

Martin Glinz

Requirements Engineering I

Kapitel 2

Darstellung von Anforderungen



Universität Zürich
Institut für Informatik

Der Gestaltungsraum

- Darstellungsaspekte
 - Funktionalität: Funktionen, Daten und Verhalten
 - Attribute: Leistungen, Qualitäten, Randbedingungen

- Freiheitsgrade
 - Wahl der Mittel
 - Deskriptive oder konstruktive Darstellung
 - Art der Gliederung / Strukturierung
 - Präzision
 - Tiefe

2.1 Darzustellende Aspekte

Darzustellende **Aspekte** (unabhängig von den verwendeten Gliederungs- und Darstellungsmethoden):

- **Funktionaler Aspekt**
 - **Daten:** Struktur, Verwendung, Erzeugung, Speicherung, Übertragung, Veränderung
 - **Funktionen:** Ausgabe, Verarbeitung, Eingabe von Daten
 - **Verhalten:** Sichtbares dynamisches Systemverhalten, Zusammenspiel der Funktionen (untereinander und mit den Daten)
 - **Fehler:** Normalfall und Fehlerfälle

Darzustellende Aspekte – 2

○ Leistungsaspekt

- Datenmengen (durchschnittlich/im Extremfall)
- Verarbeitungs- /Reaktionsgeschwindigkeit (durchschnittlich/im Extremfall)
- Verarbeitungszeiten und -intervalle
- Wo immer möglich: messbare Angaben!

○ Qualitätsaspekt

- Geforderte besondere Qualitäten (z.B. Benutzerfreundlichkeit, Zuverlässigkeit)

