

Slides of the presentation held at the **Software & Systems Quality Conferences International 2007 in Düsseldorf**

Copyright © [2007] Dr. Andrea Herrmann

Priorisierung von Qualitätsanforderungen auf der Basis von Risikoabschätzungen

Andrea Herrmann



Institut für Informatik

Neuenheimer Feld 326

D-69120 Heidelberg, Germany

<http://www-swe.informatik.uni-heidelberg.de>

herrmann@informatik.uni-heidelberg.de

RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG





- ◆ **Werdegang:**
 - 6 Jahre als Beraterin und Projektleiterin in IT-Projekten
 - Seit 2 Jahren wissenschaftliche Mitarbeiterin an der Universität Heidelberg

- ◆ **Interessengebiete:**
 - Anforderungen
 - Qualitätsanforderungen
 - Schnittstelle zu Entwurf
 - Projekt- und Wissensmanagement

Motivation: Qualitätsanforderungen

Qualitätsanforderungen

- ▶ Motivation
- Anforderungs-
bewertung
- Entwurfs-
entscheidungen
- Lessons
Learned

Erlernbarkeit (QA)

Benutzerfreundlichkeit (QA)

FA 1

FA 2

FA 3

FA 4

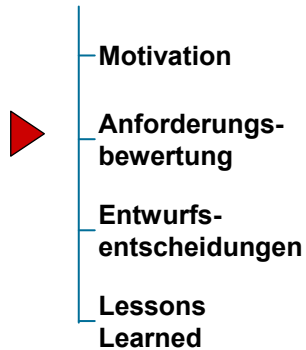
FA = funktionale Anforderung
QA = Qualitätsanforderung

Qualitätsanforderungen

- Motivation
- Anforderungs-
bewertung
- Entwurfs-
entscheidungen
- Lessons
Learned



Qualitätsanforderungen



- ◆ **Anforderungsbewertung** <- Risikoabschätzungen, z.B. in MOQARE
- ◆ **Entwurfsentscheidungen** <- Anforderungsbewertungen
- ◆ **Lessons Learned** aus Fallstudien und Anwendungsbeispielen

Bewertung der Benutzerfreundlichkeit eines Werkzeugs für Anforderungsspezifikation und Entwurf

Qualitätsanforderungen



Motivation

Anforderungs-
bewertung

Entwurfs-
entscheidungen

Lessons
Learned



Document:ReqSpecMUC

Path: ReqSpecMUC
 Actions: [Refresh](#) [All](#) [Index](#) [Printable Version](#) | [Help](#)

1. Task Level

- [1.1. Problem Statement](#)
- [1.2. Actors and Stakeholders](#)
- [1.3. User Tasks](#)
- [1.4. Quality Constraints on User Tasks](#)
- [1.5. Business Goals](#)
- [1.6. Business Damages](#)

2. Domain Level

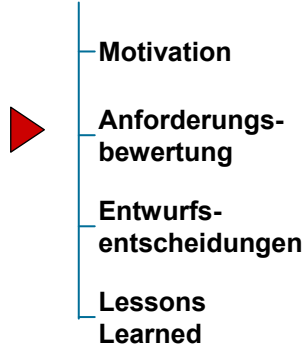
- [2.1. Domain Data](#)
- [2.2. Domain Constraints](#)
- [2.3. Activities as is](#)
- [2.4. Activities to be](#)
- [2.5. System Responsibilities](#)
- [2.6. Quality Constraints on System Responsibilites](#)

3. Interaction Level

- [3.1. UI Structure](#)
- [3.2. Workspaces](#)
- [3.3. User Interface Constraints](#)
- [3.4. Use Cases](#)

