



Software- Qualitätsmanagement

Kernfach Angewandte Informatik

Sommersemester 2004

Prof. Dr. Hans-Gert Gräbe

Organisatorisches

Kernfachvorlesung Angewandte Informatik

Termin: montags, 17:15 - 18:45 Uhr

Ort: Hörsaal 22

Umfang: 2 SWS

Abschluss: Modul-Klausur, Termin noch offen

Informationen: <http://ais.informatik.uni-leipzig.de/>

Sprechzeiten: montags, 13:00–14:00 Uhr, HG 5-53
oder nach Vereinbarung per Email

Vorlesungen

- eBusiness I (Herr Thränert)
 - Kernfach: Praktische und Angewandte Informatik
 - 2 SWS, freitags, 11:15–12:45, HS 22
- Software-Qualitätsmanagement (Prof. Gräbe)
 - Kernfach: Angewandte Informatik
 - 2 SWS, montags, 17:15-18:45, HS 22
- Engineering IT-basierter Dienstleistungen (Prof. Fähnrich)
 - Schwerpunkt Praktische und Angewandte Informatik
 - 2 SWS, mittwochs, 11:15–12:45, HS 1
- Software-Management (Prof. Fähnrich)
 - Schwerpunkt Praktische und Angewandte Informatik
 - 2 SWS, freitags, 09:15–10:45, HS 7

- Verteilte und nebenläufige Programmierung (Prof. Fähnrich, Herr Riechert)
 - Kernfach Angewandte Informatik
 - 1 SWS, donnerstags (A), 13:15 - 14:45 Uhr, HS 4
- Webdesign und Web-Ergonomie (Prof. Fähnrich, Frau Pfretzschner)
 - Kernfach Angewandte Informatik
 - 1 SWS, donnerstags (A), 15:15 - 16:45 Uhr, HS 4
- Seminar Service-Engineering (Maik Thränert)
 - Vertiefung: Praktische/Angewandte/Versicherungsinformatik im Hauptstudiums
 - 1 SWS, donnerstags (B), 15:15-16:45, HS 4
- Softwaretechnik-Praktikum (Prof. Gräbe, Herr Riechert, Herr Kühne)

Praktika / Diplomarbeiten



- Praktikum Webservices (Herr Schumacher)
- Praktika
 - Bewerbungsbogen unter <http://ais.informatik.uni-leipzig.de/>
 - Ansprechpartner: Hr. Thränert
<thraenert@informatik.uni-leipzig.de>
- Diplomarbeiten
 - Bewerbungsbogen unter <http://ais.informatik.uni-leipzig.de/>
 - Ansprechpartner: Hr. Thränert
<thraenert@informatik.uni-leipzig.de>

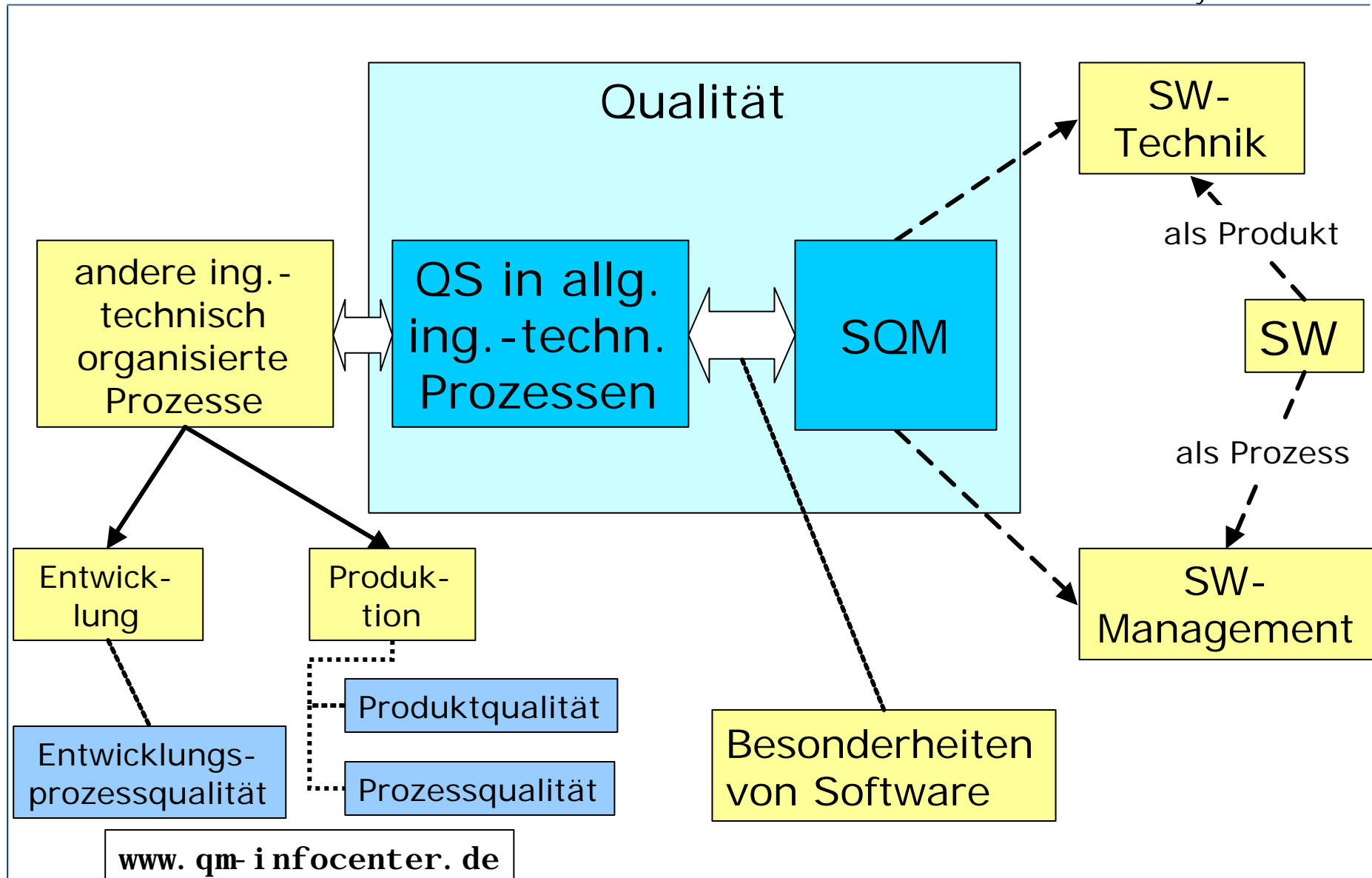
Übersicht der Vorlesung



- Einordnung, Aufgabenstellung, Grundlagen
- Methoden zur Sicherung der Produkt-Qualität
- Qualität und Dokumentation
- Methoden zur Sicherung der Prozess-Qualität