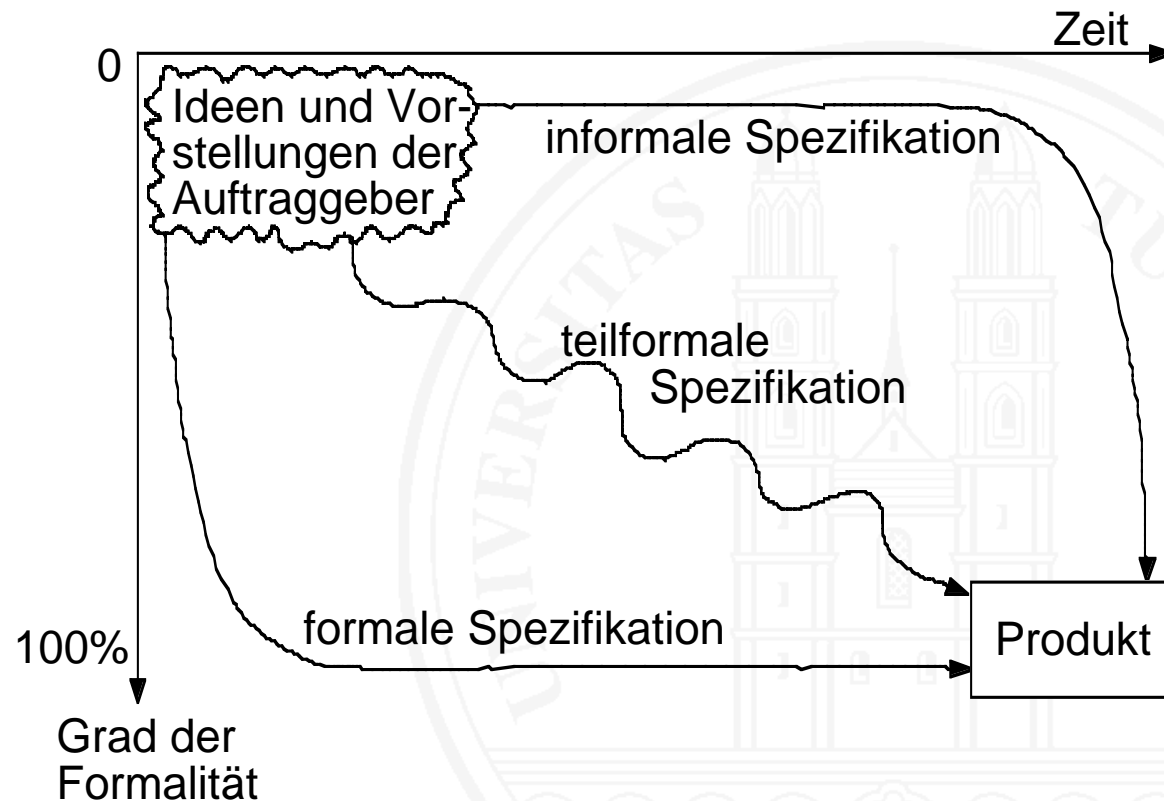
Darzustellende Aspekte – 3

- **Randbedingungsaspekt**
 - Einzuhaltende/zu verwendende **Schnittstellen**
 - Normen und **Gesetze**
 - **Datenschutz, Datensicherung**
 - **Explizite Vorgaben** des Auftraggebers

2.2 Wahl der Mittel

- Texte in natürlicher Sprache
 - Strukturmodelle
 - Interaktionsmodelle
 - Formale Modelle auf der Grundlage mathematisch-logischer Formalismen
 - Darstellung mit unterschiedlichem Formalitätsgrad
- in der Regel grafisch, angereichert mit natürlicher Sprache

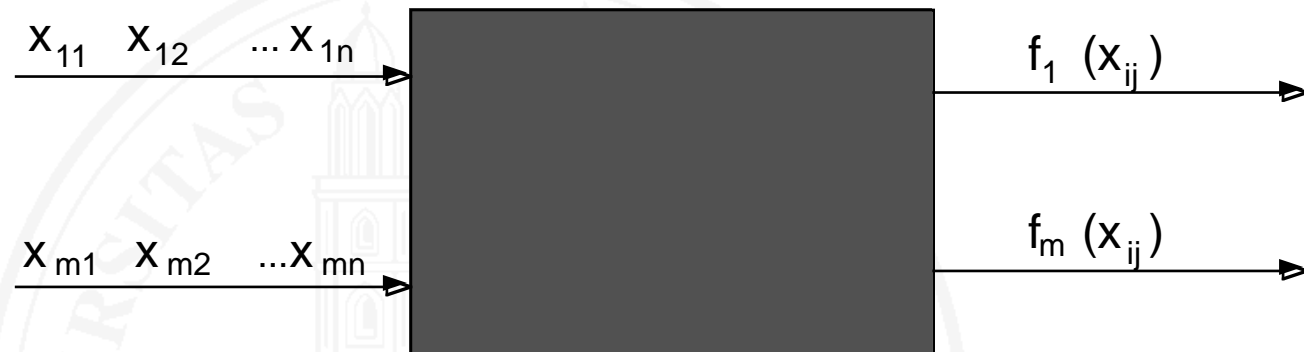
Formalitätsgrad der Darstellung



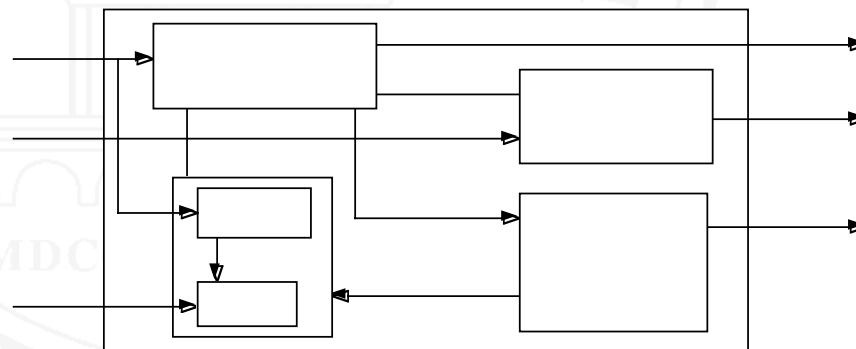
- **informal**, in der Regel deskriptiv mit natürlicher Sprache
- **formal**, deskriptive und konstruktive Verfahren möglich
- **teilformal** mit konstruktiven, anschaulichen Modellen

