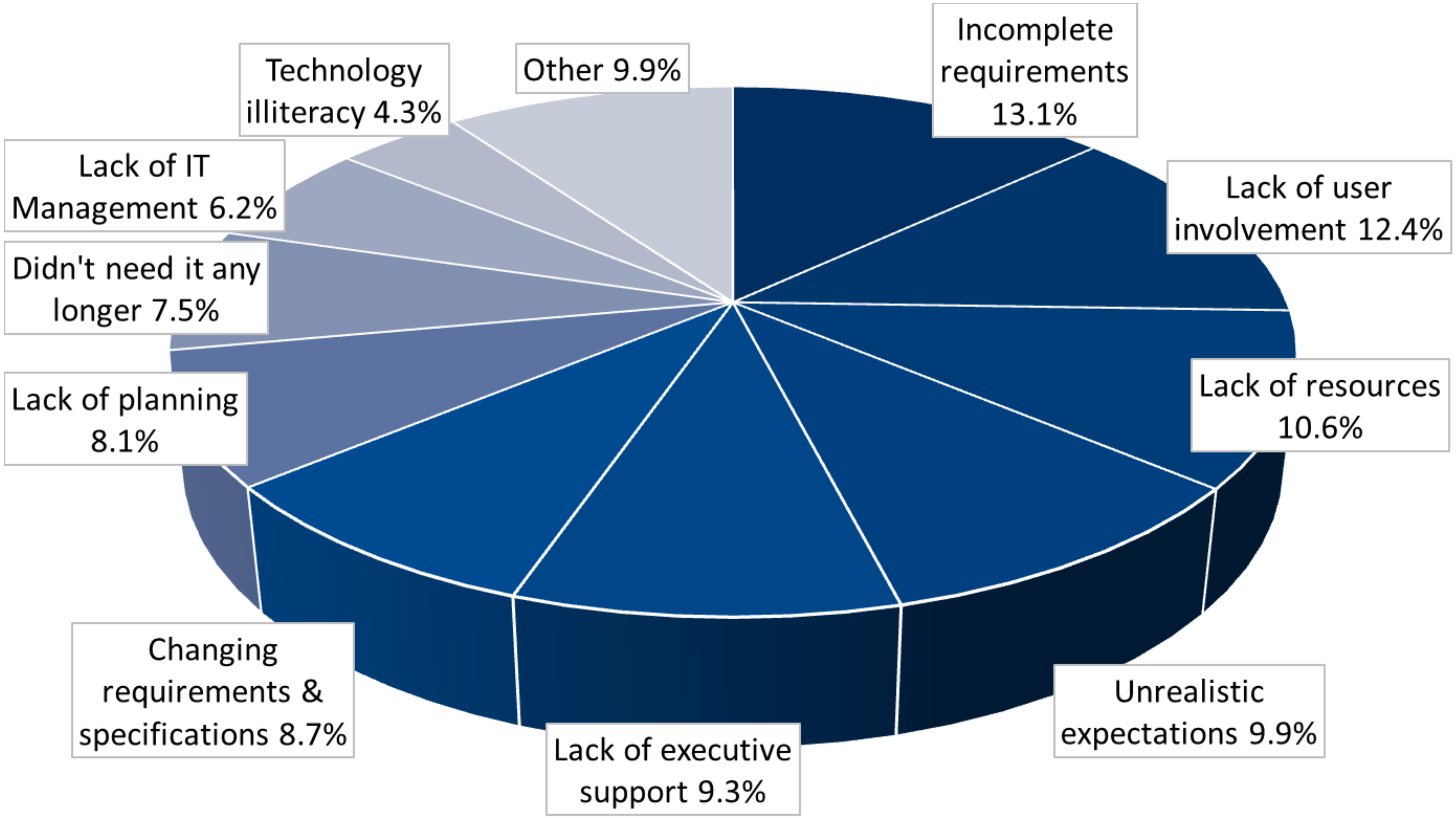
CHAOS Report – Reasons for Resource Overspend and Functional Restrictions

Reasons related to insufficient and **poor requirements engineering**! (**48.1%**)



Based on [The Standish Group 1995]

