

Martin Glinz

Software-Qualität – Ausgewählte Kapitel

Kapitel 6

Qualität definieren und erreichen



Universität Zürich
Institut für Informatik

Inhalt

- 6.1 Der Qualitätsbegriff
- 6.2 Definieren von Qualität
- 6.3 Definierte Qualität erreichen



6.1 Der Qualitätsbegriff

Frage: Ist ein Schirm, der genau einen Regen aushält, von hoher Qualität?

- Landläufig: nein
- Im Sinn des industriellen Qualitätsbegriffs: durchaus möglich
 - ➔ Es kommt auf die Anforderungen an

Qualität (quality) – der Grad, in dem ein Satz inhärenter Merkmale Anforderungen erfüllt. [ISO 9000:2000]

Inhärentes Merkmal (inherent characteristic) – eine kennzeichnende Eigenschaft einer Einheit (Produkt, Dienstleistung, Prozess, System, Person, Organisation, etc.), welche diese aus sich selbst heraus hat und die ihr nicht explizit zugeordnet ist.

Güte

- **Güte** bezeichnet im Deutschen den **intuitiven Qualitätsbegriff**
- Hohe Güte bezeichnet Wertarbeit im **absoluten** Sinn
- Im Software Engineering wird immer der **industrielle Qualitätsbegriff** (nach ISO 9000) zu Grunde gelegt
 - ➔ **Aufpassen**, wenn Kunden unter Qualität Güte verstehen

6.2 Definieren von Qualität

- Formulieren von **Anforderungen**
 - Anforderungen definieren die geforderte Qualität
 - Was sind **gute** Anforderungen bezogen auf Qualität?

Das System soll über eine Login-Prozedur verfügen, welche den Zugang zu den Kundendaten schützt.

Die Abfrage von Kundendaten muss benutzerfreundlich sein.

Das System soll bis zu einer Million Kundenstammdatensätze speichern.

Das System soll auf einer Linux-Plattform laufen.

- Jede Anforderung ist eine Qualitätsanforderung
- Vier Formen der Repräsentation:
operational • **qualitativ** • **quantitativ** • **deklarativ**

Die Repräsentation determiniert die Prüfbarkeit

- **Operational** Review, Test, formale Verifikation
- **Qualitativ** Keine direkte Prüfung möglich. Entweder subjektive Beurteilung (durch Beteiligte/Interessenvertreter) oder indirekt durch abgeleitete Maße
- **Quantitativ** Messung
- **Deklarativ** Review; teilweise auch Test oder formale Verifikation

- Ziel: alle Anforderungen prüfbar
- Qualitativ repräsentierte Qualitätsanforderungen prüfbar machen durch
 - Operationalisierung oder
 - Quantifizierung

Klassifizierung von Qualität

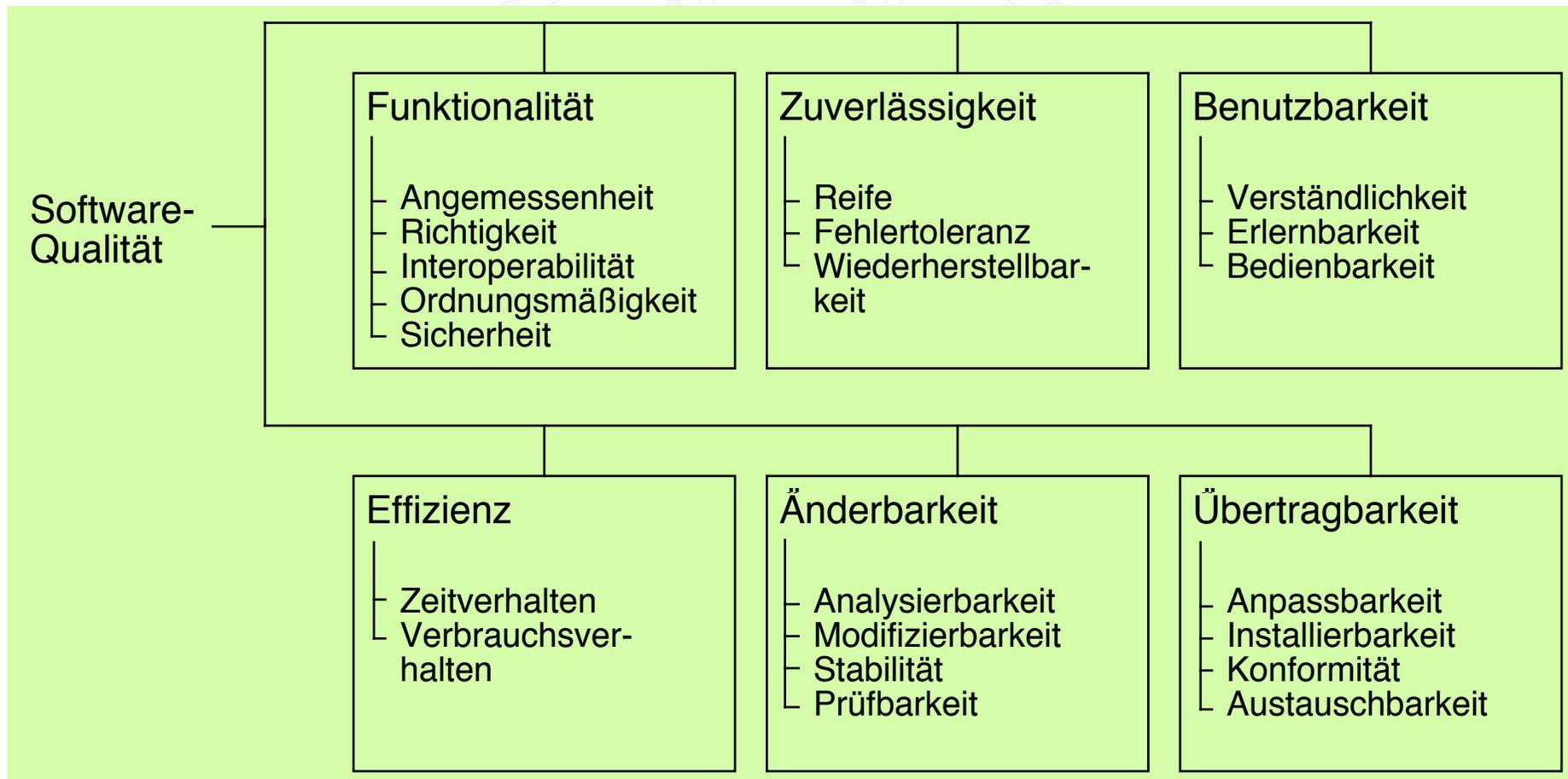
- Grundlegende Unterscheidung:
 - Produktqualität
 - Prozessqualität
 - Projektqualität
- Bei Software ist **Produktqualität** das Primärziel
- Prozessqualität und Projektqualität
 - sind Sekundärziele
 - dienen hauptsächlich der Erreichung von Produktqualität
- Klassifizierung von Produktqualität
 - Qualitätsmodelle
 - Klassifizierung von Anforderungen (welche die Produktqualität definieren)