2.3 Deskriptive oder konstruktive Darstellung

Deskriptiv:



Konstruktiv:



Deskriptive Darstellung – 1

- Das zu spezifizierende System als „schwarzer Kasten“
- Darstellung durch Beschreibung der Zusammenhänge zwischen den geforderten Resultatdaten und den gelieferten Eingabedaten

Beispiele

- Darstellung **mit Text** in natürlicher Sprache:
«Die Funktion Kontostand liefert den aktuellen Stand des Kontos für die eingegebene Kontonummer.»
- Darstellung in einer **formalen** Notation:
Sqrt: Real \rightarrow Real;
Pre: $x \geq 0$;
Post: $|\text{Sqrt}^2(x) - x| < \varepsilon \wedge \varepsilon \leq 10^{-16} \wedge \varepsilon \leq 10^{-6}x$.

Deskriptive Darstellung – 2

○ Vorteile:

- + Nur **äußeres Verhalten** spezifiziert
- + **Lösungsneutral**

○ Nachteile:

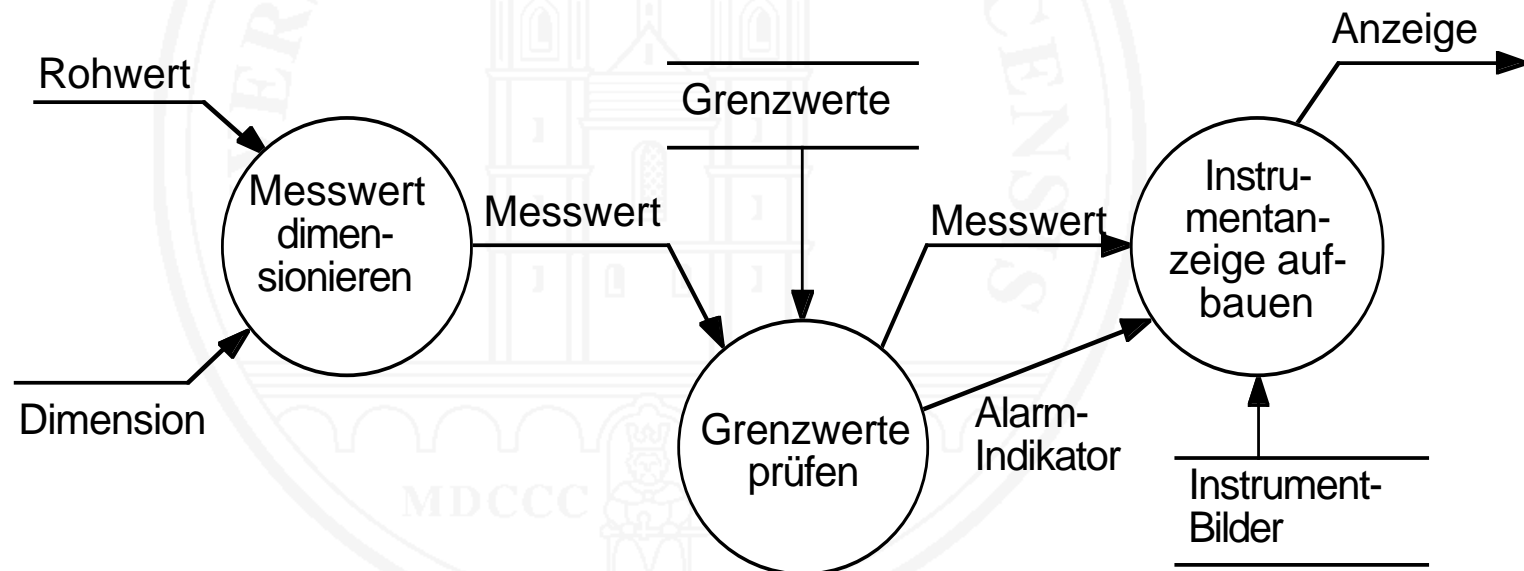
- Bei Verwendung von Text: umfangreich und **wenig strukturiert**; Zusammenhänge nicht erkennbar; **fehlerträchtig** und schwierig zu prüfen
- Bei Verwendung formaler Mitteln: **sehr schwierig zu erstellen**; **Prüfung auf Adäquatheit oft fast unmöglich**