Begleitliteratur: Helmut Balzert, Lehrbuch der Software-Technik, Bd. 2
Quelle der Grafiken und Tabellen: ebenda, wenn nicht anders angegeben

Einordnung des Gegenstands



Fragestellungen

- Was ist Qualität im ingenieur-technischen Verständnis?
- Wie lässt sich Qualität operationalisieren?
- Welche Besonderheiten ergeben sich aus dem speziellen Charakter von Software?
- Zusammenhang zwischen Software-Qualität und dem Software-Lebenszyklus
- Was ist Software-Qualität und wie lässt sie sich managen?

1. Einführung

Was ist Qualität?

Verschiedene Auffassungen der Qualität; jeder Ansatz spiegelt verschiedene betriebliche Sichten auf das Produkt wider:

- der transzendente Ansatz,
- der produktbezogene Ansatz (Entwicklung),
- der benutzerbezogene Ansatz (Marketing/Vertrieb),
- der produktbezogene Ansatz (Fertigung),
- der Kosten/Nutzen-bezogene Ansatz (Finanzen).

„**Qualität** ist die Gesamtheit von Eigenschaften und Merkmalen eines Produkts oder einer Tätigkeit, die sich auf deren Eignung zur Erfüllung gegebener Erfordernisse bezieht.“

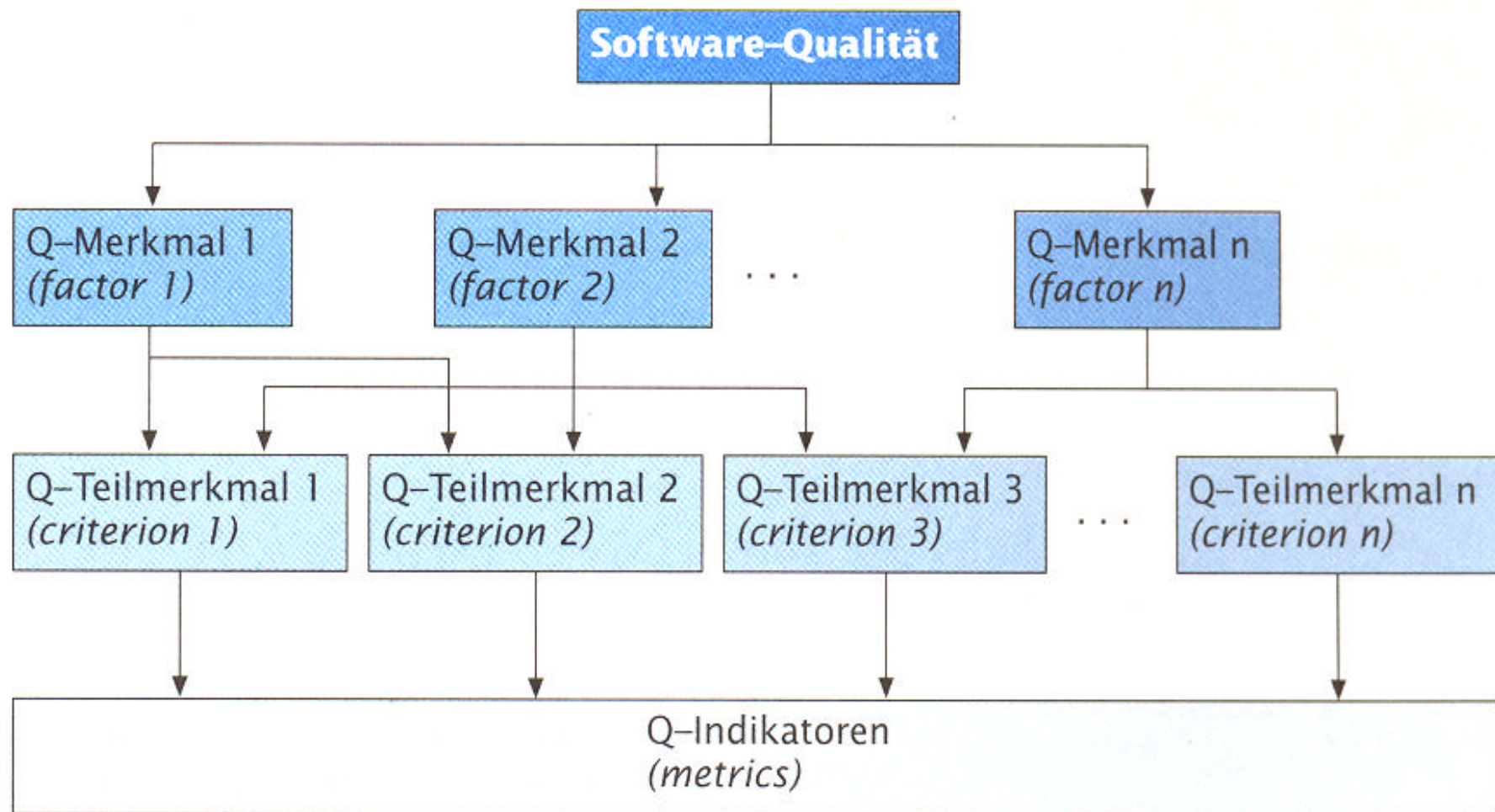
Qualität ist ein relativer Begriff.

