

# Software-Qualitätsmanagement

**Kernfach Angewandte Informatik und  
Vorlesung im Modul 10-202-2319  
Software-Management**

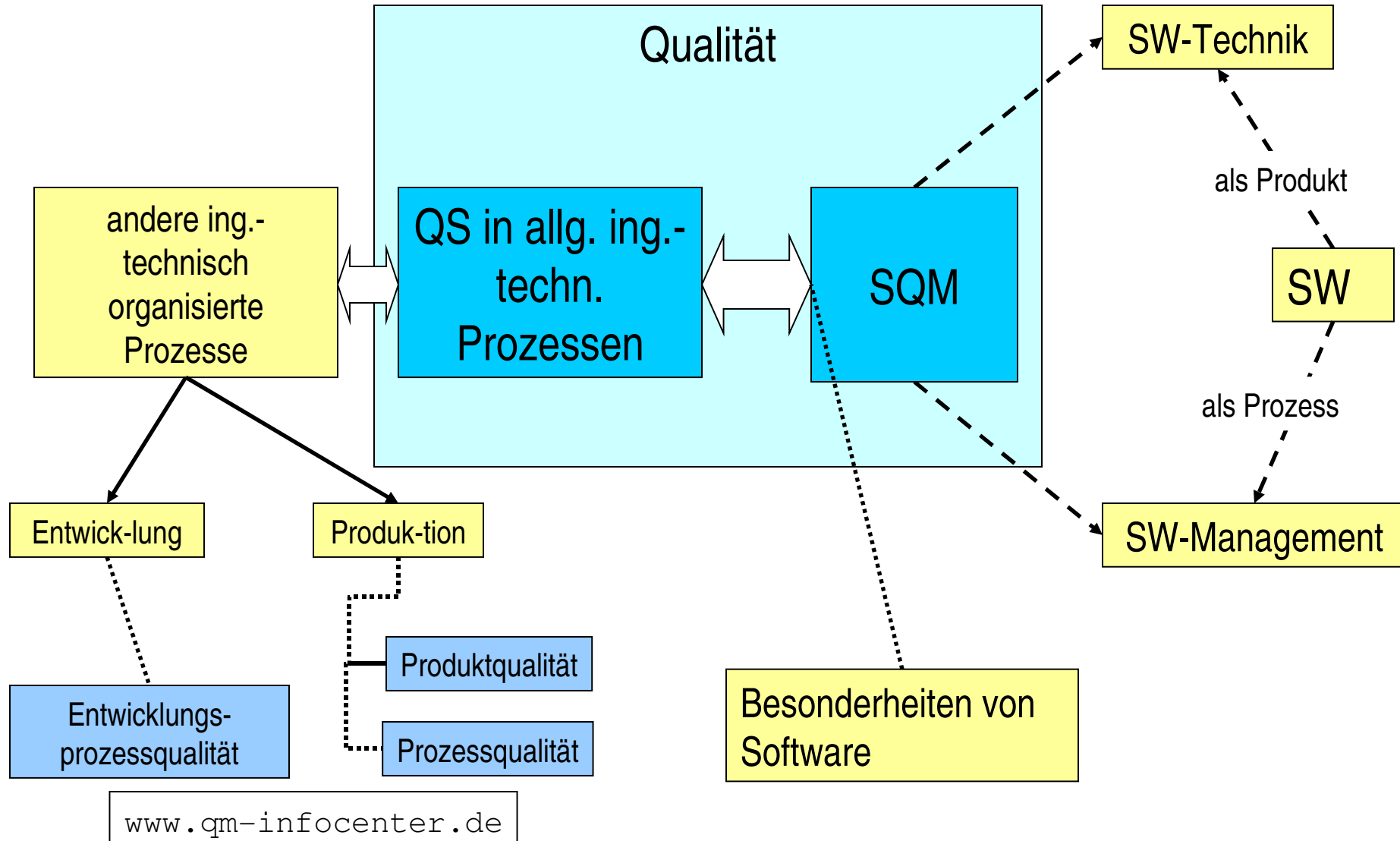
Sommersemester 2008

apl. Prof. Dr. Hans-Gert Gräbe

<http://bis.informatik.uni-leipzig.de/HansGertGraebe>

- 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



## 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?

## Was ist Qualität?

Verschiedene Auffassungen der Qualität; jeder Ansatz spiegelt verschiedene betriebliche Sichten auf das Produkt wider [Garvin 88]

- 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.“ (ISO 8402)

Qualität ist ein relativer Begriff.