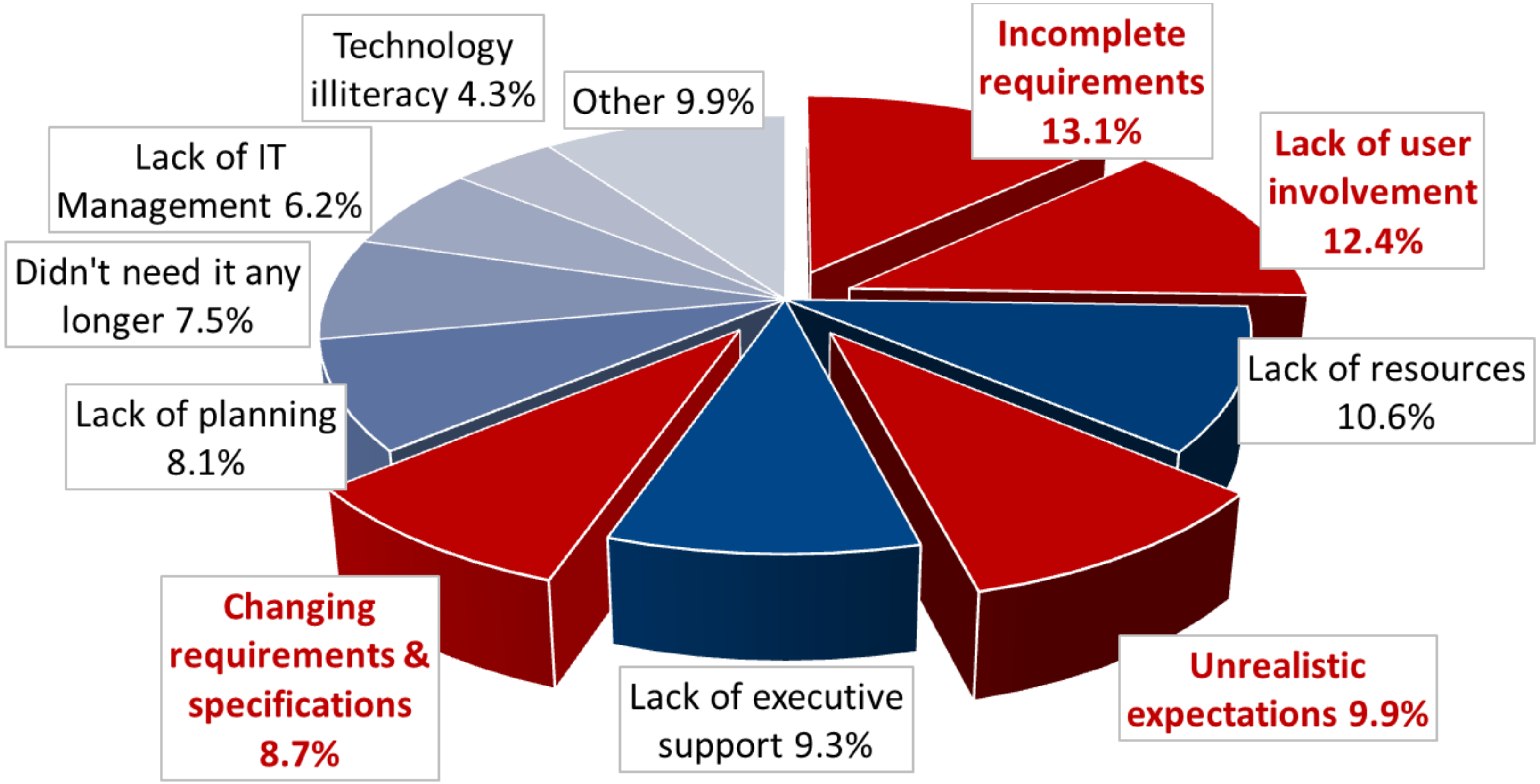
CHAOS Report – Reasons for Projects Cancelled



Based on [The Standish Group 1995]

CHAOS Report – Reasons for Projects Cancelled

Reasons related to insufficient and **poor requirements engineering**! (**44.1%**)



Based on [The Standish Group 1995]

Requirements Defects

Effort required for correcting requirements defects:

- Defect found after delivery **(small & non-critical projects)**
  - increase of effort required for correction by **factor 5**!
- Defect found after delivery **(large & critical projects)**
  - increase of effort required for correction by **factor 100**!

Based on [Boehm and Basili 2001]

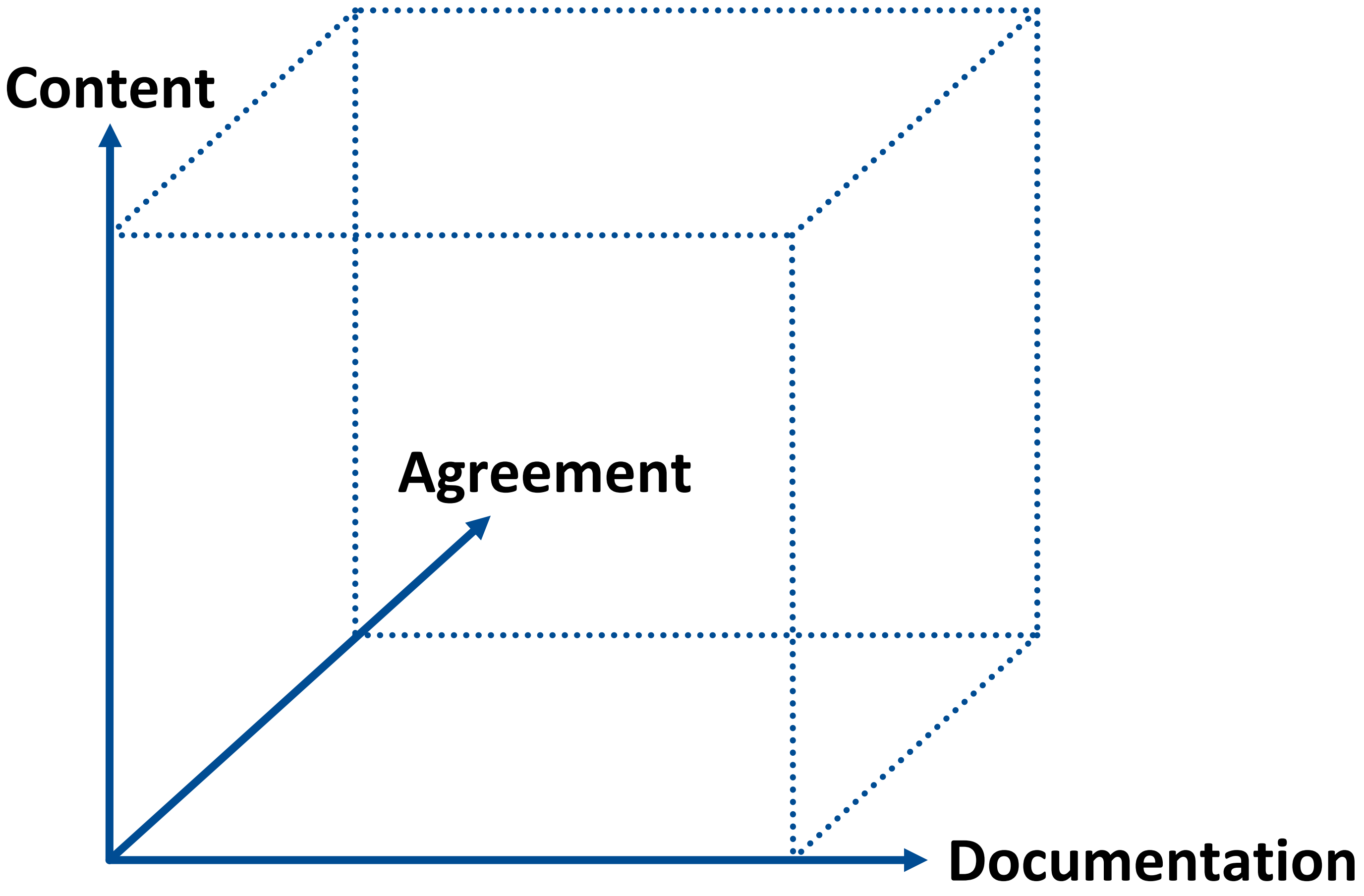
Example of Requirements Defects: The London Ambulance System (LAS)