Im Beispiel durchgeführte Schritte

Qualitätsanforderungen

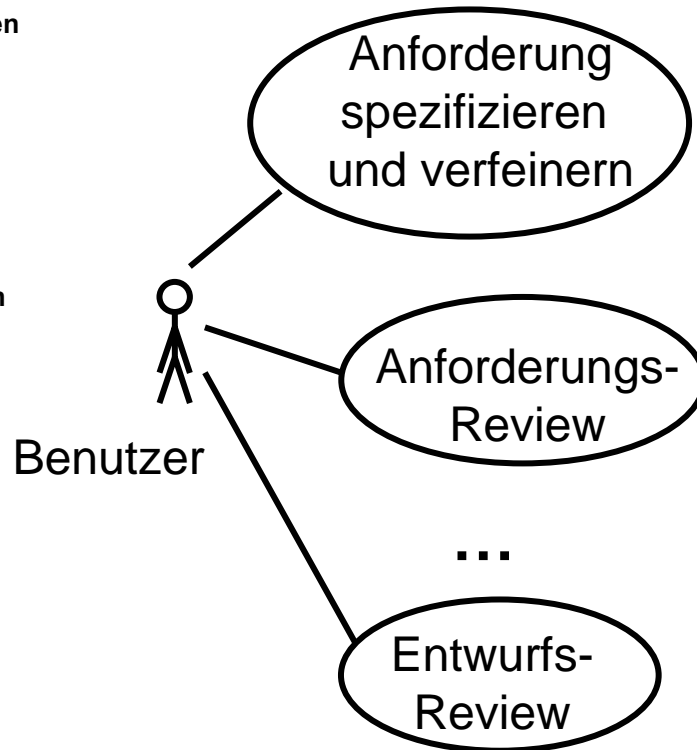


- ◆ Funktionale Anforderungen (FA)
- ◆ Konkretisierung des QA „Benutzerfreundlichkeit“
- ◆ Bewertung der QA
- ◆ Test der Software gegenüber den QA
- ◆ Versionierung der Weiterentwicklung

Funktionale Anforderungen (FA)

Qualitätsanforderungen

- Motivation
- ▶ Anforderungs-
bewertung
- Entwurfs-
entscheidungen
- Lessons
Learned



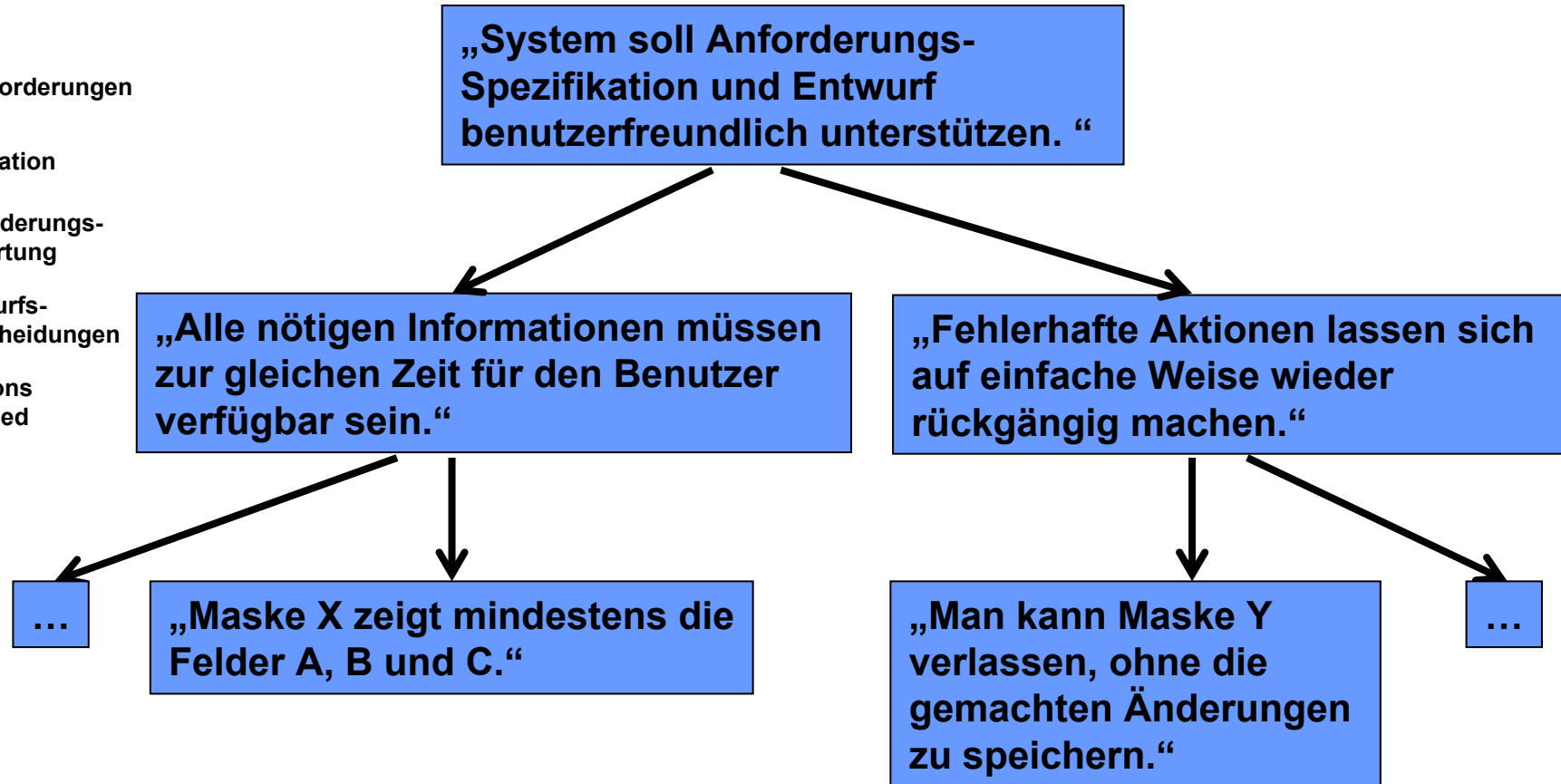
<i>Nutzersichtbündel</i>	<i>Punkte</i>
■ TORE (FA)	30
■ MOQARE (QA)	20
■ ICRAD (Entwurf)	50