⇒ Nur für die Darstellung der Anforderungen **kleiner, überschaubarer Teilprobleme** geeignet

Konstruktive Darstellung – 1

- Zu spezifizierendes System als eine Menge interagierender Komponenten modelliert

Beispiel



Konstruktive Darstellung – 2

Vorteile:

- + **Anschauliches** Modell der Problemstellung; leicht verstehbar und nachvollziehbar
- + Ermöglicht die **Zerlegung** der Gesamtaufgabe in kleinere, besser überschaubare Teilaufgaben
- + Kombination unterschiedlich stark formalisierter Teile möglich
- + Das Modell ist **idealisierte Lösung**: Tatsächliche Lösung oft analog strukturierbar

Nachteile:

- Modell ist **idealisierte Lösung**. Gefahr von
 - implementierungsorientierter Spezifikation
 - Implementierung suboptimaler Lösungen
- ⇒ Vor allem für die Modellierung von **Anforderungen im Großen** geeignet

2.4 Aufbau einer Anforderungsspezifikation

- Keine fixen Vorgaben
- Teilweise unternehmensinterne Standards
- IEEE 830-1993 als öffentlicher Standard
- Hier: drei Beispiele:
 - Einfache Struktur, angelehnt an IEEE 830-1993
 - Volere-Schablone von Robertson und Robertson (1999)
 - Spezifikationsbausteine von sd&m