- System vision: Partially **automate** the manual **processing of emergency calls**
  - Based on the location of an emergency, the LAS should **determine** and **dispatch the ambulances** ready for service in the area of the emergency

- **Requirements defects** caused by:
  - **Insufficient involvement** of ambulance crews
    - → Usability issues, inappropriate user interfaces
  - **Insufficient consideration** of the communication network capabilities
    - → No awareness of the existence of radio black spots

2. Three Dimensions of Requirements Engineering

Three Dimensions of Requirements Engineering

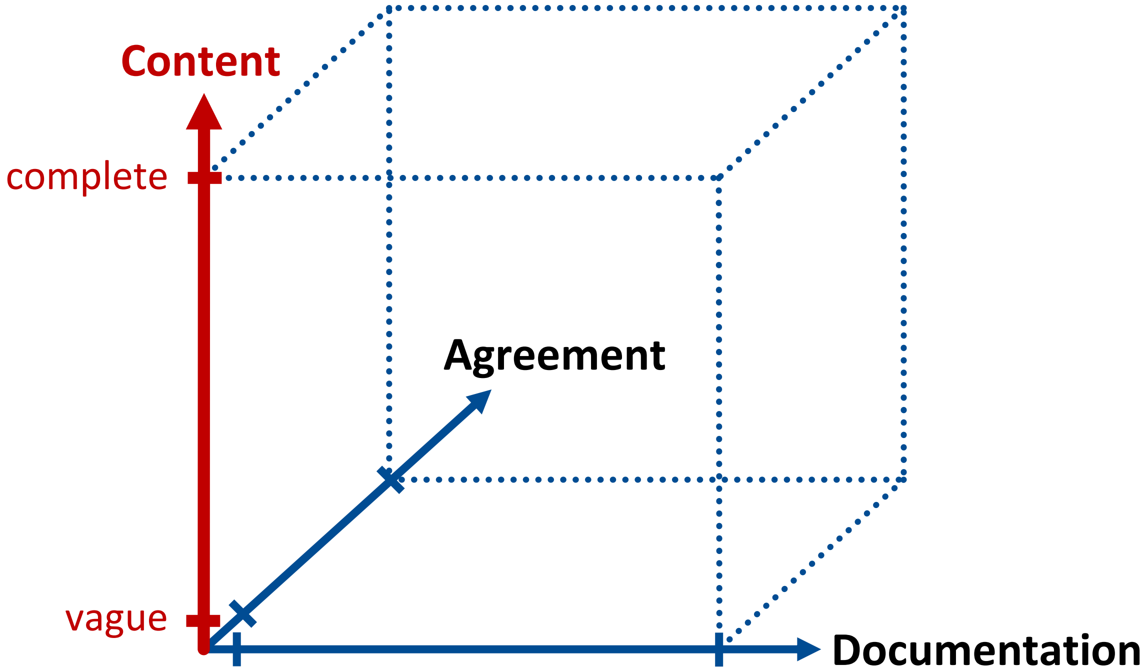


Based on [Pohl and Ulfat-Bunyadi 2013]

Content Dimension

- The content dimension deals with **the understanding of the system requirements and its context**.
- At the beginning of the requirements engineering process **only a few requirements** are known and their understanding is typically **vague**

At the end:
All requirements shall be explicitly **known** and **each** requirement shall be **understood** at the required level of detail.