Qualitätsmodelle

- Strukturieren Produktqualität
- Klassifikation ist typisch ein Baum oder azyklischer gerichteter Graph
- Nutzen?
 - Hilfe bei der Gewinnung und Gliederung von Qualitätsanforderungen
 - Wenn das Modell Maße definiert, Hilfe bei der Bestimmung indirekter Maße für qualitativ repräsentierte Anforderungen

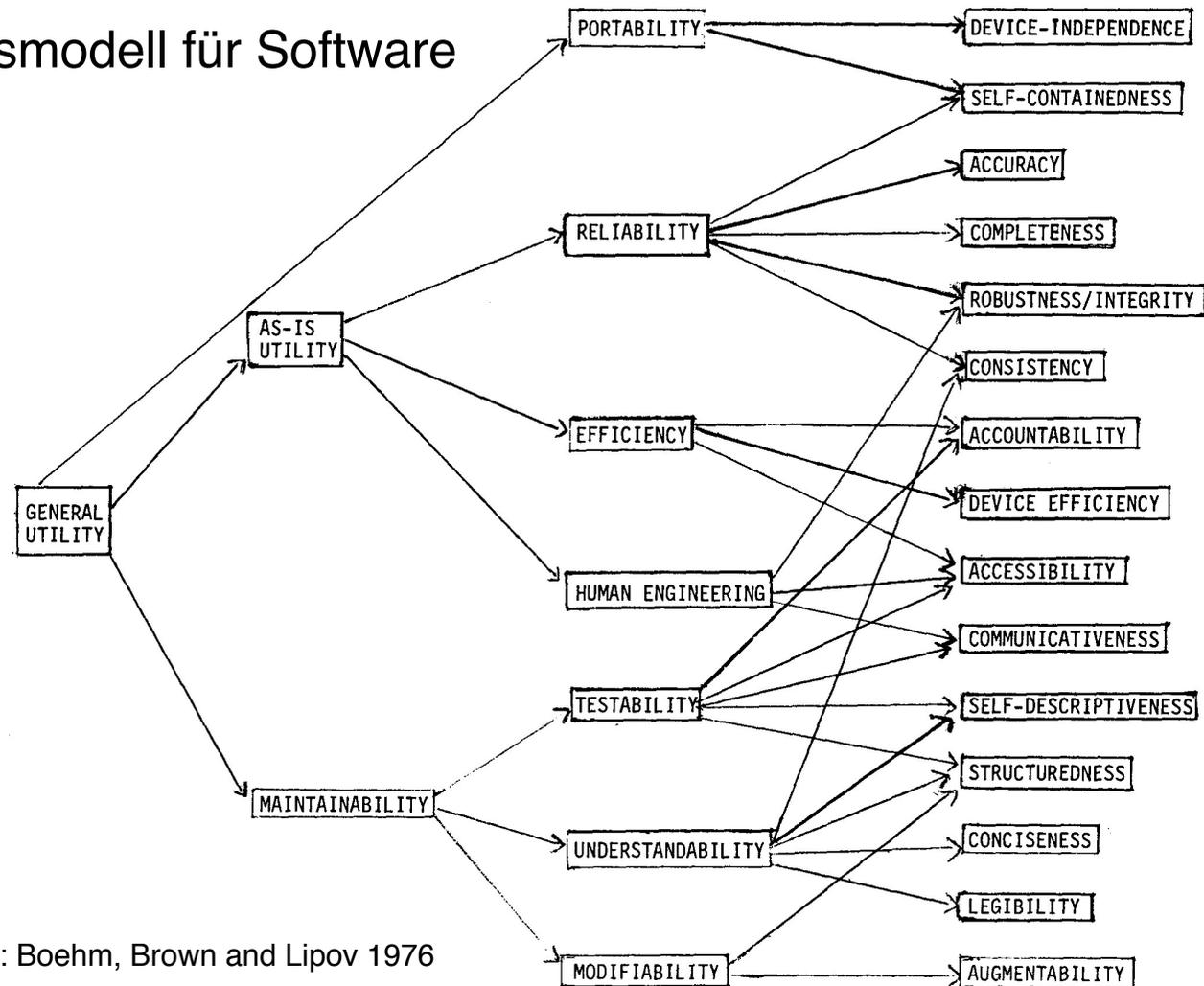
Das Qualitätsmodell der ISO/IEC 9126

- Modell für Software-Produktqualität



Das Qualitätsmodell von Boehm