1. Einführung

Operationalisierung von Qualität

- FCM-Ansatz (factor – criteria – metrics)
- Qualität wird durch ein **Qualitätsmodell** beschrieben.
- Ein Qualitätsmodell operationalisiert den allgemeinen Qualitätsbegriff durch Ableitung von Unterbegriffen: **Qualitäts-Merkmalen** und **Qualitäts-Kriterien**
- Qualitäts-Kriterien werden durch **Qualitäts-Indikatoren** erfasst.
 - Indikatoren bestehen aus einer quantitativen Skala und einer Methode, mit welcher der Wert bestimmt werden kann, den ein Indikator für ein bestimmtes Produkt oder eine bestimmte Tätigkeit aufweist.

1. Einführung



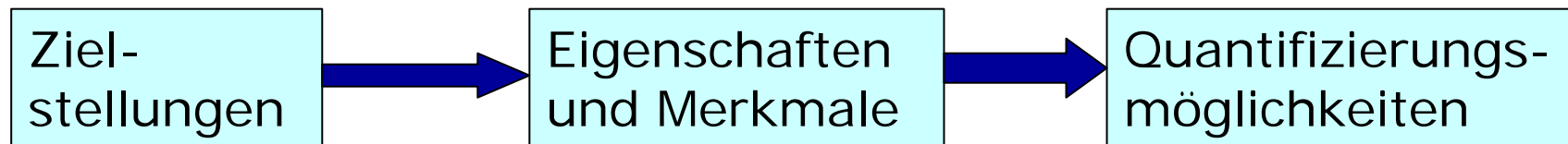
Legende:

→ wird bestimmt durch

1. Einführung

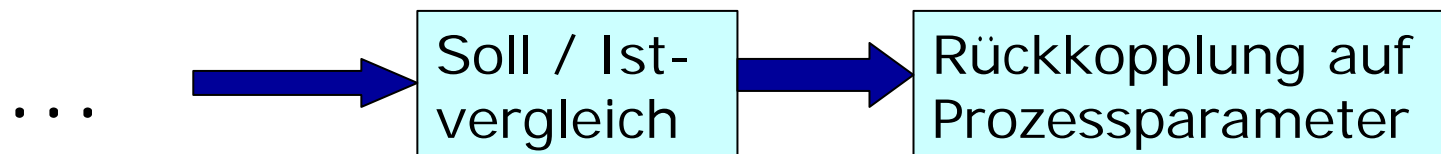
FCM - statischer Ansatz

Ausgerichtet auf Bewertung der Qualität eines vorhandenen Produkts



FCM - dynamischer Ansatz

schließt auch das Qualitätsmanagement ein



Besonderheiten von Software im Vergleich zu anderen ingenieurtechnischen Prozessen

Es gibt keinen Produktionsprozess im engeren Sinne, in dem typgleiche Produkte „ready for use“ produziert werden.

Software ist ein Produkt-**Prototyp**, der evtl. durch Konfiguration und Installation zu einem gebrauchsfertigen Produkt wird.

Das Produkt selbst hat Werkzeugcharakter.

Standardsoftware

hohe Stückzahl,
große Einsatzbreite

Parallelen zum
Werkzeugmaschinenbau

ing.-technischer Zugang

Auftragssoftware

geringe Stückzahl,
spezielle Einsatzbedingungen

Parallelen zum Anlagenbau

handwerklicher Zugang

1. Einführung

Software ist als komplexes technisches Artefakt Materialisierung einer intellektuellen Leistung, welche ohne einen wohldefinierten **Komplex von Kompetenzen** weitgehend nutzlos ist.

Darin unterscheidet sie sich nicht von anderen komplexen Werkzeugen. Die Qualität einer Software besteht damit

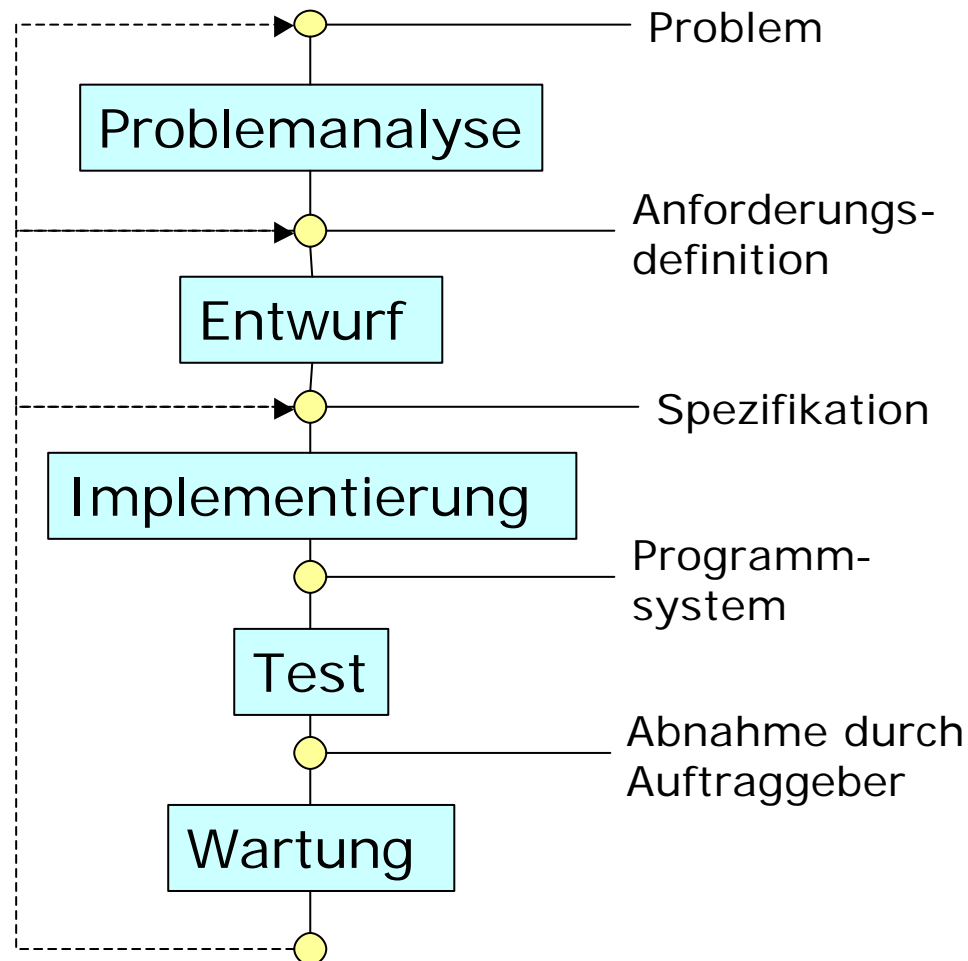
- aus der **Qualität des Softwareprodukts im engeren Sinne** und
- der **Qualität der begleitenden Kompetenzvermittlung** (Dokumentation, Schulung ...)