Einfache Struktur, angelehnt an IEEE 830-1993

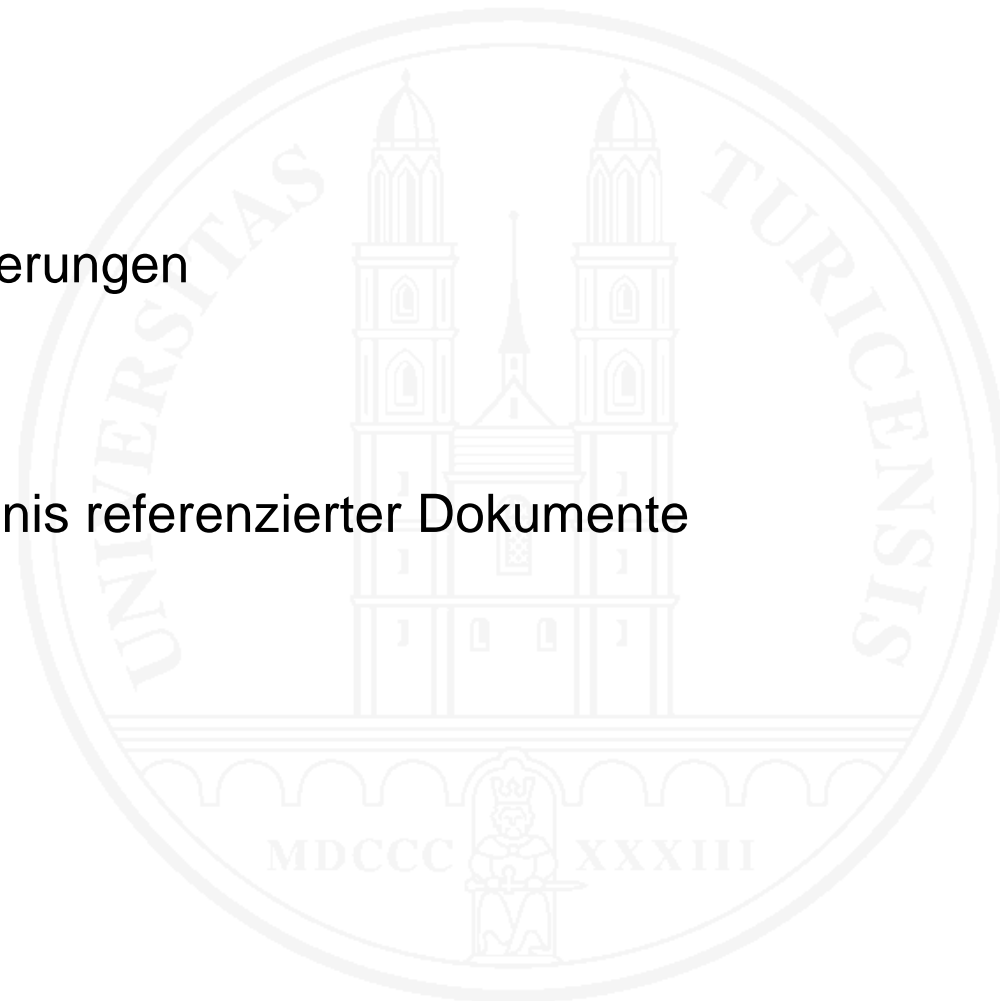
1. Einleitung

2. Überblick

3. Einzelanforderungen

Anhänge

- Glossar
- Verzeichnis referenzierter Dokumente



Einfache Struktur, angelehnt an IEEE 830-1993 – 2

1. Einleitung

1.1 Anlass

Wozu das Dokument benötigt wird: Projekt, Kunde, Umfeld, ...

1.2 Ziele

Die Zielsetzung für das spezifizierte System im Überblick; bei größeren Systemen mehrstufig: Ziele, Teilziele

1.3 Einsatzbereich

Wo das spezifizierte System verwendet wird: Umgebung, Zielplattform,...

Einfache Struktur, angelehnt an IEEE 830-1993 – 3

2. Überblick

2.1 Kontext

Betrachtungsebene, Kontext des spezifizierten Systems

2.2 Struktur der Problemstellung

Sachliche Struktur der Problemstellung

Gegebenenfalls Gliederung in Teilprobleme

2.3 Globale Attribute

Systemweit gültige Attribute (Leistungsanforderungen, besondere Qualitäten, Randbedingungen)

2.4 Annahmen

Annahmen, auf denen das spezifizierte System basiert

2.5 Perspektiven

Entwicklungsperspektiven für das spezifizierte System

Einfache Struktur, angelehnt an IEEE 830-1993 – 4

3. Einzelanforderungen

Auflistung aller Anforderungen

Sinnvoll gegliedert, Möglichkeiten:

- Hierarchisches Objektmodell: Hierarchische Gliederung des Problems in geschlossene Teilprobleme
 - Unterkapitelgliederung; die Unterkapitel fassen logisch zusammengehörige Teile zusammen
- Anhänge
- Glossar
 - Verzeichnis aller verwendeten Fachbegriffe mit Definitionen, Abkürzungen, Synonymen, etc.
 - Verzeichnis referenzierter Dokumente
 - Nachweis aller verwendeten, mitgeltenden Dokumente

Volere-Schablone (Robertson und Robertson 1999)

Project Drivers

1. The Purpose of the Project
2. Client, Customer & other Stakeholders
3. Users of the Product

Project Constraints

4. Mandated Constraints
5. Naming Conventions and Definitions
6. Relevant Facts and Assumptions

Functional Requirements

7. The Scope of the Work
8. The Scope of the Product
9. Functional and Data Requirements

Non-Functional Requirements

10. Look and Feel Requirements
11. Usability and Humanity Requirements
12. Performance Requirements
13. Operational Requirements