Insgesamt 27 Use Cases

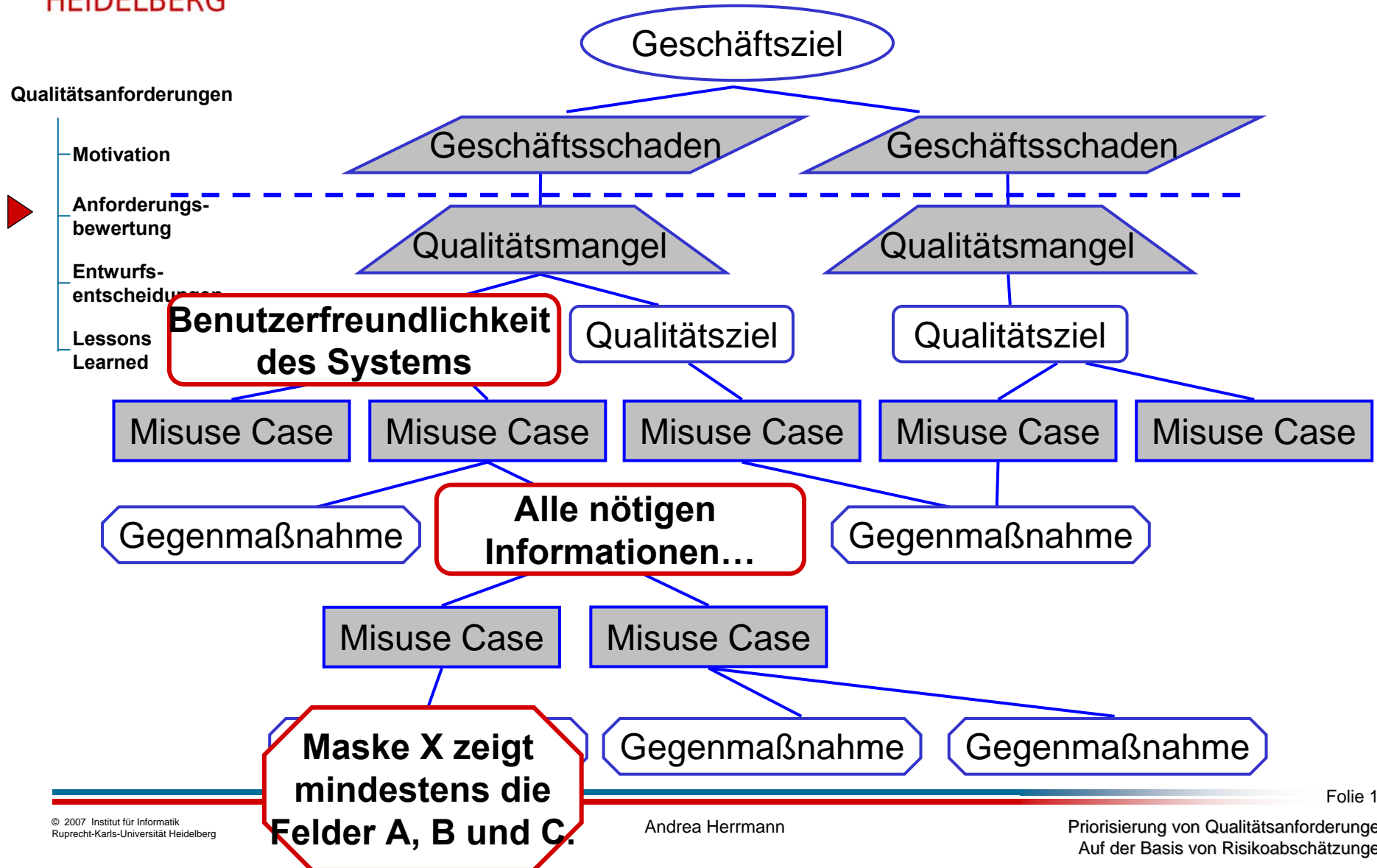
Konkretisierung der Qualitätsanforderungen