Softwaremanagement ist ein kooperativer Prozess, in dem unterschiedliche Anforderungen auch an die Kompetenzweitergabe bestehen.

Softwarequalitätsmanagement ähnelt damit am meisten dem Entwicklungsprozess-Qualitätsmanagement, mit der Besonderheit, dass das „Entwicklerteam“ (Entwurf, Installation, Wartung) im Vergleich zu klassischen ingenieur-technischen Prozessen deutlich weniger eng gekoppelt ist.

1. Einführung

Software-Qualität und der Software-Lebenszyklus



Erfasst nur die Herstellung und Wartung eines Software-Produkts

Qualitätssicherung, Dokumentation und Management sind Meta-Aufgaben, welche sich durch alle Phasen ziehen.

2. Software-Qualität

Was ist Software-Qualität und wie lässt sie sich sichern?

- [DIN ISO 9126] definiert 6 Qualitätsmerkmale für die Qualität von Software-Produkten:
 1. **Funktionalität**: Vorhandensein von Funktionen mit festgelegten Eigenschaften, welche die definierten Anforderungen erfüllen.
 - Richtigkeit, Angemessenheit, Interoperabilität, Ordnungsmäßigkeit, Sicherheit
 2. **Zuverlässigkeit**: Fähigkeit der Software, ihr Leistungsniveau unter festgelegten Bedingungen in einem festgelegtem Zeitraum zu entfalten.
 - Reife, Fehlertoleranz, Wiederherstellbarkeit
 3. **Benutzbarkeit**: Aufwand, der zur Einarbeitung und zur Benutzung erforderlich ist, und individuelle Beurteilung der Benutzbarkeit durch eine festgelegte oder vorausgesetzte Gruppe.
 - Verständlichkeit, Erlernbarkeit, Bedienbarkeit

2. Software-Qualität



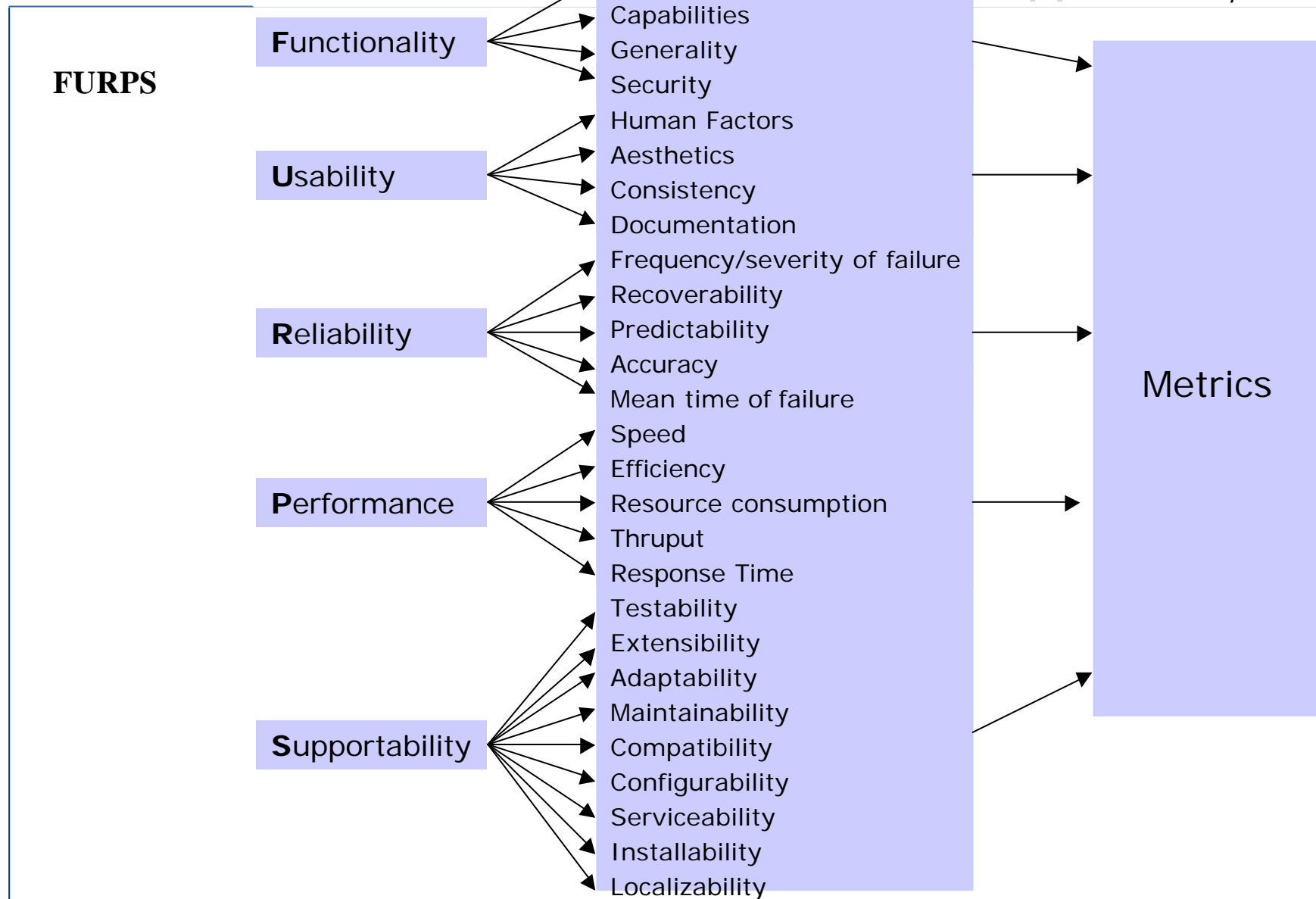
4. **Effizienz**: Verhältnis zwischen dem Leistungsniveau der Software und dem Umfang der eingesetzten Betriebsmittel unter festgelegten Bedingungen.
 - Zeitverhalten, Verbrauchsverhalten
5. **Änderbarkeit**: Aufwand, der zur Durchführung vorgegebener Änderungen notwendig ist. Änderungen können Korrekturen, Verbesserungen oder Anpassungen an Änderungen der Umgebung, der Anforderungen und der funktionalen Spezifikationen einschließen.
 - Analysierbarkeit, Modifizierbarkeit, Stabilität, Prüfbarkeit
6. **Übertragbarkeit**: Eignung der Software, von einer Umgebung in eine andere übertragen zu werden. Umgebung kann organisatorische Umgebung, Hardware- oder Softwareumgebung einschließen.
 - Anpassbarkeit, Installierbarkeit, Konformität, Austauschbarkeit