- Ältestes Qualitätsmodell für Software

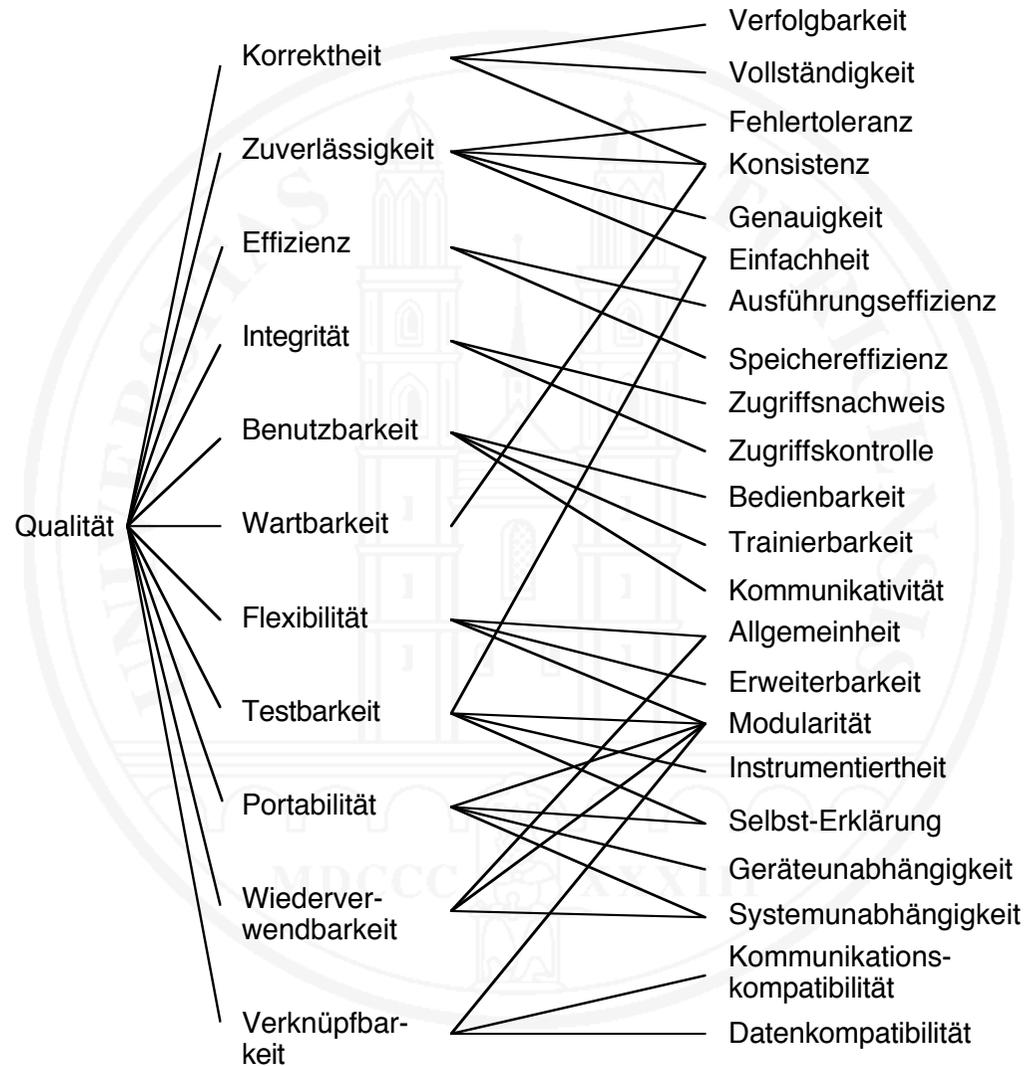


Quelle: Boehm, Brown and Lipov 1976

Das Qualitätsmodell von McCall und Matsumoto

- Dreistufiges Modell
 - eine Menge „allgemeingültiger“ Qualitätsziele (**Faktoren**)
 - ein Satz charakteristischer **Merkmale** zu jedem Faktor
 - messbare **Kenngößen** zu jedem Merkmal
- Bekanntes und oft zitiertes Qualitätsmodell für Software [McCall und Matsumoto 1980]