Qualitätsanforderungen

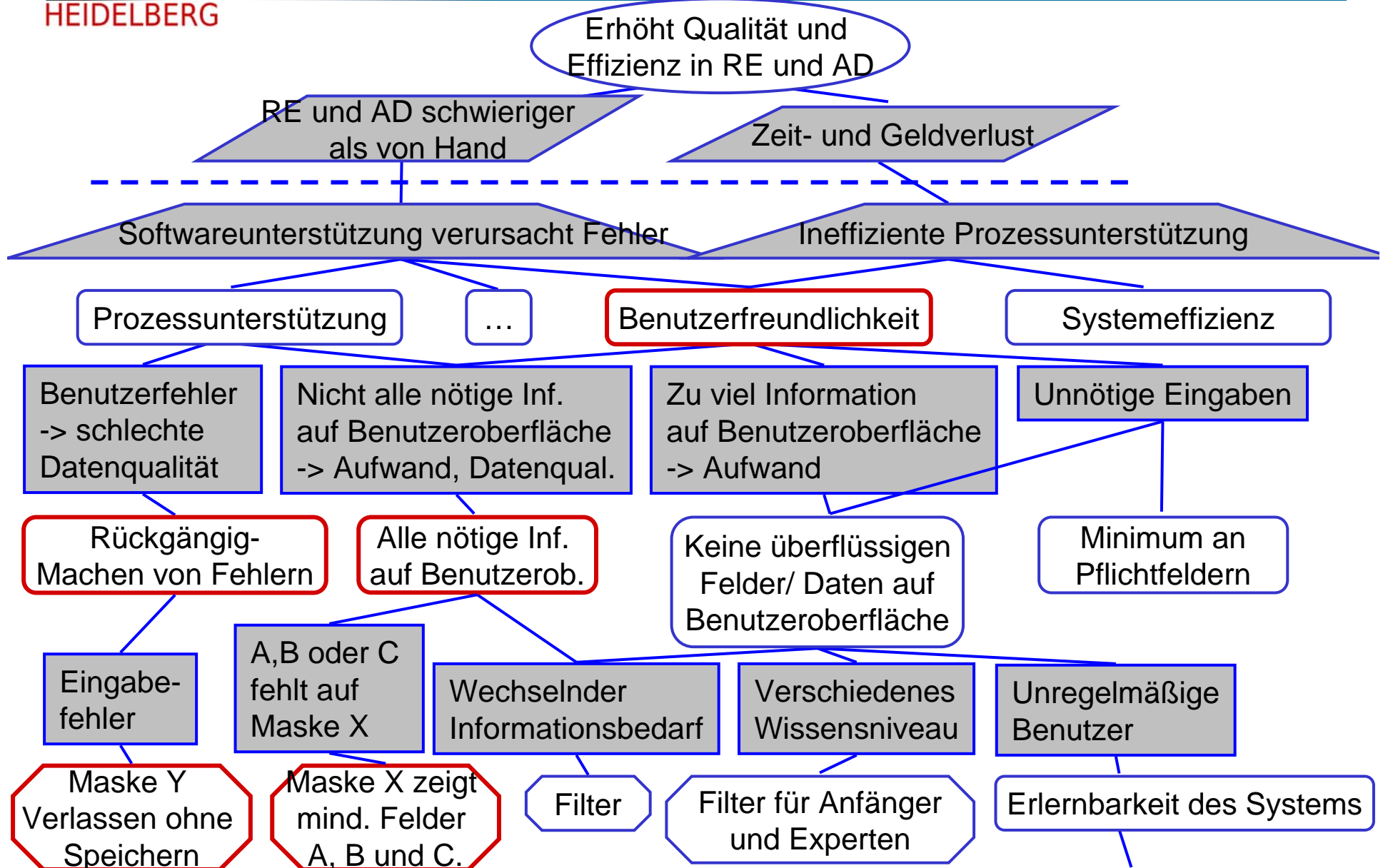
- Motivation
- ▶ Anforderungs-
bewertung
- Entwurfs-
entscheidungen
- Lessons
Learned



Konkretisierung & Abhängigkeiten in MOQARE



MOQARE Misuse Tree Ausschnitt



Was bedeutet „wichtig“? Bewertungskriterien

Qualitätsanforderungen



Motivation

Anforderungs-
bewertung

Entwurfs-
entscheidungen

Lessons
Learned

- Nutzen
- Kosten; (Kalender-)Zeit
- Bedeutung der Quelle
- Risiko, z.B. Fehleranfälligkeit
- Dringlichkeit, Sanktion
- Volatilität

Nutzen – Kosten-
Abwägung

Bewertung in MOQARE

Faustformel: Nutzen umso höher,
je näher zu den Geschäftszielen,
d.h. in höherer Ebene

Erhöht Qualität
Effizienz in RE

RE und AD schwieriger
als von Hand

Zeit- und Geldverlust

Softwareunterstützung verursacht Fehler

Ineffiziente Prozessunterstützung

Prozessunterstützung

...