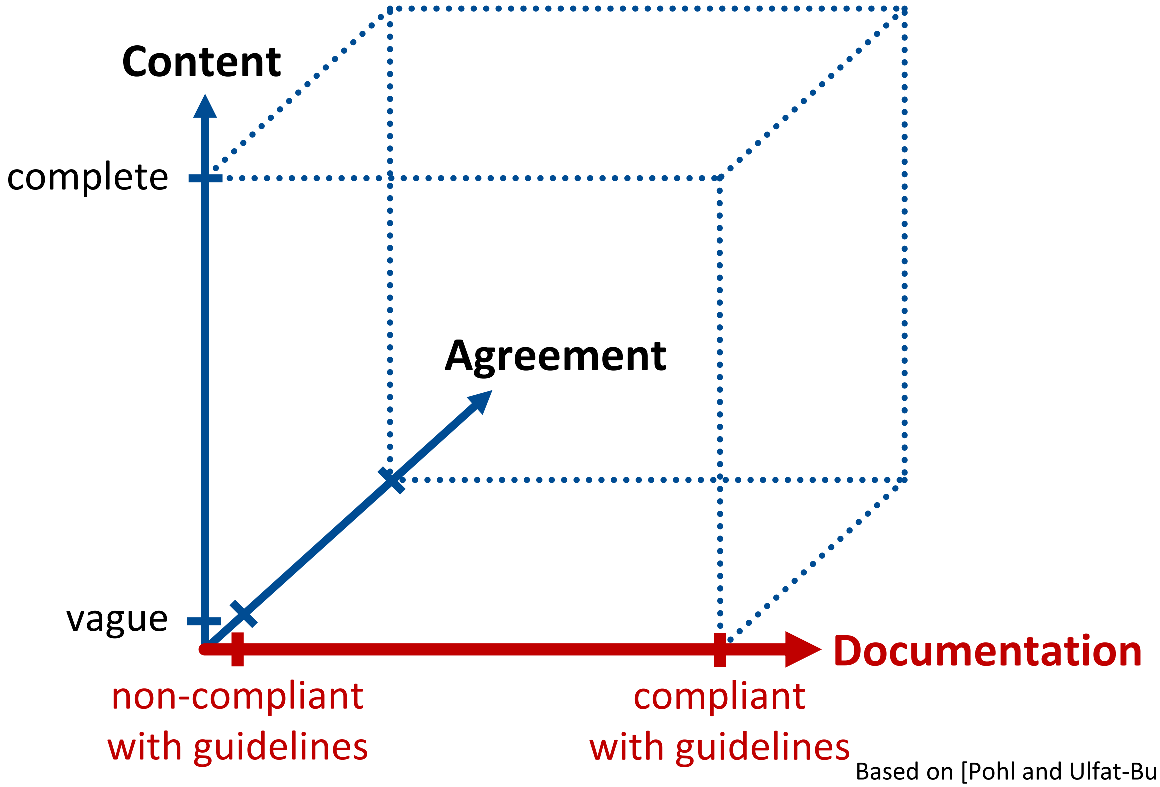


Based on [Pohl and Ulfat-Bunyadi 2013]

Documentation Dimension

- The documentation dimension deals with **documenting and specifying the system requirements** using **appropriate documentation and specification formats**.
- At the beginning information is typically documented **non-compliant with documentation/specification guidelines**, e.g., either as a note, sketch, statement in a minute, hand drawing, …

At the end: All requirements should be **documented** /**specified** in **compliance** with the relevant **documentation / specification guidelines** and formats.

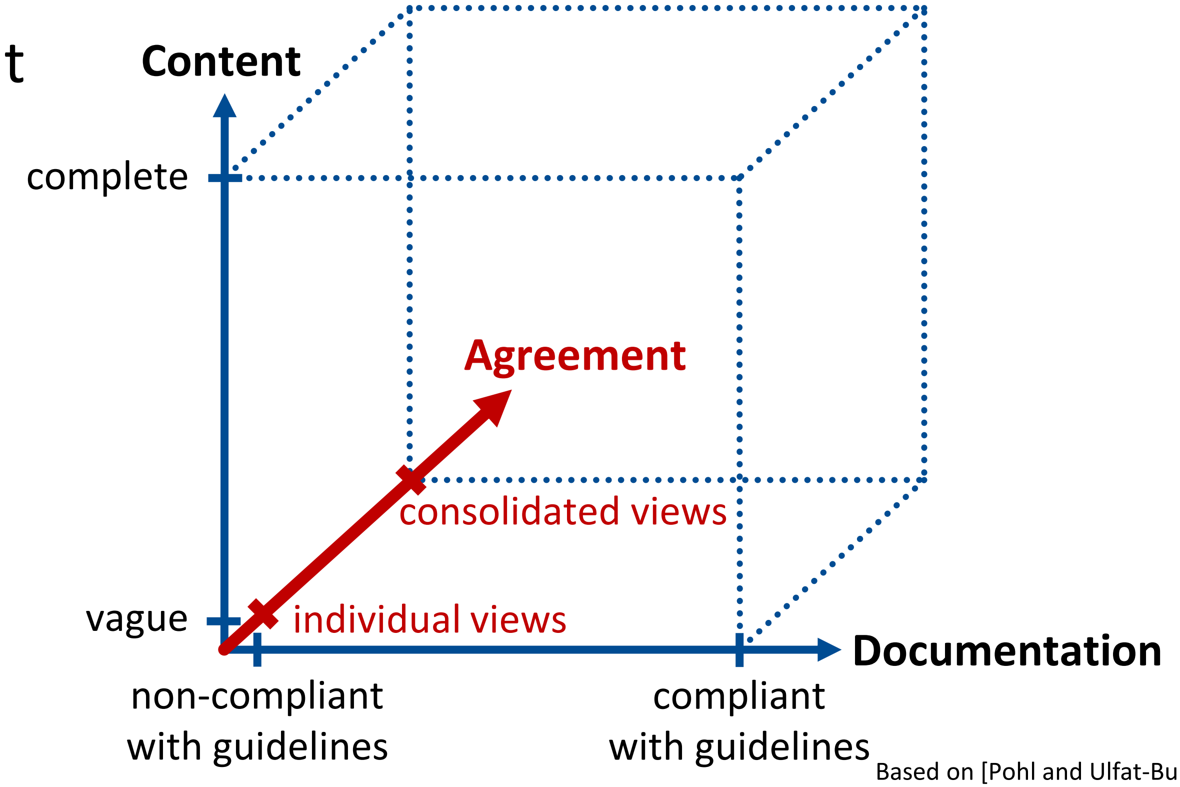


Based on [Pohl and Ulfat-Bunyadi 2013]