Das Qualitätsmodell von McCall und Matsumoto – 2



Definition der Qualitätsfaktoren (IEEE 610.12-1990)

- **Korrektheit (Correctness)**
Das Ausmaß, in dem Software ihre Spezifikation erfüllt
- **Zuverlässigkeit (Reliability)**
Die Fähigkeit eines Systems, die verlangte Funktionalität unter gegebenen Randbedingungen für eine gegebene Zeit zu erfüllen
- **Effizienz (Efficiency)**
Das Ausmaß, in dem ein System seine Leistungen mit einem Minimum an Ressourcenverbrauch erbringt
- **Integrität (Integrity)**
Das Ausmaß, in dem ein System unberechtigte Zugriffe auf Programme und Daten bzw. deren unberechtigte Veränderung verhindert

Definition der Qualitätsfaktoren – 2

- **Verwendbarkeit (Usability)**
Die Leichtigkeit, mit der ein Benutzer die Bedienung eines Systems, die Vorbereitung von Daten dafür und die Interpretation seiner Ergebnisse lernen kann
- **Wartbarkeit (Maintainability)**
Die Leichtigkeit, mit der ein System geändert werden kann, um Fehler zu beheben, seine Fähigkeiten zu erhöhen oder es an eine veränderte Umgebung anzupassen
- **Flexibilität (Flexibility)**
Die Leichtigkeit, mit der ein System abgeändert werden kann, um es in Anwendungen oder Umgebungen zu benutzen, für die es nicht entwickelt worden ist

Definition der Qualitätsfaktoren – 3

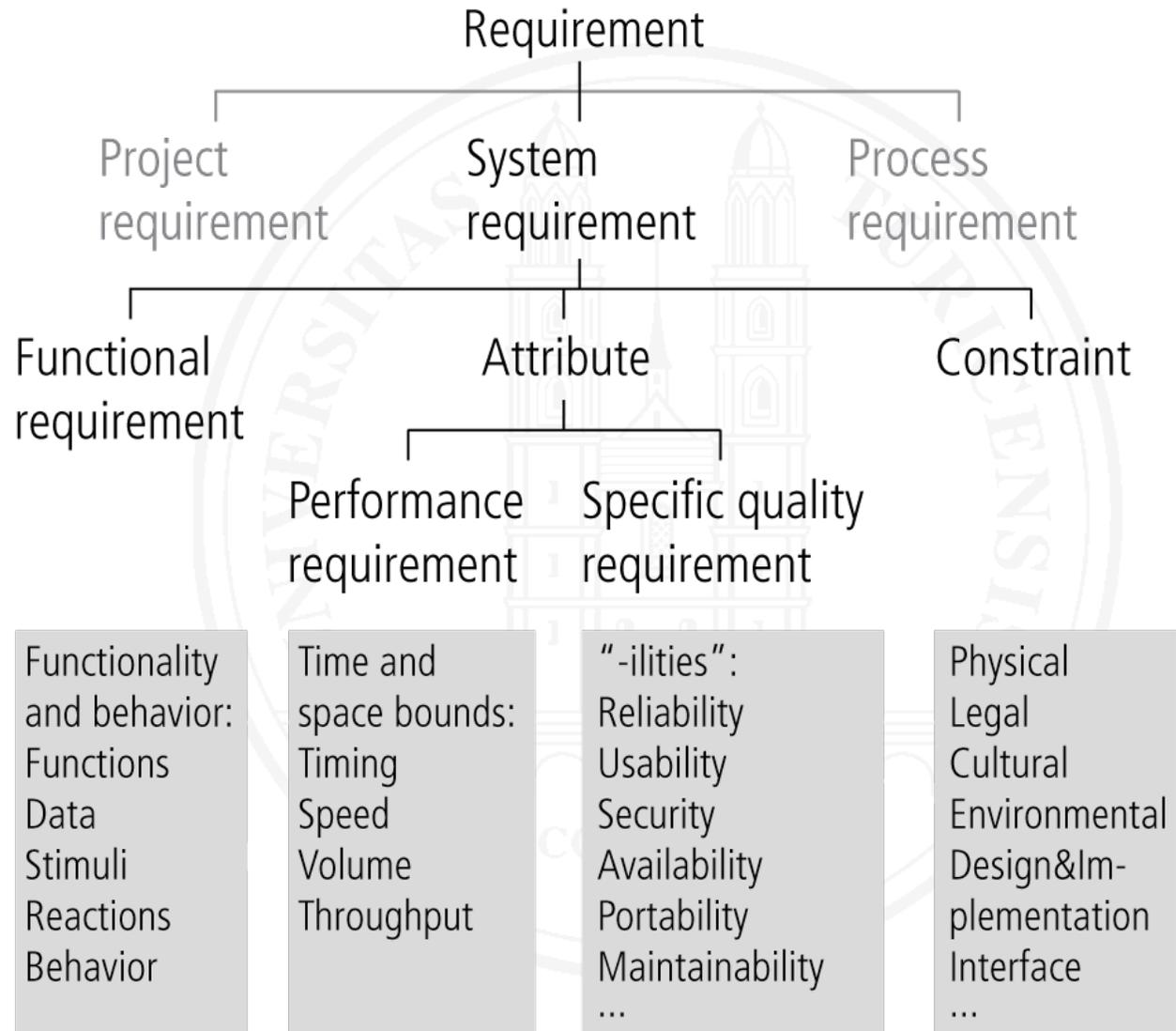
- **Testbarkeit (Testability)**
Das Ausmaß, in dem ein System das Erstellen von Testbedingungen sowie die Durchführung von Tests zur Feststellung, ob die Bedingungen erfüllt sind, erleichtert
- **Portabilität (Portability)**
Die Leichtigkeit, mit der ein System von einer Hard- bzw. Software-Umgebung in eine andere transferiert werden kann
- **Wiederverwendbarkeit (Reusability)**
Das Ausmaß, in dem ein Stück Software in mehr als einem Programm oder Software-System verwendet werden kann
- **Verknüpfbarkeit (Interoperability)**
Die Leichtigkeit, mit der zwei oder mehrere Systeme Informationen austauschen und die ausgetauschten Informationen benutzen können.