Benutzerfreundlichkeit

Systemeffizienz

Benutzerfehler
-> schlechte
Datenqualität

Nicht alle nötige Inf.
auf Benutzeroberfläche
-> Aufwand

Zu viel Information
auf Benutzeroberfläche
-> Zeitverlust

Unnötige Eingaben

Rückgängig-
Machen von Fehlern

Alle nötige Inf.
auf Benutzerob.

Keine überflüssige
Felder/ Daten auf
Benutzeroberfläche

Minimum an
Pflichtfeldern

Eingabe-
fehler

A,B oder C
fehlt auf
Maske X

Wechselnder
Informationsbedarf

Verschiedenes
Wissensniveau

Unregelmäßige
Benutzer

Maske Y
Verlassen ohne
Speichern

Maske X zeigt
mind. Felder
A, B und C.

Filter

Filter für Anfänger
und Experten

Erlernbarkeit des Systems

Bewertung in MOQARE

graue (unerwünschte)
Elemente: Risiko
= Wahrscheinlichkeit
x Nutzenverlust

Erhöht Qualität und
Effizienz in RE und AD

RE und AD schwieriger
als von Hand

Zeit- und Geldverlust

Softwareunterstützung verursacht Fehler

Ineffiziente Prozessunterstützung

Prozessunterstützung

...

Benutzerfreundlichkeit

Systemeffizienz

Benutzerfehler
-> schlechte
Datenqualität

Nicht alle nötige Inf.
auf Benutzeroberfläche
-> Aufwand

Zu viel Information
auf Benutzeroberfläche
-> Zeitverlust

Unnötige Eingaben

Rückgängig-
Machen von Fehlern

Alle nötige Inf.
auf Benutzeroberf.

Keine überflüssige
Felder/ Daten auf
Benutzeroberfläche

Minimum an
Pflichtfeldern

Eingabe-
fehler

A,B oder C
fehlt auf
Maske X

Wechselnder
Informationsbedarf

Verschiedenes
Wissensniveau

Unregelmäßige
Benutzer

Maske Y
Verlassen ohne
Speichern

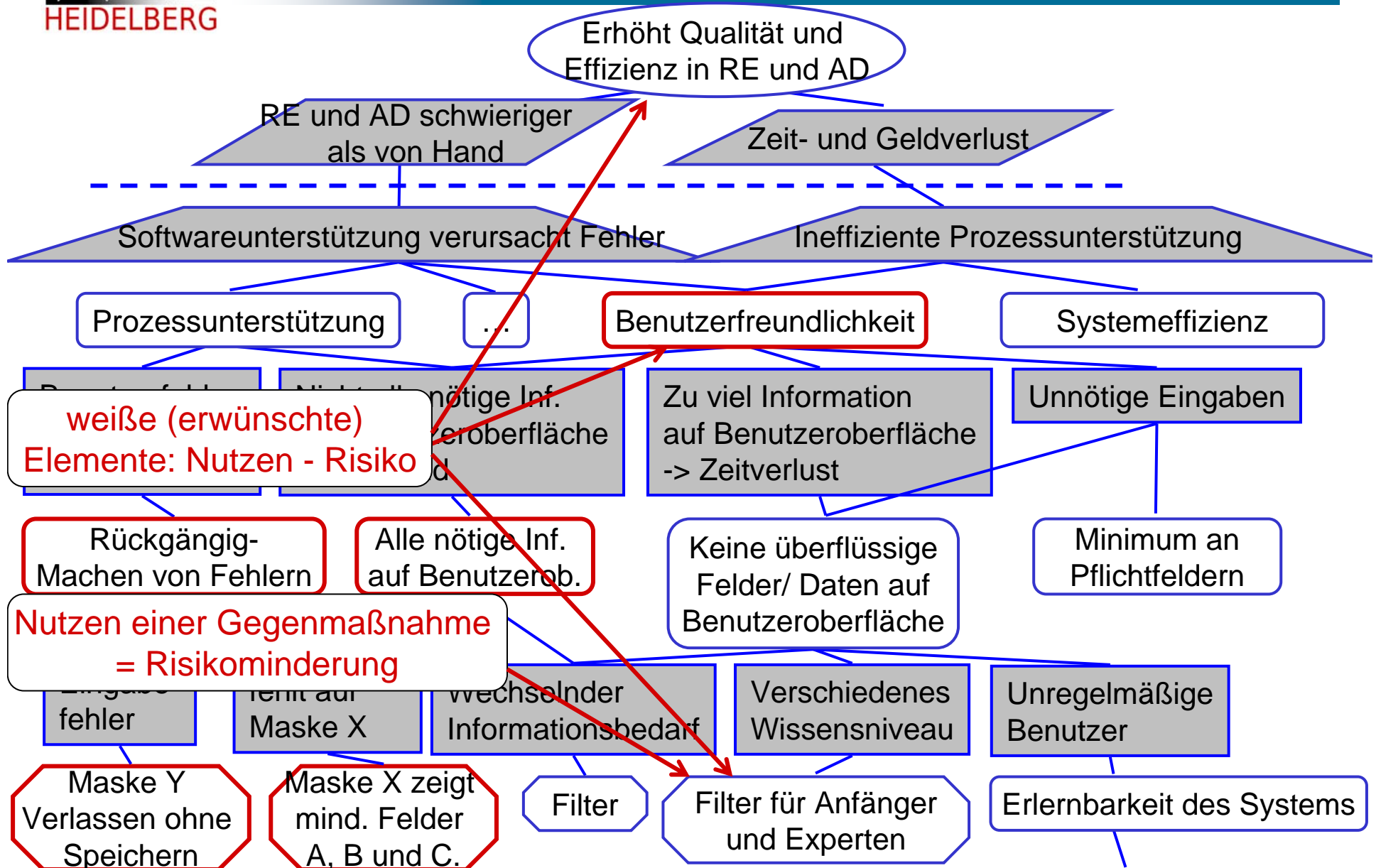
Maske X zeigt
mind. Felder
A, B und C.