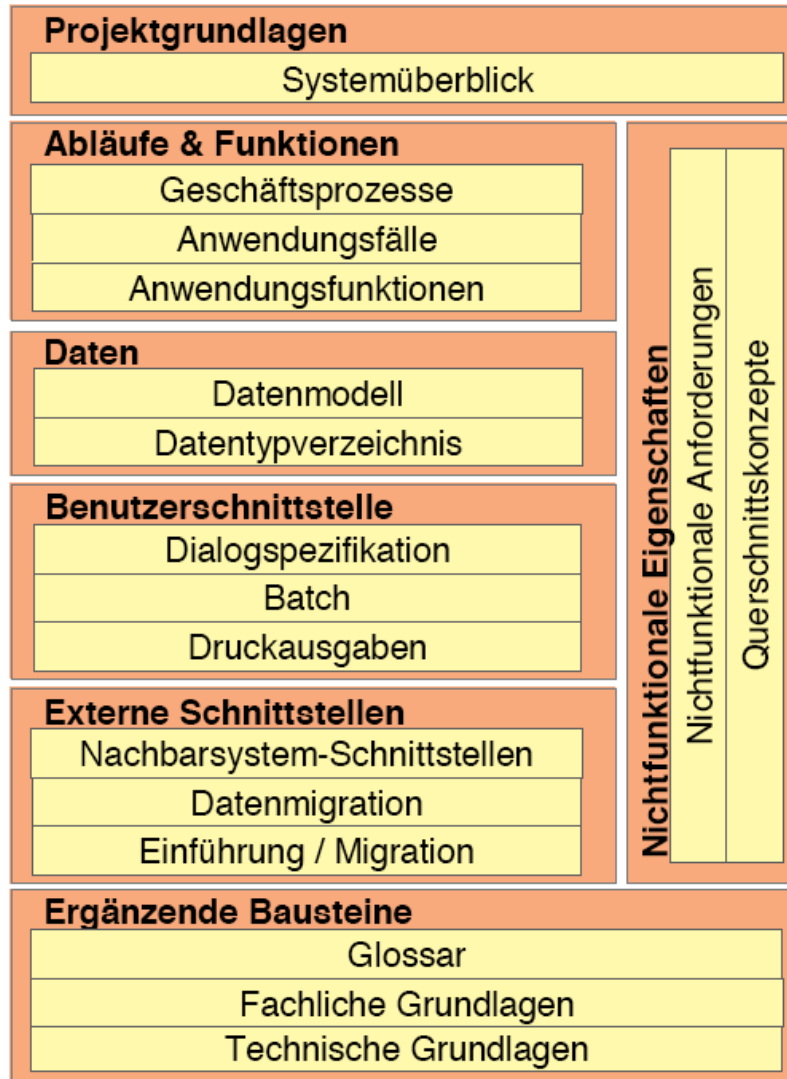
Project Issues

14. Maintainability and Support Requirements
15. Security Requirements
16. Cultural and Political Requirements
17. Legal Requirements

Project Issues

18. Open Issues
19. Off-the-Shelf Solutions
20. New Problems
21. Tasks
22. Cutover
23. Risks
24. Costs
25. User Documentation and Training
26. Waiting Room
27. Ideas for Solutions