Qualität im Sinne der IEEE-Norm 729-1983 hebt die Erwartungen der Kunden hervor:

- Die Gesamtheit der Funktionalität und Charakteristika einer Software, welche sie zur Erfüllung definierter Anforderungen geeignet macht, etwa Spezifikations-Konformität.
- Der Grad, mit welchem in der Software verschiedene Eigenschaften miteinander kombinierbar sind.
- Der Grad, in welchem der Nutzer spürt, dass die Software seinen komplexen Erwartungen entspricht.

Die Beurteilung dieser Zielgrößen ist undurchsichtig und schwer auszuwerten. Kunden nehmen nur einen Teil der in Anspruch genommenen Qualität wahr. (Überschuss an Qualität)

Qualität in diesem Sinne ist **Produktqualität**.

Wenn darüber hinaus Qualität konstruktiv realisiert werden soll, muss ein Bewertungsansatz für den Entwicklungs- und Pflegeprozess aufgestellt werden (**Prozessqualität**).

## Qualität ist ein relativer Begriff

Qualitätsbewertung ist ein Vergleich zwischen

- Qualitätsvorgaben (Soll-Werten), welche aus den gegebenen Erfordernissen abgeleitet sind
- und den tatsächlich erreichten Ausprägungen der Merkmale (Ist-Werte)

Drei zentrale Fragen [Boehm 78]:

- Problem der Definition: Gibt es überhaupt solche quantitativen, messbaren Parameter?
- Problem der Prüfung: Wie gut kann man die Merkmale erfassen, welche die Qualität bestimmen?
- Problem der Lenkung: Wie kann man Informationen über die Qualität zur Steuerung im Softwarelebenszyklus einsetzen?

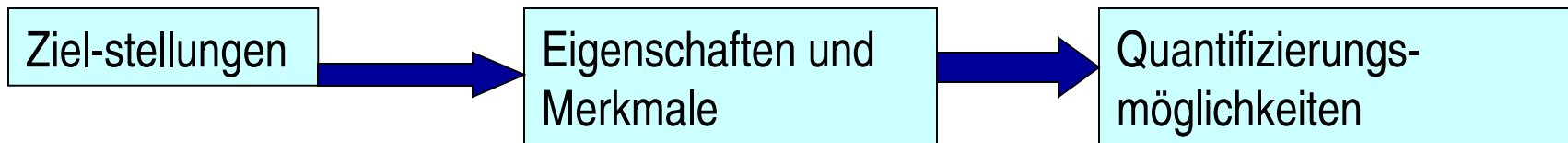
## Operationalisierung von Qualität

- 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.



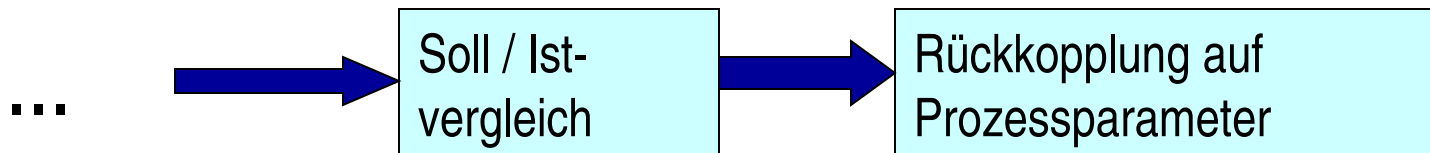
### FCM – definatorischer Ansatz

Ausgerichtet auf Bewertung der Qualität eines vorhandenen Produkts



### FCM – prozessualer Ansatz

schließt auch das Qualitätsmanagement ein.



Kann weniger formalisiert sein.

## 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

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.