Filter

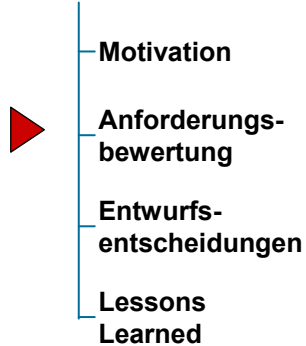
Filter für Anfänger
und Experten

Erlernbarkeit des Systems

Bewertung in MOQARE

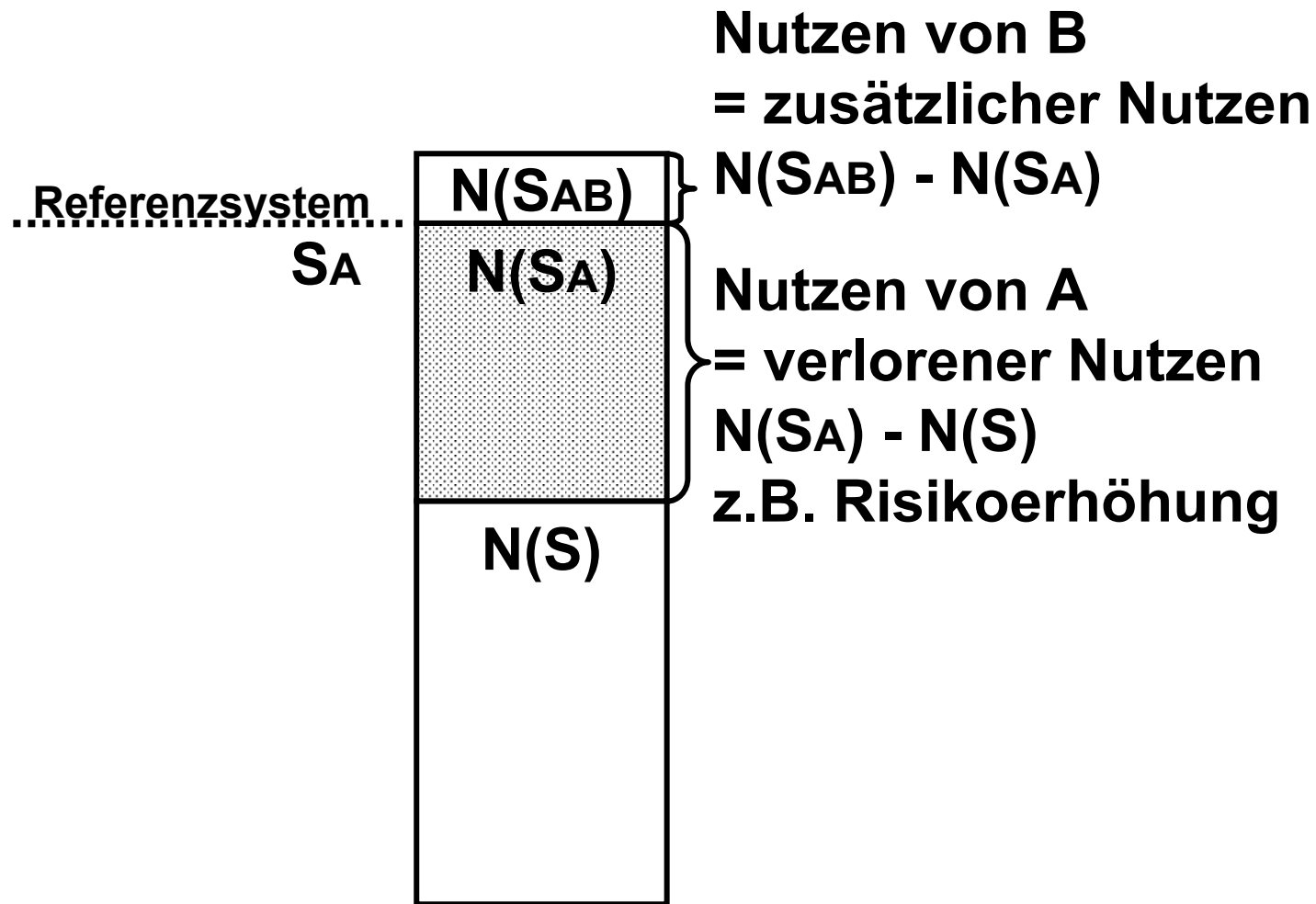


Qualitätsanforderungen



Risiko abhängig von vielen Faktoren:

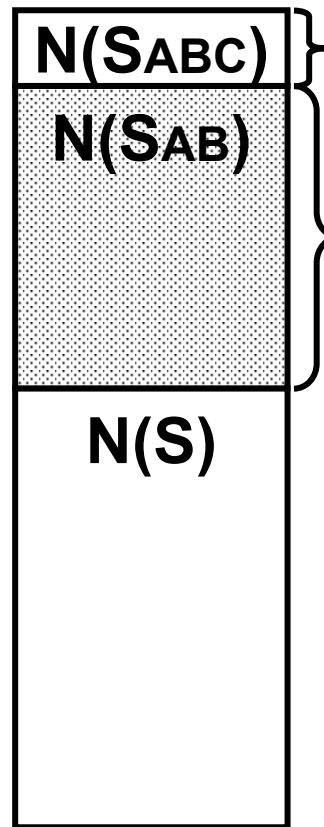
- Umfeld: Firma, Kunden, usw.
- Zeithorizont
- System (Referenzsystem)
- andere Risiken bzw. Anforderungen



Referenzsystem: Achtung!

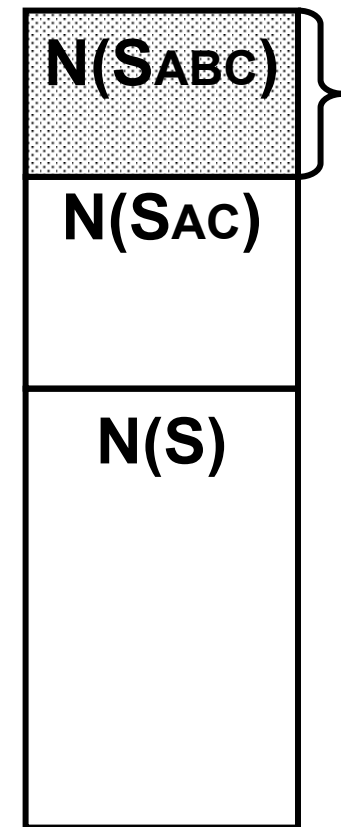
Nutzen von B
= $N(S_{ABC}) - N(S_{AC})$!

Referenzsystem =
Perfektes System
SABC



Nutzen von C
= $N(S_{ABC}) - N(S_{AB})$

Nutzen von B
= $N(S_{AB}) - N(S_A)?$



Nutzen von A + Nutzen von B + Nutzen von C \neq $N(S_{ABC})$!

Qualitätsanforderungen



Motivation

Anforderungs-
bewertung

Entwurfs-
entscheidungen

Lessons
Learned

- Test ->
effektiver Nutzen des Systems
= Nutzen des Referenzsystems – Risiko
- Versionierung der Gegenmaßnahmen:
höchste Priorität, wenn noch nicht umgesetzt
& höchster Nutzen

Qualitätsanforderungen

Motivation

Anforderungs-
bewertung

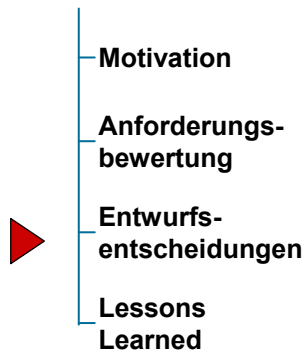
Entwurfs-
entscheidungen

Lessons
Learned

- ◆ **Anforderungsbewertung** <- Risikoabschätzungen, z.B. in MOQARE
- ◆ **Entwurfentscheidungen** <- Anforderungsbewertungen
- ◆ **Lessons Learned** aus Fallstudien und Anwendungsbeispielen