Beispiel eines FCM-Modells : FURPS

(Functionality-Usability-Reliability-Performance-Supportability)

- Wurde 1985 von der Firma HP entwickelt, um die Qualität ihrer Produkte zu verbessern.
- Kunde soll Software erhalten, die
 - die gewünschte Funktionalität besitzt,
 - vom Benutzer leicht zu bedienen ist,
 - zuverlässig ist,
 - schnell abläuft und
 - aus der Sicht des Kunden eine schnelle Unterstützung bietet.
- Enthält auch Indikatoren.
- Beschränkt aber im Wesentlichen auf Erfassung und Auswertung von Fehlerdaten und Kosten.
- Führte (nach eigenen Angaben) zu einer Reduzierung der Fehler und einer gleichzeitigen Kostensenkung bei der Firma HP.

2. Software-Qualität



Andere Modelle für Software-Produktqualität

- eines der ältesten Modelle [McCall, Richards, Walters 1977]: Zuordnung von Qualitätsmerkmalen zu verschiedenen Szenarien:
 - **Produkt-Einsatz** (Nutzbarkeit, Integrität, Effizienz, Sicherheit, Zuverlässigkeit)
 - **Produkt-Revision** (Wartbarkeit, Testbarkeit, Flexibilität)
 - **Produkt-Weiterentwicklung** (Wiederverwendbarkeit, Portierbarkeit, Interoperabilität)
- Sichten als Ausdruck der Relativität des Qualitätsbegriffs.
- Modell von Boehm [Boehm et al. 1978]: Untergliederung der Gesamtnutzung eines Produkts in die Unterpunkte Anwendernutzung, Portabilität, und Wartbarkeit

2. Software-Qualität

Zuordnung von Qualitätsmerkmalen zu einzelnen Anwendungsklassen

Anwendungsklasse	Qualitätsmerkmale
Menschliches Leben ist betroffen	Zuverlässigkeit, Korrektheit, Testbarkeit
Sehr hohe Entwicklungskosten	Zuverlässigkeit, Flexibilität
Lange Einsatzdauer	Wartbarkeit, Portierbarkeit, Flexibilität
Echtzeit-Anwendungen	Effizienz
Eingebettete Anwendungen	Effizienz, Zuverlässigkeit
verteilte Anwendungen	Interoperabilität



3. Prozessqualität

- Es gibt bisher keine Qualitätsmodelle, welche die **Qualität des Software-Entwicklungsprozesses** beschreiben. Einzelne Qualitätsmerkmale werden aber in der Literatur genannt [Rombach,Basili87]:
 - Produktivität,
 - Qualität des Entwurfsprozesses,
 - Qualität der Inspektionen und Reviews,
 - Qualität des Systemtests,
 - Qualität des Abnahmetests,
 - Qualität des Komponententests,
 - Rechtzeitige Bereitstellung ausreichender Ressourcen.
- [Balzert98] empfindet die Merkmale
 - Planbarkeit,
 - Transparenz und
 - Kontrollierbarkeitdes Prozesses auch als relevant.

4. Qualitätsmonitoring

- Um Produktqualität zu erreichen, muss die Qualität zunächst erfasst werden.
- Dazu müssen die Q.-Ziele festgelegt werden, in Q.-Fragen operationalisiert und diese mit entsprechend quantifizierbaren Parametern untersetzt sein.
- *Goal-Question-Metric-Ansatz (GQM-Ansatz)* [Basili, Rombach87]:
Gibt eine systematische Vorgehensweise zur Erstellung eines solchen Qualitätsmodells an.
 1. Definiere Auswertungsziele
 2. Leite alle Fragenstellungen ab
 3. Leite alle Maße ab
 4. Entwerfe einen Mechanismus
 5. Validiere die Messwerte
 6. Interpretiere die Messergebnisse