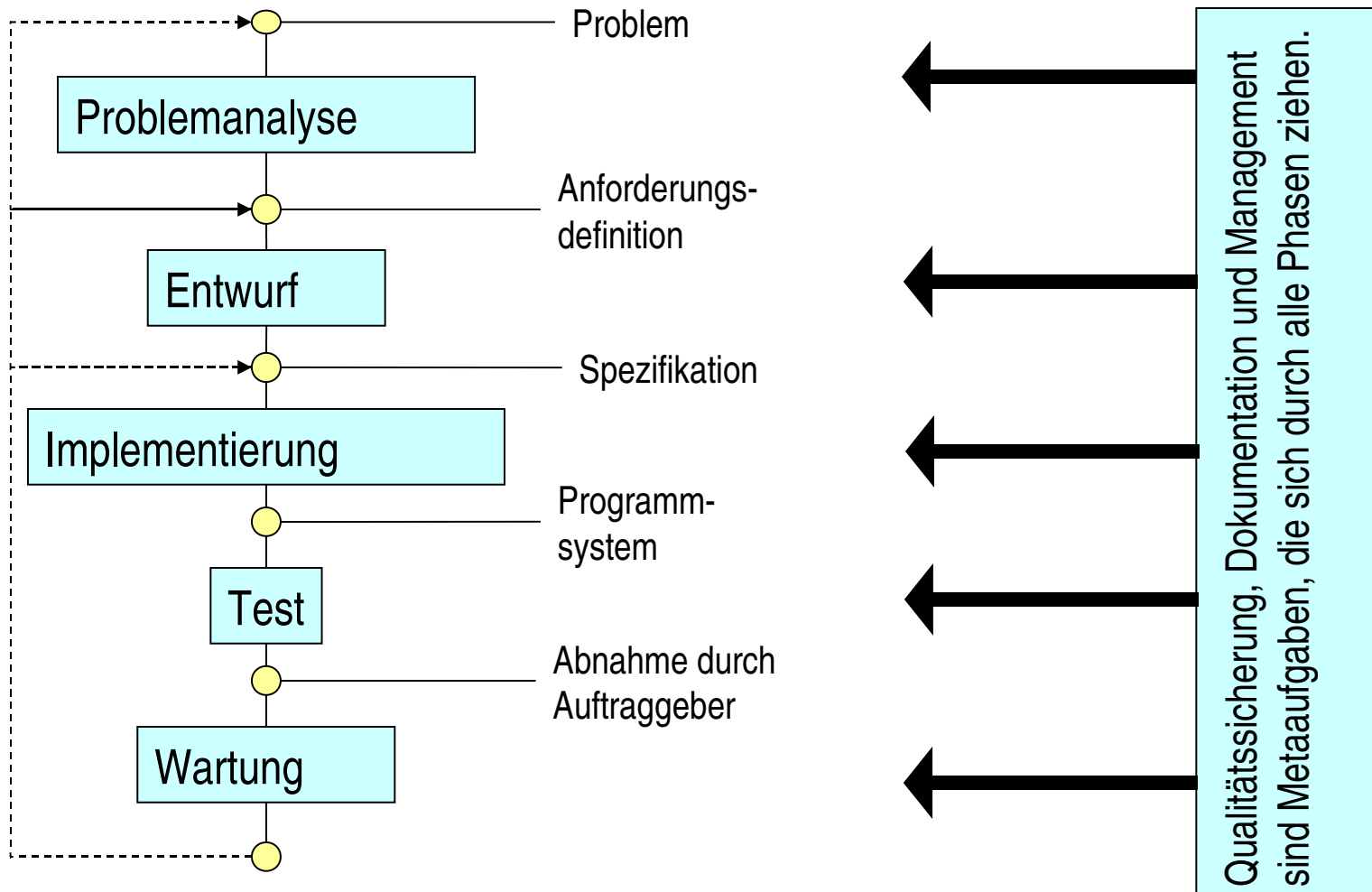
Softwarequalitätsmanagement ähnelt damit am meisten dem Entwicklungsprozess-Qualitätsmanagement, mit der Besonderheit, dass das „Entwicklerteam“ (Entwurf, Installation, Wartung) im Vergleich zu klassischen ingenieur-technischen Prozessen deutlich weniger eng gekoppelt ist.