ICRAD: Bewertungskriterien

Qualitätsanforderungen



- ◆ Nutzen
 - ◆ Risiko
 - ◆ Implementierungskosten
 - ◆ Wartungskosten
- } Nutzen – Risiko
= effektiver Nutzen
- } IK + WK
= Gesamtkosten
- ◆ Nettowert = effektiver Nutzen - Gesamtkosten
 - ◆ Nutzen-Kosten-Verhältnis = effektiver Nutzen /
Gesamtkosten

ICRAD: Vergleich v Entwurfsalternativen

	Alternative I	Alternative II	Unterschied (II)-(I)
Implement.kosten	C1	C2	C2-C1
Wartungskosten	CC1	CC2	CC2-CC1
Risiko	R1	R2	R2-R1
Nutzen	N1	N2	N2-N1
Effektiver Nutzen	N1-R1	N2-R2	(N2-R2)-(N1-R1) = ΔEB
Gesamtkosten	C1+CC1	C2+CC2	(CC2-CC1)+(C2-C1) = ΔTC
Nettowert	$(N1-R1)-(C1+CC1)$	$(N2-R2)-(C2+CC2)$	$(N2-R2)-(C2+CC2) - (N1-R1) + (C1+CC1)$
Effektiver Nutzen/ Gesamtkosten	$(N1-R1) / (C1+CC1)$	$(N2-R2) / (C2+CC2)$	$\Delta EB / \Delta TC$ = $[(N2-R2)-(N1-R1)] / [(CC2-CC1)+(C2-C1)]$

Qualitätsanforderungen

Motivation

Anforderungs-
bewertung

Entwurfs-
entscheidungen

Lessons
Learned

- ◆ **Anforderungsbewertung** <- Risikoabschätzungen, z.B. in MOQARE
- ◆ **Entwurfsentscheidungen** <- Anforderungsbewertungen
- ◆ **Lessons Learned** aus Fallstudien und Anwendungsbeispielen

Lessons Learned aus Fallstudien

Qualitätsanforderungen

Motivation

Anforderungs-
bewertung

Entwurfs-
entscheidungen

Lessons
Learned

- ◆ **Anforderungserfassung:** MOQARE-Baum als Interviewleitfaden
- ◆ **Anforderungsbewertung:** unterstützt Entscheidungen (Entwurf, Versionierung, ...), ist aber schwierig -> Bündel bewerten & klares Referenzsystem definieren
- ◆ Herleitung und Bewertung von **Testkriterien** für manuelle Systemtests: misst Qualität und Qualitätsverbesserung
- ◆ **Wissensmanagement** durch Listen, Teilbäume und Statistiken

Qualitätsanforderungen

Motivation

Anforderungs-
bewertung

Entwurfs-
entscheidungen

Lessons
Learned

Anforderungsbewertung verlangt

- ◆ Konkrete Anforderungen
- ◆ Klare Kriterien
- ◆ Berücksichtigung von Risiken
- ◆ Berücksichtigung von Abhängigkeiten
- ◆ Referenzsystem als Messlatte

Anforderungsbewertung unterstützt:

- ◆ Entscheidungen
- ◆ Qualitätsmessung

Qualitätsanforderungen

Motivation

Anforderungs-
bewertung

Entwurfs-
entscheidungen

Lessons
Learned

- ◆ A. Herrmann, B. Paech: „MOQARE = Misuse-oriented Quality Requirements Engineering - Über den Nutzen von Bedrohungsszenarien beim RE von Qualitätsanforderungen“, *Softwaretechnik-Trends* 26:1, Feb. 2006, S. 13-14
- ◆ A. Herrmann, B. Paech: “Software Quality by Misuse Analysis. Technical Report SWEHD-TR-2005-01 (University of Heidelberg, 2005)”, <http://www-swe.informatik.uni-heidelberg.de/research/publications/reports.htm>
- ◆ A. Herrmann, B. Paech, D. Plaza: “ICRAD: An Integrated Process for Requirements Conflict Solution and Architectural Design”, *IJSEKE (International Journal of Software Engineering and Knowledge Engineering)*, Vol. 16, no. 6, Dec. 2006, S. 917-950
- ◆ A. Herrmann, B. Paech: „Lernen aus dokumentierten Architektur-Entscheidungen“, *Softwaretechnik-Trends* 26:4, Nov. 2006, S. 22-27

Vielen Dank für Ihre Aufmerksamkeit!

Qualitätsanforderungen

Motivation

Anforderungs-
bewertung

Entwurfs-
entscheidungen

Lessons
Learned

Andrea Herrmann

**Institut für Informatik
Neuenheimer Feld 326
D-69120 Heidelberg
Germany**

**<http://www-swe.informatik.uni-heidelberg.de>
*herrmann@informatik.uni-heidelberg.de***