Agreement Dimension

- The agreement dimension deals with the level of **agreement achieved between the relevant stakeholders** about the known requirements.
- At the beginning stakeholders typically have **different, conflicting views** about the goals and requirements of the system.
- Conflicts between stakeholders about requirements have to be **detected** and **resolved** as early as possible.

At the end:
The relevant **stakeholders** should have **established a sufficient agreement about the system requirements**.



Based on [Pohl and Ulfat-Bunyadi 2013]

Orthogonality of Content and Agreement

- A **complete understanding** of a requirement **does not imply** a sufficient **agreement** between the stakeholders about this requirement!
- An **agreement** between stakeholders about a requirement **does not imply** a **complete understanding** of this requirement!

- **Progress**
  - in the **agreement dimension** (e.g., a creative solution for a conflict on requirements) **can lead** to **new requirements** which are **not well understood** (content dimension).
  - in the **content dimension** (e.g., by eliciting new requirements) **may lead to new conflicts** between stakeholders, or may uncover existing conflicts (agreement dimension).

Orthogonality of Documentation and Content

- **Compliance** of a requirement to the documentation guidelines **does not imply** a **complete understanding** of the requirement!
- A **complete understanding** of a requirement **does not imply the compliance** of the documentation of the requirement **with the documentation guidelines**!

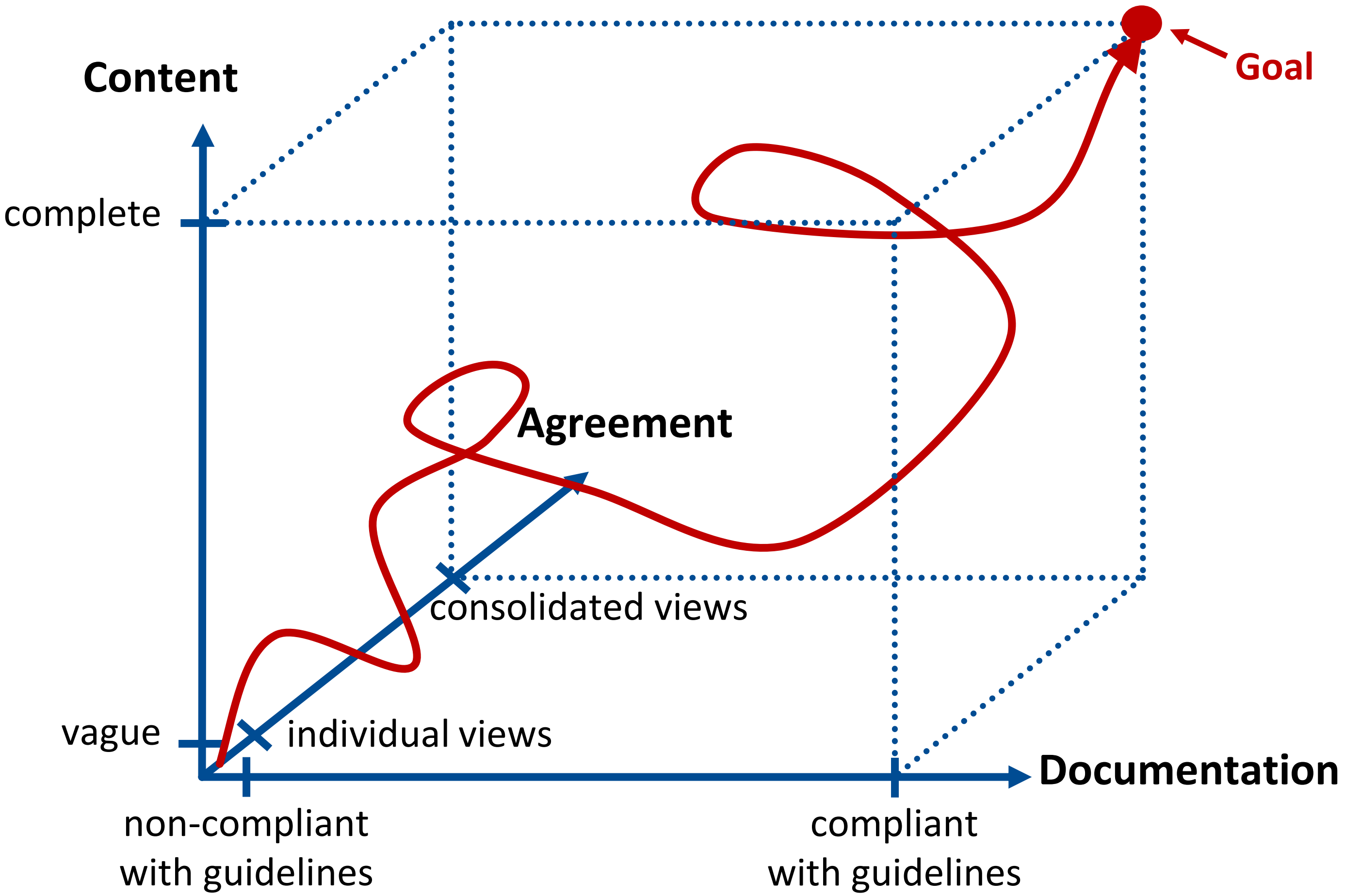
- **Progress**
  - in the **documentation dimension** (e.g., the formalization of a requirement according to the guidelines) **can reveal some gaps** within the content dimension.
  - in the **content dimension** (e.g., a new elicited requirement) **can lead** to a **drawback** in the **documentation dimension** since, most likely, the new requirement is not directly documented according to the documentation guidelines.