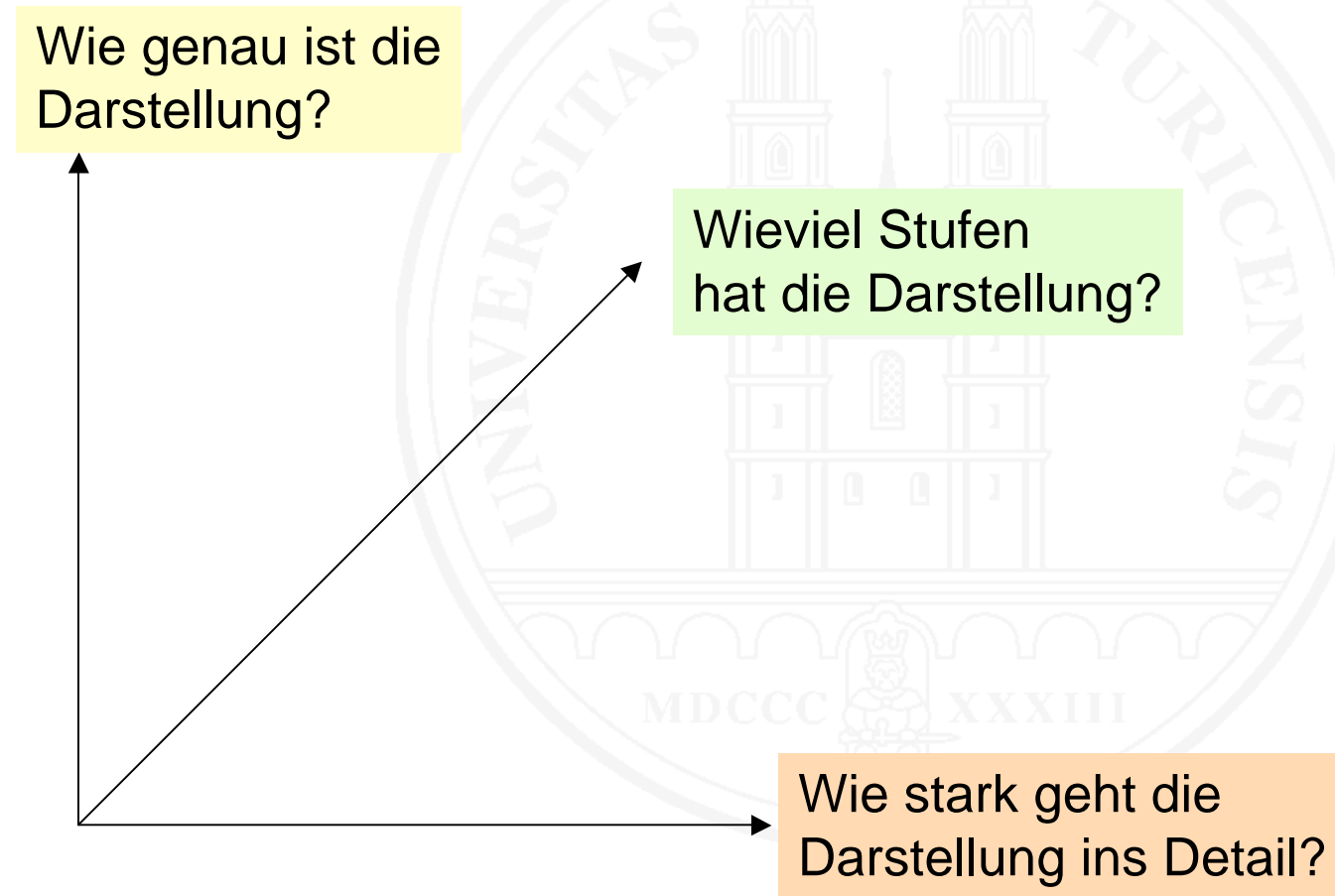
Spezifikationsbausteine von sd&m¹⁾



¹⁾ aus: Andreas Birk (2004). Anforderungsspezifikation in großen IT-Projekten. *Jahrestagung der GI-Fachgruppe Requirements Engineering*, Kaiserslautern.