4. Qualitätsmonitoring

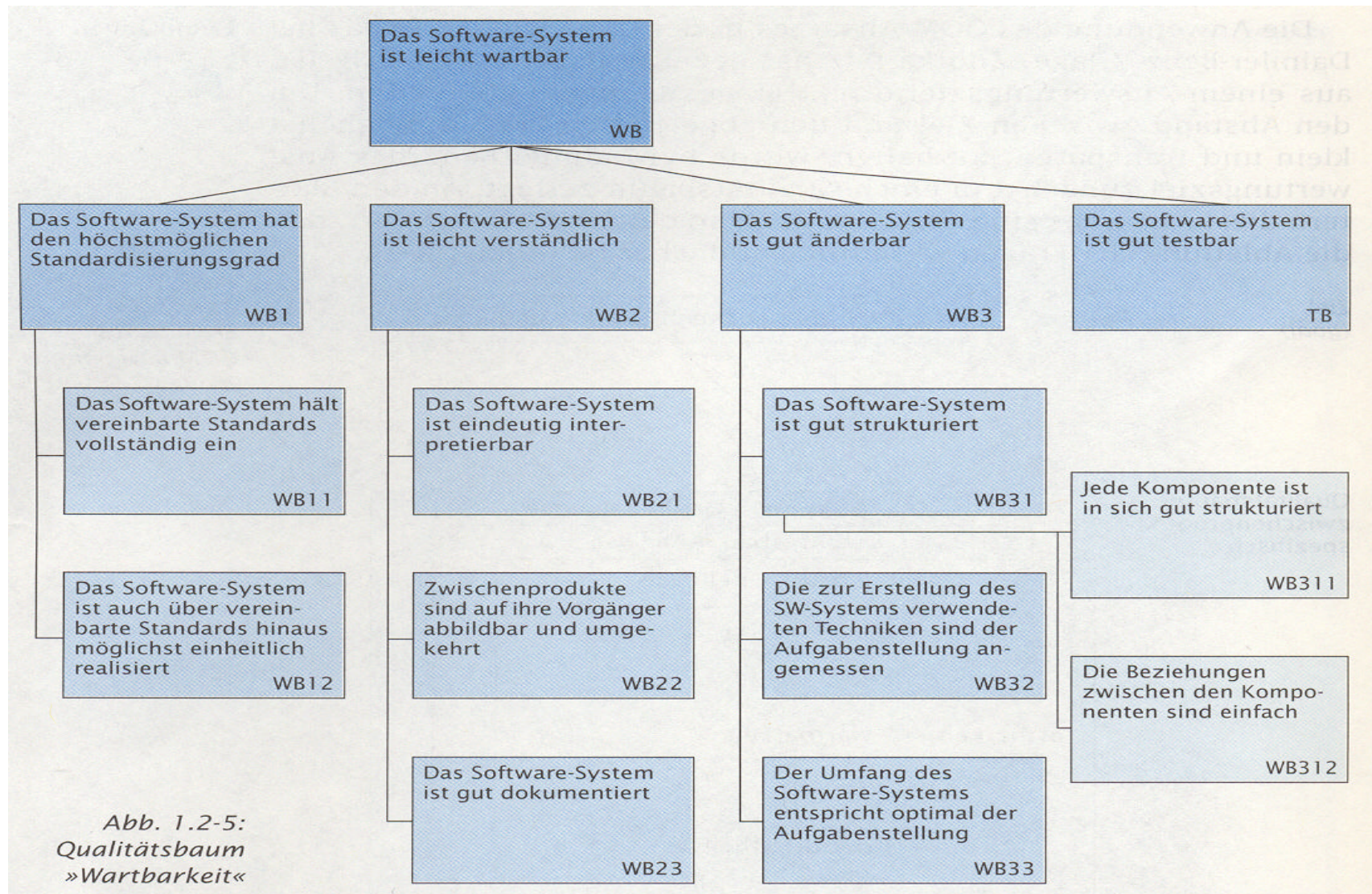


Abb. 1.2-5:
Qualitätsbaum
»Wartbarkeit«

4. Qualitätsmonitoring

Methode: Qualitätsbaum

- Qualitätsziele werden schrittweise verfeinert, bis die Quantifizierung an den Blättern des Baums einfach ist.
 - Modifikation: DAG. Ein Blatt ist für mehrere Qualitätsziele relevant.
- Zu allen Blättern werden Fragen und Kennzahlen entwickelt.
- Für jede Frage wurden die dazugehörigen Informationen für jedes Zwischenprodukt in ein Datenblatt geschrieben.

4. Qualitätsmonitoring

Zwischen- produkt	funktionale Spezifikation	Qualitätsmerkmal	Wartbarkeit
Ziel-Id	WB11FS		
Ziel	Die funktionale Spezifikation hält vereinbarte Standards vollständig ein.		
Frage-Id	FS1		
Frage	Existiert für jeden Prozeß entweder eine Minispezifikation oder eine Verfeinerung in einem DFD?		
Erläuterung/ Bemerkung/ Abgrenzung	B: aus: Richtlinie für die Systementwicklung mit ADW. B: Bei Einordnung in den Qualitätsbaum außerhalb der Standards würde diese Frage an folgende Stelle gehören: System ist gut strukturiert.		
Maß/ Bewertung	0, wenn alle Prozesse entweder durch eine Minispec beschrieben sind oder in einem DFD verfeinert sind. 1 sonst. oder Anzahl der Prozesse ohne Minispec und ohne Verfeinerung / alle Prozesse		
Meßverfahren	Review aller Prozesse der DFD. ADW: <i>Decomposition Diagrammer</i> erzeugt eine hierarchische Übersicht der Prozesse. Dabei sind alle Elementarprozesse mit S und alle anderen anderen Prozesse mit einem P gekennzeichnet. An den Blättern dieses Diagramms dürfen nur Prozesse mit einem S zu finden sein.		
Aspekte des Maßes	Bedeutung des Maßes: 0 Durchgängigkeit: 0,5 Erhebungsaufwand (manuell/Werkzeug): 0,5/0		
Regeln/ Handlungs- anweisungen/ Bemerkungen	H: Beschreibe alle beanstandeten Prozesse mittels Minispec oder verfeinere sie in einem DFD. B: ADW: Elementarprozesse können durch Anlegen einer Minispezifikation definiert werden oder durch explizite Angabe im Datenflußdiagramm (ohne Erzeugen einer Minispec). Im ersten Fall existiert eine Minispec zwangsläufig, im zweiten Fall kann es jedoch vorkommen, daß keine Minispec existiert.		