Vor- und Nachteile von Qualitätsmodellen

- + Qualitätsfaktoren sind über Merkmale und Kenngrößen **nachvollziehbar** und **messbar** definiert
- + die Vorstellungs- und Begriffswelt über Qualitäten wird **vereinheitlicht**
- Kausale Zusammenhänge zwischen Kenngrößen, Merkmalen und Faktoren sind hypothetisch und **nicht statistisch abgesichert**
- Standardisierte Qualitätsmodelle nehmen keine Rücksicht auf die **individuellen Qualitätsforderungen** von Projekten/Produkten

Beispiel: im Modell nach McCall und Matsumoto kommt **Verfügbarkeit** nicht vor. Dies ist jedoch für viele Systeme ein extrem wichtiger Qualitätsfaktor

Klassifizierung von Anforderungen



[Glinz 2007]

Vorgehen beim Klassifizieren

- Grundsatz: nach der zu Grunde liegenden **Intention**, *nicht* nach der Darstellung der Anforderung
- **Schema:**

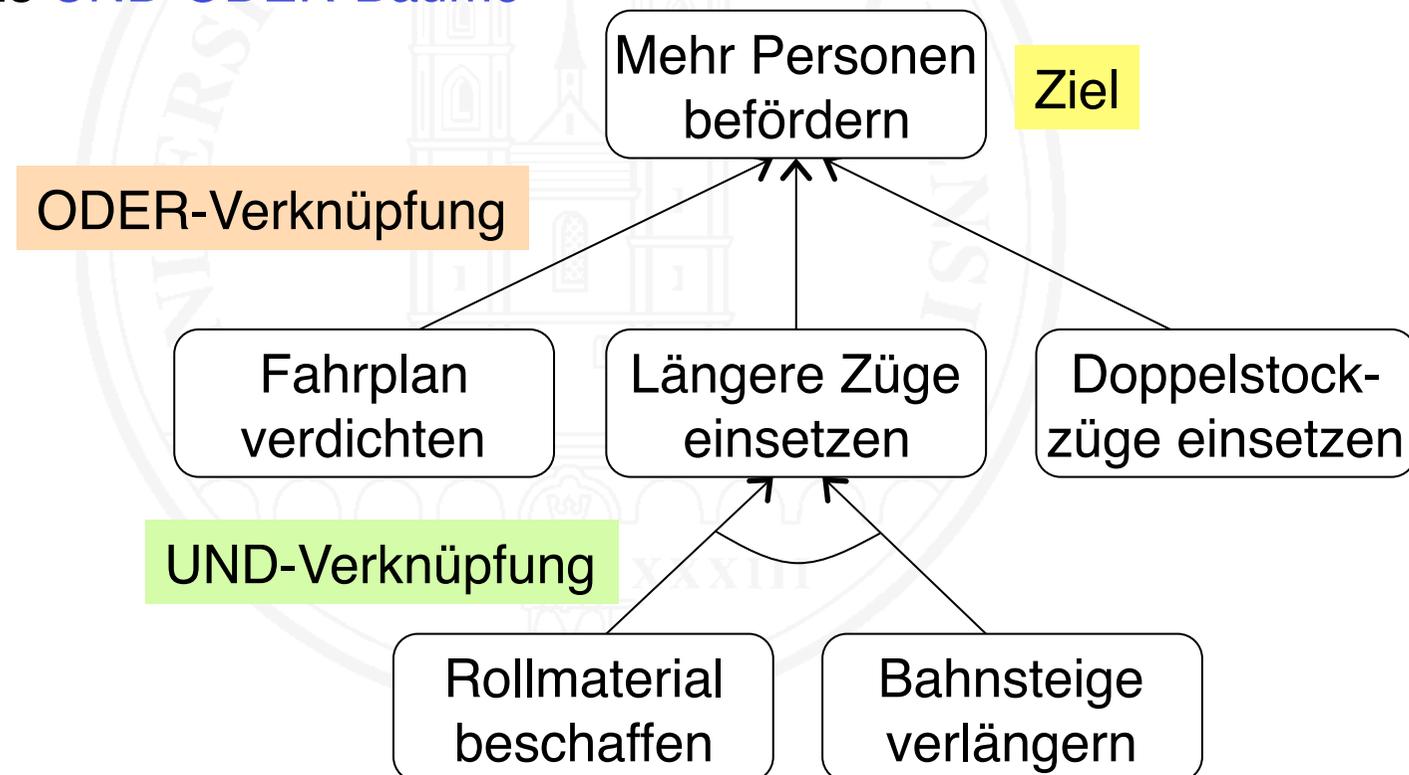
| | |
|--|-------------------------|
| Wird diese Anforderung gestellt, weil... | |
| ... Systemverhalten, Daten, Eingaben oder Reaktionen auf Eingaben zu spezifizieren sind – unabhängig davon, wie dies geschehen soll? | funktionale Anforderung |
| ... Restriktionen bezüglich Verarbeitungs-/ Reaktionszeiten, Datenmengen oder Datenraten zu spezifizieren sind? | Leistungsanforderung |
| ... eine spezielle Qualität, die das System aufweisen soll, zu spezifizieren ist? | besondere Qualität |
| ... irgend eine andere Restriktion zu spezifizieren ist? | Randbedingung |