2.5 Präzision, Detaillierung und Tiefe

Drei Dimensionen:



Präzision

- Anforderungen **sprachlich präzise** ausdrücken, beispielsweise durch
 - Aussagekräftige Namen
 - Definierte Subjekte: kein „man“, kein „es wird“
 - Definierte Bedeutungen für „kann“, „muss“, „soll“, etc.

- Anforderungen **inhaltlich präzise** beschreiben
 - Eindeutige Formulierung
 - Hinreichend detailliert
 - Prüfbar
 - testbar
 - quantifiziert → messbar

Präzision – 2

Beispiele:

vage Anforderungen

«schnell»

«Für alle Kurse müssen Teilnehmerlisten erstellt werden»

präzise Anforderungen

« $< 0,2 \text{ s}$ »

«10 Arbeitstage vor Beginn eines Kurses soll das System die Teilnehmerliste für den Kurs erzeugen»

Aufgabe 2.1: Wieviel Detail?

Welche Variante ist besser:

- A. «Die Teilnehmer-Eingabemaske enthält Felder für Name, Vorname, Geschlecht und Adresse des Teilnehmers. »
- B. «Die Teilnehmer-Eingabemaske enthält Felder für Name, Vorname, Geschlecht und Adresse des Teilnehmers. Namen- und Vornamenfelder sind je maximal 32 Zeichen lang und obligatorisch. Das System verwendet Unicode als Zeichensatz. Für die Eingabe des Geschlechts enthält die Maske zwei Ankreuzfelder, beschriftet mit männlich und weiblich. Die Voreinstellung ist männlich, Ankreuzungen schließen sich gegenseitig aus, eine Ankreuzung ist erforderlich. ...»

Detallierung

Der notwendige Detaillierungsgrad wird bestimmt durch Abwägung

- der Kosten
- des Risikos, unbrauchbare Systeme zu erhalten
- des Entscheidungsspielraums für die Entwickler

⇒ Risikogerechte Detaillierung anstreben

⇒ Mehr Präzision erfordert typisch mehr Details

⇒ Mehr Details erfordern typisch mehr Tiefe in der Darstellung

Tiefe

- Je präziser und detaillierter, desto umfangreicher
- Lesbarkeit erhalten durch Strukturierung in die Tiefe:

4.3 Teilnehmerverwaltung

4.3.1 Erfassen neuer Teilnehmer

4.3.1.1 Eingabemaske

4.3.1.1.1 Name: maximal 32 Zeichen, Unicode-Zeichensatz, obligatorisch

4.3.1.1.2 Vorname: maximal 32 Zeichen, Unicode-Zeichensatz, obligatorisch

4.3.1.1.3 Geschlecht: zwei Ankreuzfelder, beschriftet mit männlich und weiblich. Voreinstellung: männlich
Ankreuzungen schließen sich gegenseitig aus, eine Ankreuzung erforderlich

...

4.3.1.2 Eingabebestätigung

...

2.6 Regeln für die Darstellung von Anforderungen

- Einzelanforderungen in **kleinen Einheiten** fassen
- Zu jedem geforderten **Resultat** die **Funktion** und die **Eingabedaten**, welche das Resultat erzeugen, spezifizieren
- Mögliche **Ausnahmesituationen** spezifizieren
- **Implizite Annahmen aufdecken** und explizit formulieren
- Leistungs- und Qualitätsanforderungen **quantitativ** spezifizieren
- Anforderungen **strukturieren**
- **Redundanz** nur dort, wo für leichtes Verständnis notwendig

Literatur

Birk, A. (2004). Anforderungsspezifikation in großen IT-Projekten. *Jahrestagung der GI-Fachgruppe Requirements Engineering*, Kaiserslautern.

Glinz, M. (2004). *Software Engineering I*. Vorlesungsskript, Universität Zürich.

IEEE (1998). *IEEE Recommended Practice for Software Requirements Specifications*. IEEE Standard 830-1998.

Robertson, S., Robertson, J. (1999). *Mastering the Requirements Process*. Addison-Wesley.