Orthogonality of Agreement and Documentation

- An **agreement** between stakeholders about a requirement **does not imply compliance** of the documentation of the requirement **to the documentation guidelines**!
- **Compliance** of a requirement **with the documentation guidelines does not imply** a sufficient **agreement** of the stakeholders about this requirement!

- **Progress**
  - in the **documentation dimension** (e.g., the documentation of a requirement according to the guidelines) can **surface a conflict** about the requirement (e.g., due to better comprehension of the requirement by a stakeholder).
  - in the **agreement dimension** (e.g., the stakeholders solve a conflict by agreeing on a new alternative requirement) can lead to a **drawback in the documentation dimension**, if the new documented requirement is initially non-compliant with documentation guidelines.

Requirements Engineering Process within the Three Dimensions



Based on [Pohl and Ulfat-Bunyadi 2013]

Three Dimensions of Requirements Engineering

(Goal of) Requirements Engineering

D Requirements engineering (RE) is a **cooperative**, **iterative** and **incremental process** which aims at ensuring that:

(1) All **relevant requirements** are **explicitly known and understood** at the **required level of detail**.

(2) A **sufficient agreement** about the system requirements is achieved **between the stakeholders** involved.

(3) All requirements are **documented and specified** in compliance with the relevant **documentation/specification guidelines**.

3. Three Types of Requirements

Requirement: Definition

D A requirement is a

(1) **statement** that **translates or expresses** a need and its associated **constraints** and **conditions**.

(2) **condition or capability** that shall be met or **possessed by**

(3) **a system, system component, product or service** to satisfy a contract, an agreement, standard, specification or other formally imposed documents.

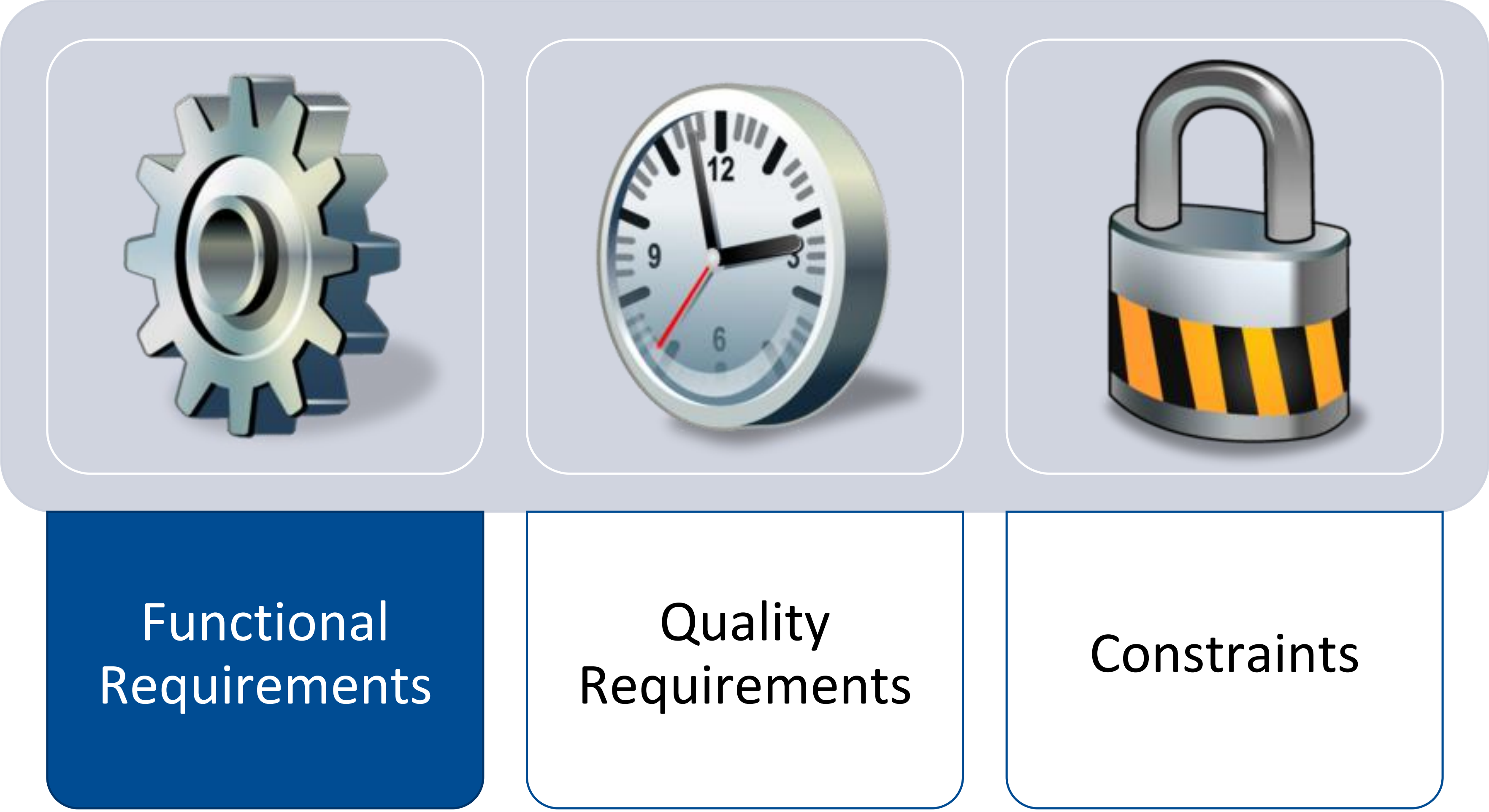
(4) **provision** that **contains criteria** to be fulfilled.