## Software als Produkt

Im Sinne der IEEE-Norm 729-1983 besteht deshalb ein Software-Produkt aus den Teilen Quellcode, Objektcode und Dokumentation.

Voraussetzung für qualitätsbewusstes Umgehen mit Software ist u.a. das Vorliegen einer aktuellen und ausreichenden Dokumentation, in welcher neben den Programmen auch die Daten beschrieben werden.

## Software-Qualität und der Software-Lebenszyklus



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

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

- **Effizienz**: Verhältnis zwischen dem Leistungsniveau der Software und dem Umfang der eingesetzten Betriebsmittel unter festgelegten Bedingungen.
  - Zeitverhalten, Verbrauchsverhalten
- **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
- **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

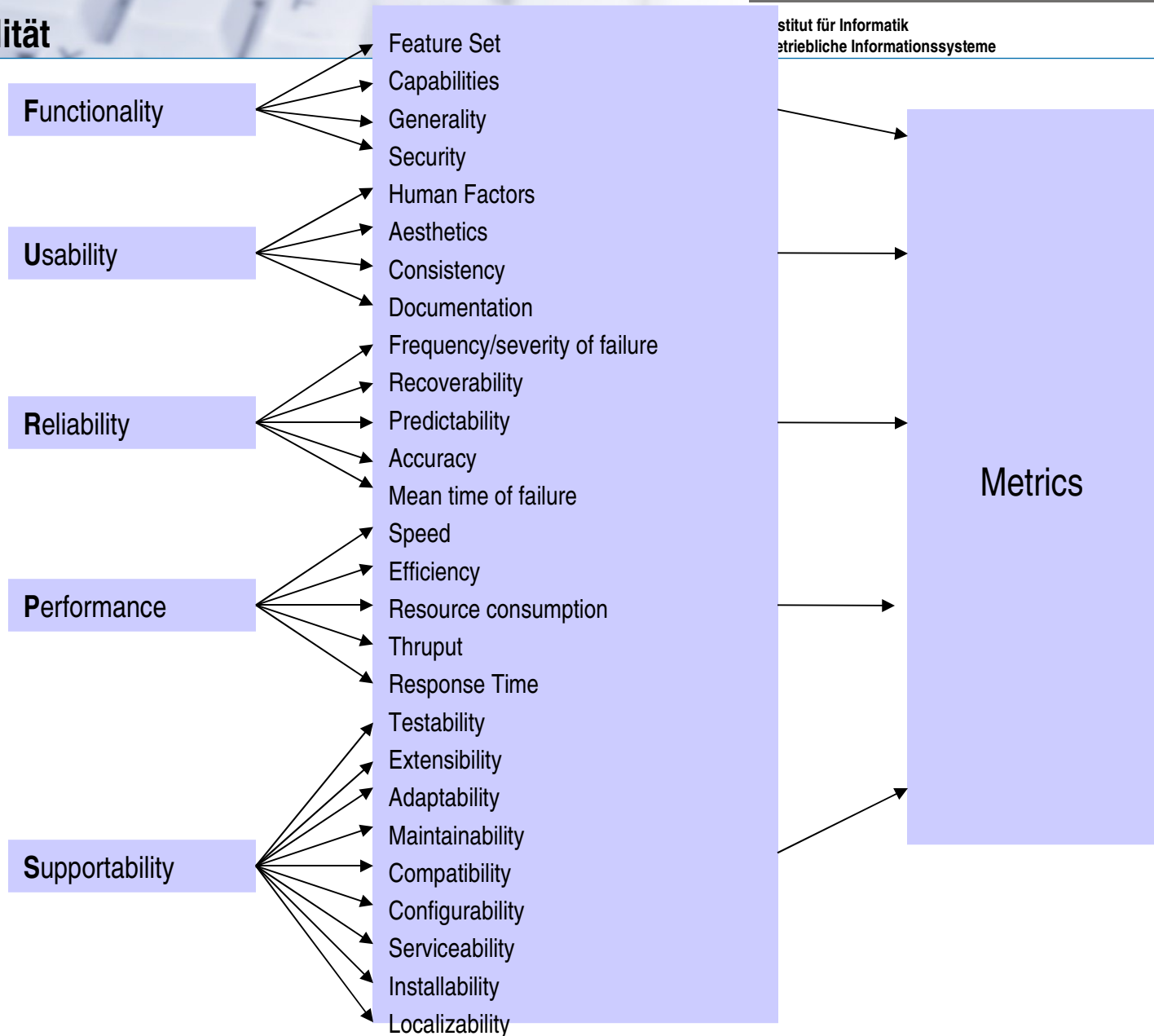
(Functionality-Usability-Reliability-Performance-Supportability)