4. Qualitätsmonitoring

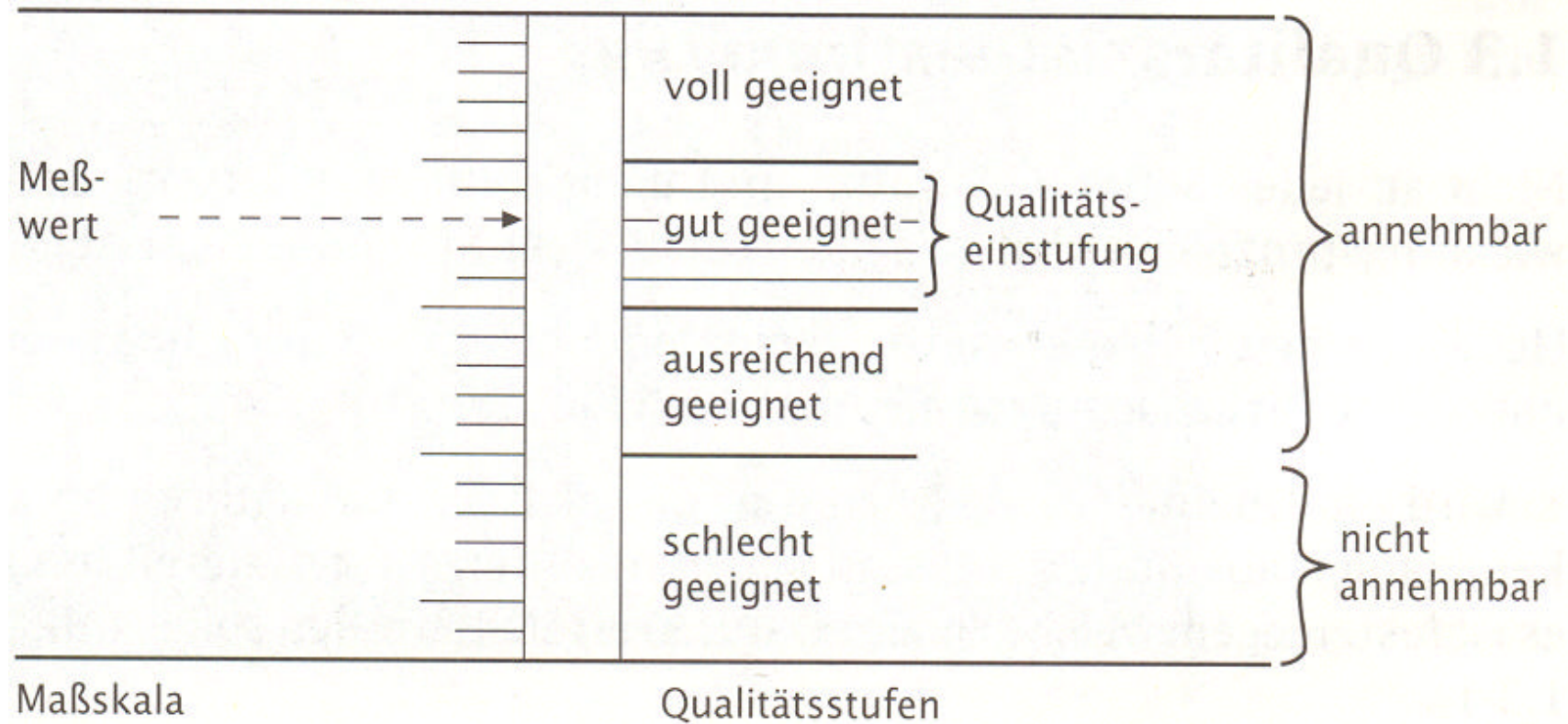
Zwischen- produkt	funktionale Spezifikation	Qualitätsmerkmal	Wartbarkeit
Ziel-Id	WB 311 FS		
Ziel	Die funktionale Spezifikation ist in sich gut strukturiert.		
Frage-Id	FS15		
Frage	Sind alle DFD in sich gut strukturiert?		
Erläuterung/ Bemerkung/ Abgrenzung	B: Eine Beantwortung dieser Frage setzt eine detaillierte Betrachtung der Struktur der DFD voraus.		
Maß/ Bewertung	für jedes DFD: # der Prozesse (Funktionen) # der Datenflüsse durchschnittliches und maximales Gewicht der Datenflüsse (Gewicht eines Datenflusses ist die Anzahl seiner Felder) # der Datenspeicher # der Datenflüsse/# der Prozesse (= durchschnittliche # der Datenflüsse aller Prozesse) max. # der Datenflüsse eines Prozesses Für alle DFD: $\sum \# \text{ der Prozesse über alle DFD}$ max. # der Prozesse für alle DFD durchschnittliche # der Prozesse durchschnittliches und maximales Gewicht der Datenflüsse (Gewicht eines Datenflusses ist bestimmt durch die # der Felder)		
Meßverfahren	Zählen aller Prozesse, Datenflüsse und Datenspeicher eines jeden DFD. Jeder Datenfluß zwischen zwei Objekten wird gezählt.		
Aspekte des Maßes	Bedeutung des Maßes: 0 Durchgängigkeit: 0,5 Erhebungsaufwand (manuell/Werkzeug): 1/0		

Legende: # = Anzahl

5. Qualitätslenkung

- Wird ein Qualitätsmodell benutzt, so sind **Qualitätsstufen** zu definieren, und es ist festzulegen, welche Stufen erreicht werden sollen.
- Qualitätsstufe ist ein Wertebereich auf einer Skala, mit deren Hilfe Software festgelegten Qualitätsforderungen zugeordnet werden kann.
- Auf dieser Basis wird eine **Qualitätszielbestimmung** vorgenommen.
- Der Geltungsbereich von Qualitätszielen kann sich erstrecken auf:
 - eine softwareproduzierende Einheit,
 - auf Teilprodukte eines Software-Produkts,
 - auf den gesamten Software-Erstellungsprozess,
 - auf Teile des Software-Erstellungsprozesses.

5. Qualitätslenkung



5. Qualitätslenkung

- In der Regel ist eine Qualitätszielbestimmung pro Produkt erforderlich und als **Qualitätsanforderung** zu fixieren.
 - Legen fest, welche Qualitätsziele als relevant betrachtet werden.
 - Manchmal reicht eine Qualitätszielbestimmung für eine ganze Klasse ähnlicher Software-Produkte aus.
- Die Qualitätsanforderungen sind vor dem Entwicklungsbeginn zu fixieren und z.B. im Pflichtenheft zu dokumentieren.
 - Zwingend, da die zu erreichenden Qualitätsparameter Auswirkung auf Termin und Kosten haben.
- durch entwicklungsbegleitende **Qualitätsprüfungen** sind die Anforderungen sicherzustellen
- Neue, die Qualität betreffende Ergebnisse erfordern eine Wiederholung der Qualitätszielbestimmung.
- Sind alle Anforderungen erfüllt, kann bei der Abnahme ein entsprechendes **Produktzertifikat** vergeben werden.