6.3 Definierte Qualität erreichen

- Letztlich das Problem einer **systematischen Zielerreichung**
- Von Zielen zu Anforderungen
- Von Anforderungen zu Produktmerkmalen
- Zwei grundlegende Techniken
 - Zielbäume
 - Ziel-Mittel Matrizen

Zielbäume

- Systematische **Zerlegung** von Zielen in Teilziele
- Zerlegen, bis den Teilzielen **Systemaufgaben** zugeordnet werden können
- Typisch als **UND-ODER-Bäume**



Ziel-Mittel Matrizen

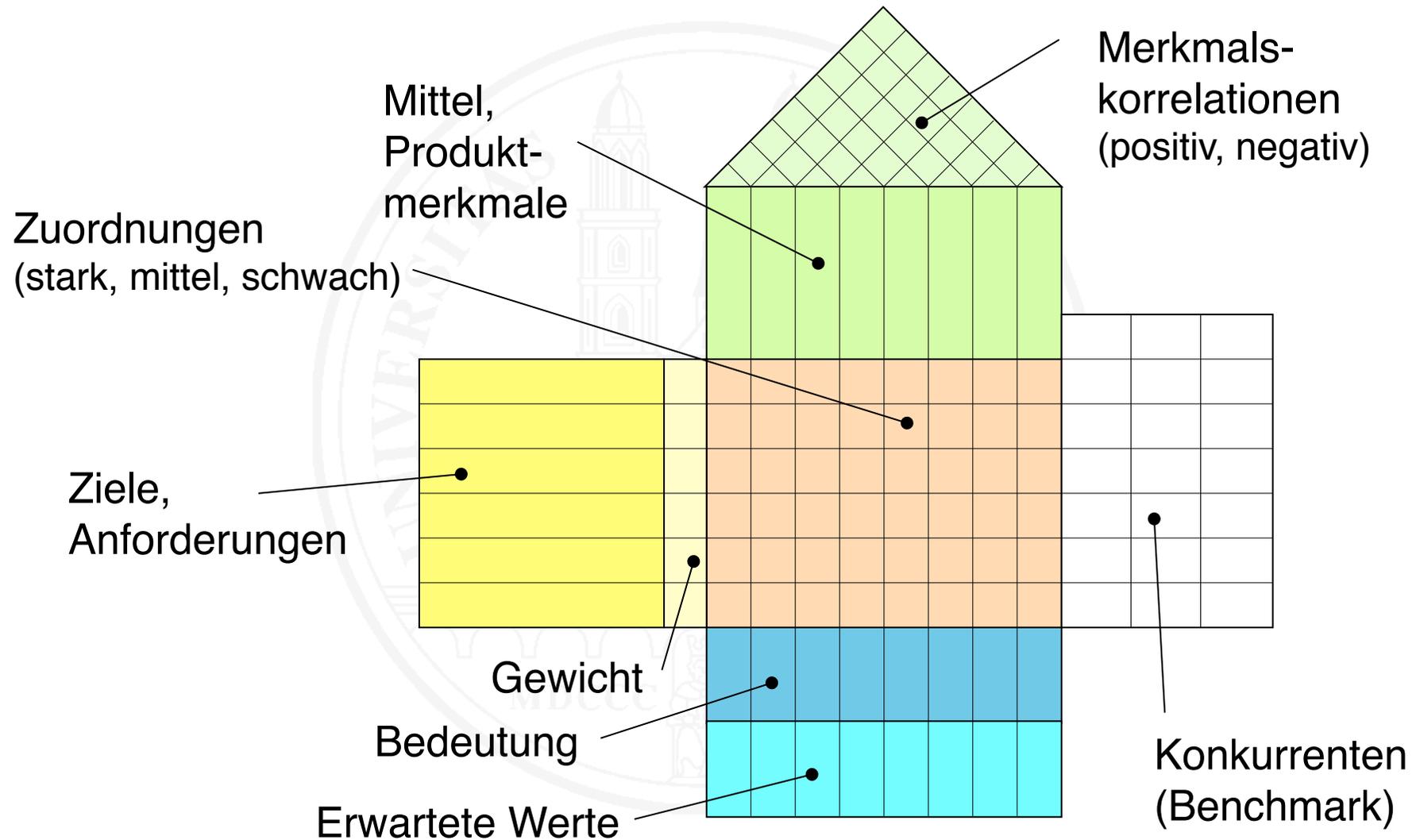
- Matrix, welche Zielen Mittel zur Zielerreichung zuordnet
- Beispielsweise Zuordnung von
 - Anforderungen zu Zielen
 - Produktmerkmalen zu Anforderungen
- Beispiel:

| Ziele | Anforderungen | Geführte Dialoge | Bedienung über Tastatur | Online Hilfe | ... |
|---------------------------------|---------------|------------------|-------------------------|--------------|-----|
| Zufriedene Routinebenutzer | | x | | | |
| Zufriedene Gelegenheitsbenutzer | x | | x | | |
| ... | | | | | |

Quality Function Deployment (QFD)

- Eine verfeinerte und spezialisierte Form von Ziel-Mittel-Matrizen
- Zielsetzung: Kundenanforderungen systematisch in Produktmerkmale umsetzen ⇨ **kundenorientierte Produktentwicklung**
- Entstanden in Japan (hauptsächlich Arbeiten von Akao [Akao 1990])
- Zuerst in der Schiffs- und Fahrzeugindustrie (Mitsubishi Kobe Shipyards 1972, Toyota 1979)
- Zentrales Element ist das so genannte **Qualitätshaus** (house of quality)

QFD: Das Qualitätshaus



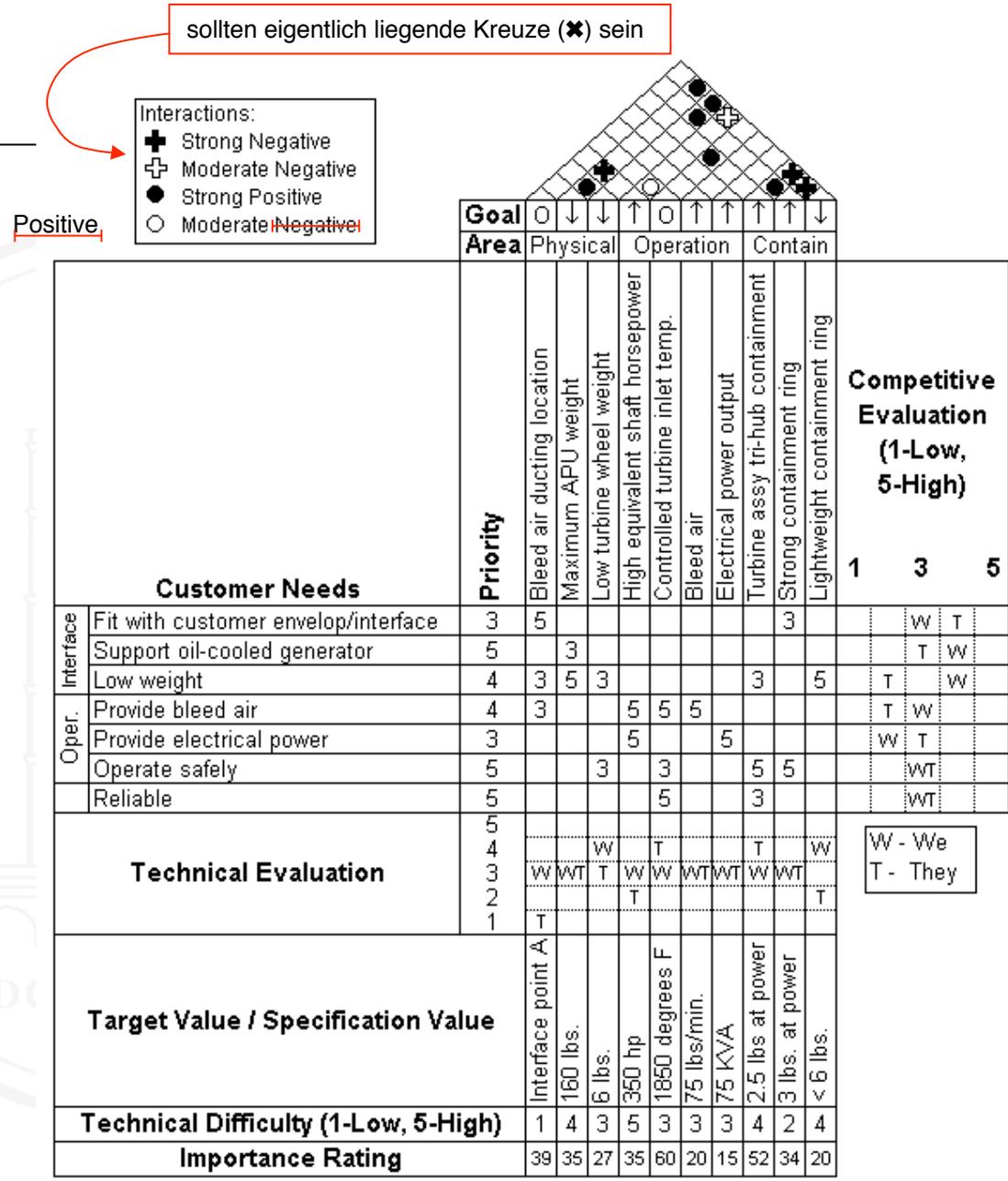
QFD – Beispiel

Qualitätshaus für ein Hilfsaggregat (auxiliary power unit)

[Crow 2002a]

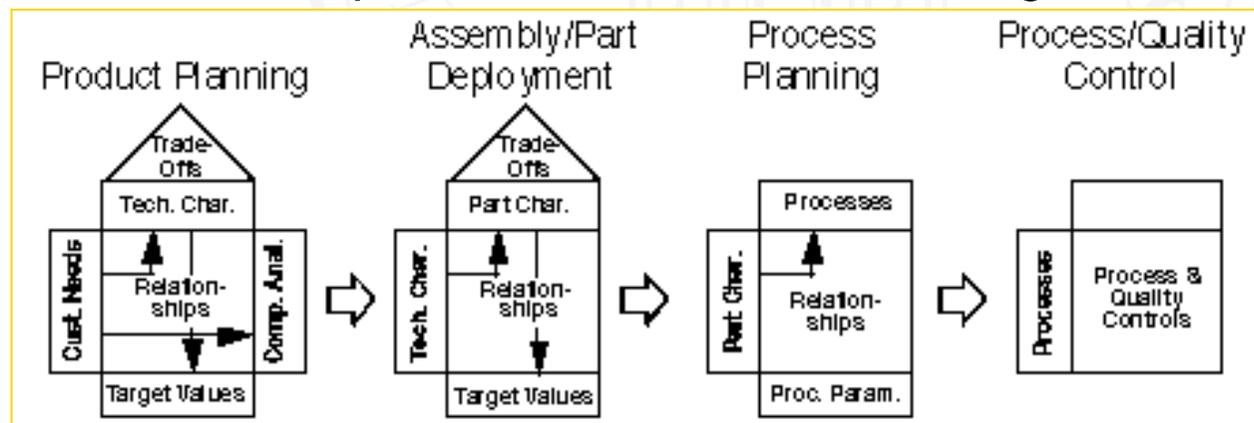
Hinweise:

- Gegenstand ist Produktplanung
- Stockwerke im „Keller“ sind anders angeordnet als auf der vorhergehenden Folie
- Zusätzliche Stockwerke:
 - Optimierungsrichtung für Merkmale („Goal“)
 - Technische Evaluation der Merkmale



QFD – Variationen

- Es gibt viele Varianten betreffend Gewichtung und Bewertung
- Viele QFD-Ansätze verwenden mehrere Qualitätshäuser:
Die Spalten des ersten Hauses werden zu den Zeilen des zweiten, etc., beispielsweise bei technischen Produkten
 - Anforderungen → Produktmerkmale