- Wurde 1985 von der Firma HP entwickelt, um die Qualität ihrer Produkte zu verbessern.
- Kunde soll Software erhalten, welche
  - die gewünschte Funktionalität besitzt,
  - vom Benutzer leicht zu bedienen ist,
  - zuverlässig ist,
  - gute Leistungsparameter hat und
  - gut wartbar ist.
- Enthält auch Indikatoren.
- Beschränkt aber im Wesentlichen auf Erfassung und Auswertung von Fehlerdaten und Kosten (statisches FCM)
- Führt (nach eigenen Angaben) zu einer Reduzierung der Fehler und einer gleichzeitigen Kostensenkung bei der Firma HP.



## 2. Software-Qualität

### FURPS



stitut für Informatik  
etriebliche Informationssysteme

## 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, Portier-barkeit, 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

## Prozessqualität

- Von der Beurteilung der Produktqualität sind Fragen der Beurteilung der Qualität des Software-Entwicklungsprozesses zu unterscheiden.
- Schwerpunkt der Prozessqualität liegt nicht auf dem **Ergebnis**, sondern auf dem Prozess der **Herstellung**.
- Verwendet ähnliche Ansätze wie FCM:
  - Welche Ziele (Merkmale) sind zu erreichen?
  - Durch welche Kriterien werden die Ziele charakterisiert?
  - Mit welchen Indikatoren kann man die Kriterien erfassen?
- Es existieren verschiedene Normenwerke und Ansätze
  - ISO 9000, TQM, CMM und CMM-I, ISO 15504

### Qualitätszielbestimmung

Um Produktqualität zu erreichen, müssen zunächst Qualitätsmaßstäbe in einem FCM-Modell festgelegt und die Indikatoren erfasst werden.

**Qualitätsmodell:** Systematik, nach welcher die Q.-Ziele festgelegt, in Kriterien operationalisiert und mit entsprechenden Indikatoren untersetzt sein.

Beispiel: *Goal-Question-Metric-Ansatz (GQM)* [Basili, Rombach87]:

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

### Qualitätsziele identifizieren: Qualitätsbaum

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

### Beispiel Qualitätsbaum

- Das System ist leicht wartbar
  - Standardisierungsgrad
    - vorhandene Standards werden vollständig eingehalten
    - Einheitlichkeit auch über vorhandene Standards hinaus
  - Verständlichkeit
    - eindeutig interpretierbar
    - gut dokumentiert
  - Änderbarkeit
    - gut strukturiert
      - Komponenten in sich gut strukturiert
      - Beziehungen zwischen Komponenten sind einfach
    - Techniken sind der Aufgabenstellung angemessen
    - Umfang des SW-Systems entspricht der Aufgabenstellung
  - Testbarkeit



Zwischenprodukt	funktionale Spezifikation	Qualitätsmerkmal	Wartbarkeit
<b>Ziel-Id</b>	WB11FS		
<b>Ziel</b>	Die funktionale Spezifikation hält vereinbarte Standards vollständig ein.		
<b>Frage-Id</b>	FS1		
<b>Frage</b>	Existiert für jeden Prozeß entweder eine Minispezifikation oder eine Verfeinerung in einem DFD?		
<b>Erläuterung/ Bemerkung/ Abgrenzung</b>	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.		
<b>Maß/ Bewertung</b>	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		
<b>Meßverfahren</b>	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.		
<b>Aspekte des Maßes</b>	Bedeutung des Maßes: 0 Durchgängigkeit: 0,5 Erhebungsaufwand (manuell/Werkzeug): 0,5/0		
<b>Regeln/ Handlungs- anweisungen/ Bemerkungen</b>	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.		



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

Legende: # = Anzahl



## Qualitätsmerkmale und 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