(5) **condition or capability** that must be present in a product, service, or result to satisfy a contract or other formally imposed specification.

Based on [ISO/IEC/IEEE 24765:2017]

Three Types of Requirements

Functional Requirements (1)



Functional Requirements

Quality Requirements

Constraints

Functional Requirements (2)

D A functional requirement is a

(1) **Statement defining the results** a product or process shall produce.

(2) **Requirement specifying** a function a system or system component shall perform.

Based on [ISO/IEC/IEEE 24765:2017]

Three Types of Requirements

Functional Requirements (3)

E

Functional Requirement
R-2: The electronic door locking system shall generate monthly reports containing all granted and denied admittances to the house.
R-3: If the user enters a correct PIN (personal identification number) at the keypad, the system shall open the door and record the granted access, i.e. the system should record the date and time, and the name of the PIN owner.
R-6: If a sensor detects a broken or damaged glass pane, the system shall inform the security company.

R-x: unique identifier

Quality Requirements (1)

Functional Requirements

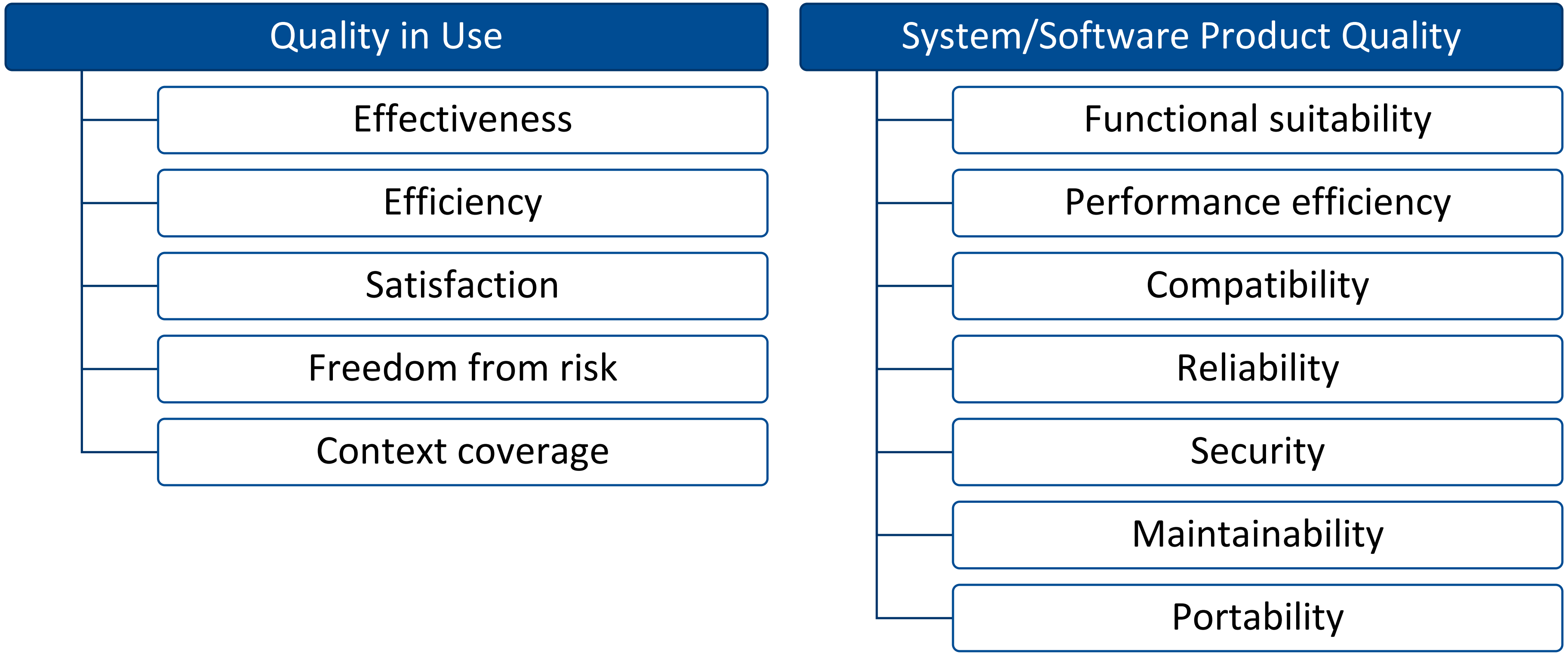
Quality Requirements

Constraints

Quality Requirements (2)

D A quality requirement defines **the expected level and/or boundaries** regarding a specific quality property of the entire system, a system component, a service, or a system function

Quality Requirements: Taxonomy



Quality in Use
- Effectiveness
- Efficiency
- Satisfaction
- Freedom from risk
- Context coverage

System/Software Product Quality
- Functional suitability
- Performance efficiency
- Compatibility
- Reliability
- Security
- Maintainability
- Portability

Based on [ISO/IEC 25010:2011]

Three Types of Requirements

Quality in Use

Quality Requirement	Description
Effectiveness	Accuracy and completeness with which users achieve specified goals.
Efficiency	Resources expended in relation to the accuracy and completeness with which users achieve goals.
Satisfaction	Degree to which user needs are satisfied when a product or system is used in a specified context of use.
Freedom from risk	Degree to which a product or system mitigates the potential risk to economic status, human life, health or the environment.
Context coverage	Degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in both contexts of use and in contexts beyond those initially explicitly identified.