 - Produktmerkmale → Teile
 - Teile → Produktionsprozesse
 - Produktionsprozesse → Qualitätslenkung



[Crow 2002]

Literatur

Y. Akao (ed.) (1990). *Quality Function Deployment: Integrating Customer Requirements into Product Design*. Cambridge, MA.: Productivity Press.

B. Boehm, J.R. Brown, and M. Lipow (1976). Quantitative Evaluation of Software Quality. *Proceedings of the 2nd International Conference on Software Engineering*, San Francisco. 592-605.

K. Crow (2002). *Customer-Focused Development with QFD*. <http://www.npd-solutions.com/qfd.html>
Besucht 2007-05-14

K. Crow (2002a). *Auxiliary Power Unit Product Planning Matrix*. <http://www.npd-solutions.com/apuppm.gif>
Besucht 2007-05-14

M. Glinz (2007). On Non-Functional Requirements. *Proceedings of the 15th IEEE International Requirements Engineering Conference*. Delhi, India. 21-26.

J.R. Hauser, D. Clausing (1988). The House of Quality. *Harvard Business Review* **66**, 3 (May/Jun). 63-73.

IEEE (1990). *Standard Glossary of Software Engineering Terminology*. IEEE Std 610.12-1990. IEEE Computer Society Press.

ISO 9000:2000. *Qualitätsmanagementsysteme – Grundlagen und Begriffe*. Deutsche Fassung der Europäischen Norm EN ISO 9000 (deutsch/englisch/französisch).

ISO/IEC 9126-1 (2001). *Software Engineering – Product Quality – Part 1: Quality Model*. Internationale Norm ISO/IEC 9126-1.

Literatur – 2

J.A. McCall, M.T. Matsumoto (1980). *Software Quality Measurement Manual*, Vol. II. Rome Air Development Center, RADC-TR-80-109-Vol-2.