5. Qualitätslenkung

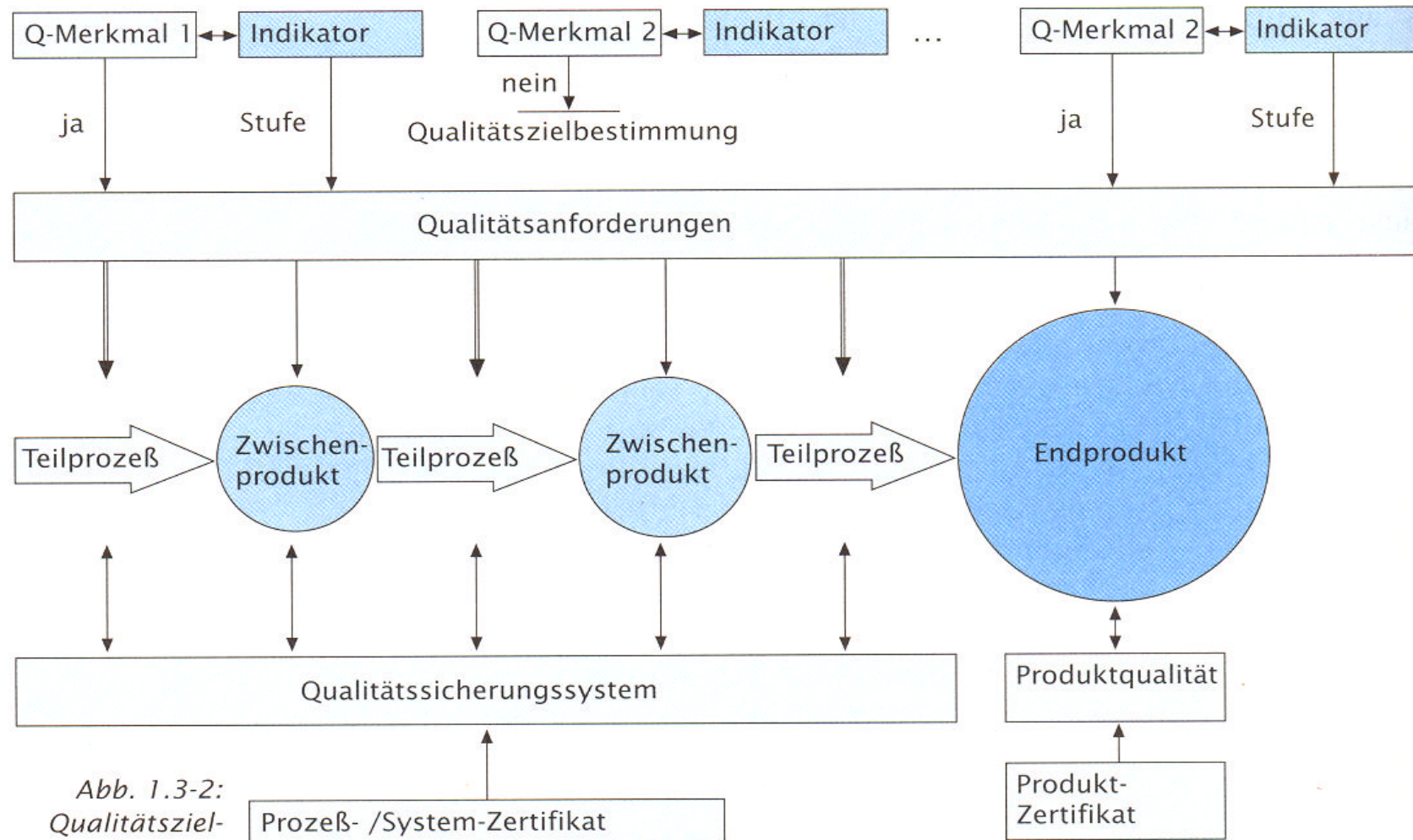


Abb. 1.3-2:
Qualitätsziel-
bestimmung
und ihre
Auswirkungen

Legende:
 Prozeßorientierte Anforderungen
 Produktorientierte Anforderungen

6. Zusammenfassung

- FCM-Modell =

typisches **Strukturmodell**, über welches der Qualitätsbegriff operationalisiert werden kann

- Q.-**Merkmale** werden an quantifizierbare Q.-**Kriterien** gebunden und für diese Q.-**Indikatoren** identifiziert.
- Ergebnis ist ein FCM-Baum oder FCM-Netz, welches den Zusammenhang zwischen (qualitativen) Merkmalen und (quantifizierbaren) Indikatoren herstellt.

- GQM-Ansatz =

typisches **Vorgehensmodell** zur Planung des QS-Prozesses.

- Q.-**Ziele** und deren Wichtung werden projektbezogen bestimmt und im Rahmen der Q.-**Zielbestimmung** die Q.-**Anforderungen** sowie die zu erreichenden Q.-**Stufen** festgelegt.

Literaturangaben

[Balzert98]

Balzert H., Lehrbuch der Software-Technik, 1998

[Boehm et al. 78]

Boehm B.W., Brown J.R., Kaspar H., Lipow M., MacLeod G.J., Meritt M.J.,
Characteristics of Software Quality, 1978

[DGQ 86]

DGQ/NTG: Software-Qualitätssicherung, 1986

[DIN ISO 9126]

Informationstechnik – Beurteilen von Softwareprodukten, 1991

[Hesse et al.84]

Hesse W., Keutgen H., Luft A.L., Rombach H.D., Ein Begriffssystem
für die Software-Technik, 1984

[McCall,Richards,Walters77]

MacCall J.A., Richards P.K., Walters G.F., Factors in Software Quality,
1977

[Rombach,Basili 87]

Rombach H.D., Basili V.R., Quantitative Software-Qualitätssicherung, 1987