[ISO/IEC 25010:2011]

System/Software Product Quality (1)

Quality Requirement	Description
Functional suitability	Degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions.
Performance efficiency	Performance relative to the amount of resources (e.g., other software products, hardware configuration, etc.) used under stated conditions.
Compatibility	Degree to which a product, system or component can exchange information with other products, systems or components and/or perform its required functions, while sharing the same hardware or software environment.

[ISO/IEC 25010:2011]

System/Software Product Quality (2)

Quality Requirement	Description
Reliability	Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.
Security	Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorisation.
Maintainability	Degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers.
Portability	Degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operation or usage environment to another.

There are several other taxonomies for categorizing quality requirements!

[ISO/IEC 25010:2011]

Three Types of Requirements

Quality Requirements

E

Quality Requirement	Type of Quality
R-12: The user interface shall be easy to use for the house owner.	Satisfaction (Quality in Use)
R-15: The release of the locking mechanism shall take 0.8 seconds at most.	Performance efficiency (Quality in Use)
R-17: The user password stored in the system shall be protected against unauthorized access.	Security (System/Software Product Quality)

Three Types of Requirements

Constraints (1)

Functional Requirements

Quality Requirements

Constraints

Three Types of Requirements

Constraints (2)

D A constraint is an **organisational or technological requirement**, which restricts the way the system shall be developed.

Based on [Robertson and Robertson 2013]

Constraints (3)

E

Constraint		Type of Constraint
🔒 C-1:	Due to current conditions defined by the insurance company, only the security technician is allowed to deactivate the control function of the system.	organisational
🔒 C-2:	A fire protection requirement demands that the terminals in the sales rooms do not exceed the size 120 cm (height) × 90 cm (width) × 20 cm (depth).	physical
🔒 C-3:	The user interface shall not contain symbols or graphics abusive for any culture.	culture

Constraints (3)

E

Constraint	Type of Constraint
🔒 C-4: The system shall process personal data in compliance with the EU’s Data Protection Directive 95/46/EC.	legal
🔒 C-5: The effort for system development shall not exceed 480 person months.	project

Summary

- Requirements engineering is a critical success factor for software-intensive systems.
- Three main goals of requirements engineering;
derived from the three dimensions (content, documentation and agreement).
- Three types of requirements:
Functional requirements, quality requirements and constraints.

Literature (1)

[Boehm and Basili 2001] B. Boehm, V. Basili: Software Defect Reduction Top 10 List. IEEE Computer, Vol. 34, No. 1, IEEE Computer Society, Los Alamitos, 2001, pp. 135-137.

[Brooks 1987] F.P. Brooks: No Silver Bullet – Essence and Accidents of Software Engineering. IEEE Computer, Vol. 20, No. 4, IEEE Computer Society, 1987, pp. 10-19.

[ESI 1996] European Software Engineering Institute: European User Survey Analysis. Technical Report, ESI-1996-TR95104, 1996.

[Hall et al. 2002] T. Hall, S. Beecham, A. Rainer: Requirements Problems in Twelve Companies – An Empirical Analysis. In: Proc. 6th Int. Conf. Empirical Assessment and Evaluation in Software Engineering (EASE 2002), Keele University, 2002.

[ISO/IEC 25010-2011] ISO/IEC 25010:2011: Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models, International Organization for Standardization (ISO), 2011.

Literature (2)

[ISO/IEC/IEEE 24765:2017] Institute of Electrical and Electronics Engineers: ISO/IEC/IEEE 24765 - Systems and Software Engineering – Vocabulary. IEEE, 2017.

[Pohl and Ulfat-Bunyadi 2013] K. Pohl, N. Ulfat-Bunyadi: The Three Dimensions of Requirements Engineering: 20 Years Later. In J. Bubenko et al., eds. Seminal Contributions to Information Systems Engineering. Springer Berlin Heidelberg, pp. 81–87, 2013.

[Robertson and Robertson 2006] S. Robertson, J. Robertson: Mastering the Requirements Process. 2nd edition, Addison-Wesley, Amsterdam, 2006.

[The Standish Group 1995] The Standish Group Report: CHAOS. The Standish Group, 1995.

[Wikipedia 2014] Wikipedia: CHAOS-Studie. URL: https://de.wikipedia.org/wiki/Chaos-Studie, Accessed on 19/10/2015, 2014.

Image References

[1] Licensed by http://www.iconshock.com/

[2] Provided by Microsoft Office

Legend

D Definition

E Example

UNIVERSITÄT
DUISBURG
ESSEN

Offen im Denken

Requirements Engineering & Management

Vielen Dank für Ihre Aufmerksamkeit

© SSE, Prof. Dr. Klaus Pohl

PALUNO
The Ruhr Institute for Software Technology