

Software- Qualitätsmanagement

**Vorlesung im Modul 10-202-2319
Software-Management**

Sommersemester 2009

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

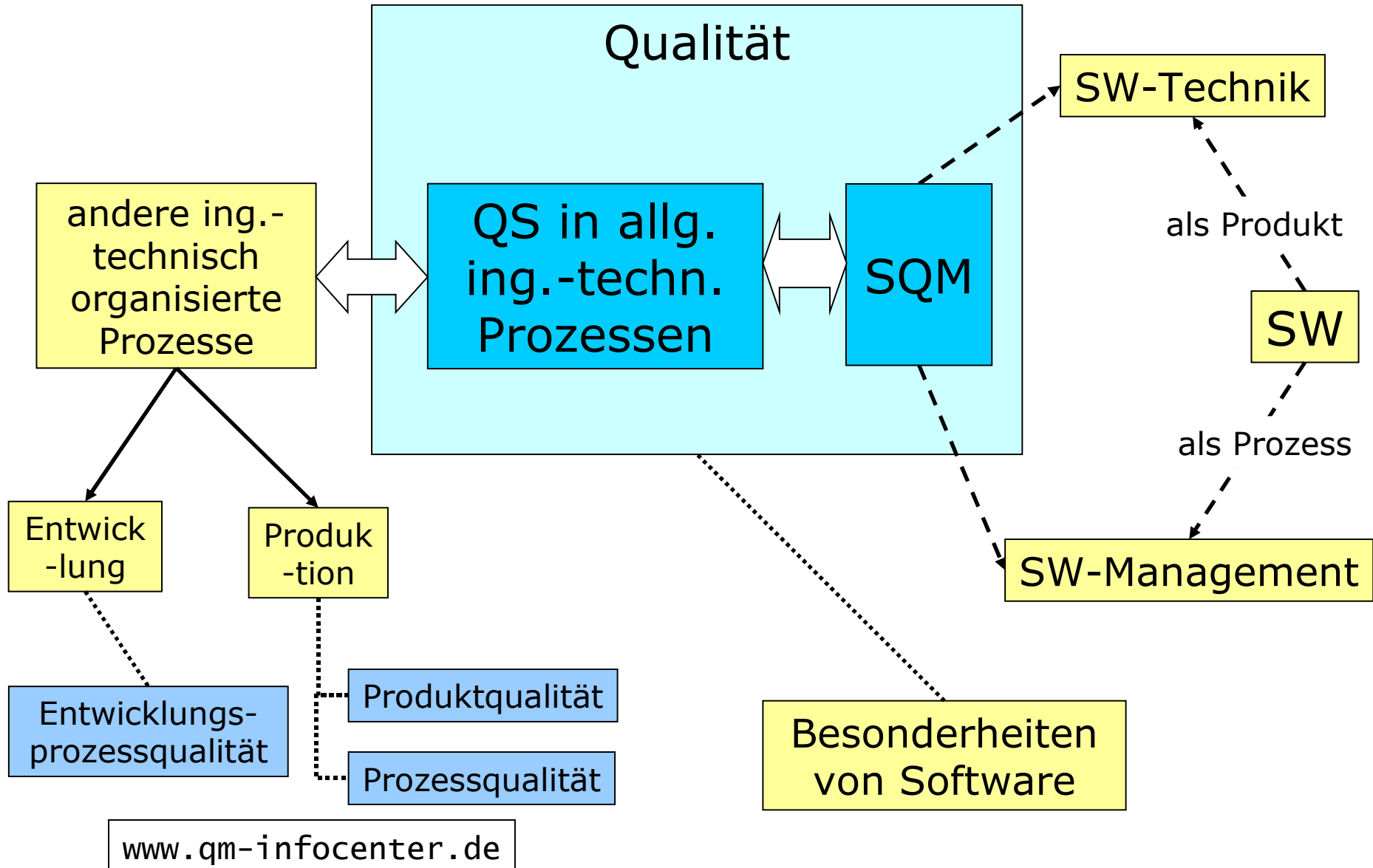
Begleitliteratur: Helmut Balzert, Lehrbuch der Software-Technik,
Bd. 2, Spektrum-Verlag

Mehr zu Vorlesung und Modul im BIS-OLAT-Portal

<http://olat.informatik.uni-leipzig.de>

im Kurs **S09.BIS.SW-Management**.

Siehe dort insbesondere Folien zur Vorlesung sowie das Forum.



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.

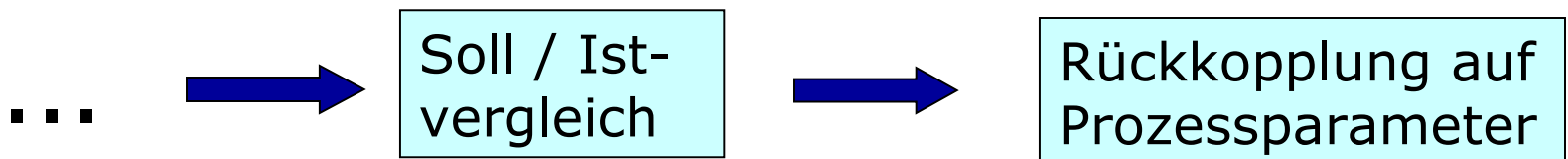
FCM – definitorischer Ansatz

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.

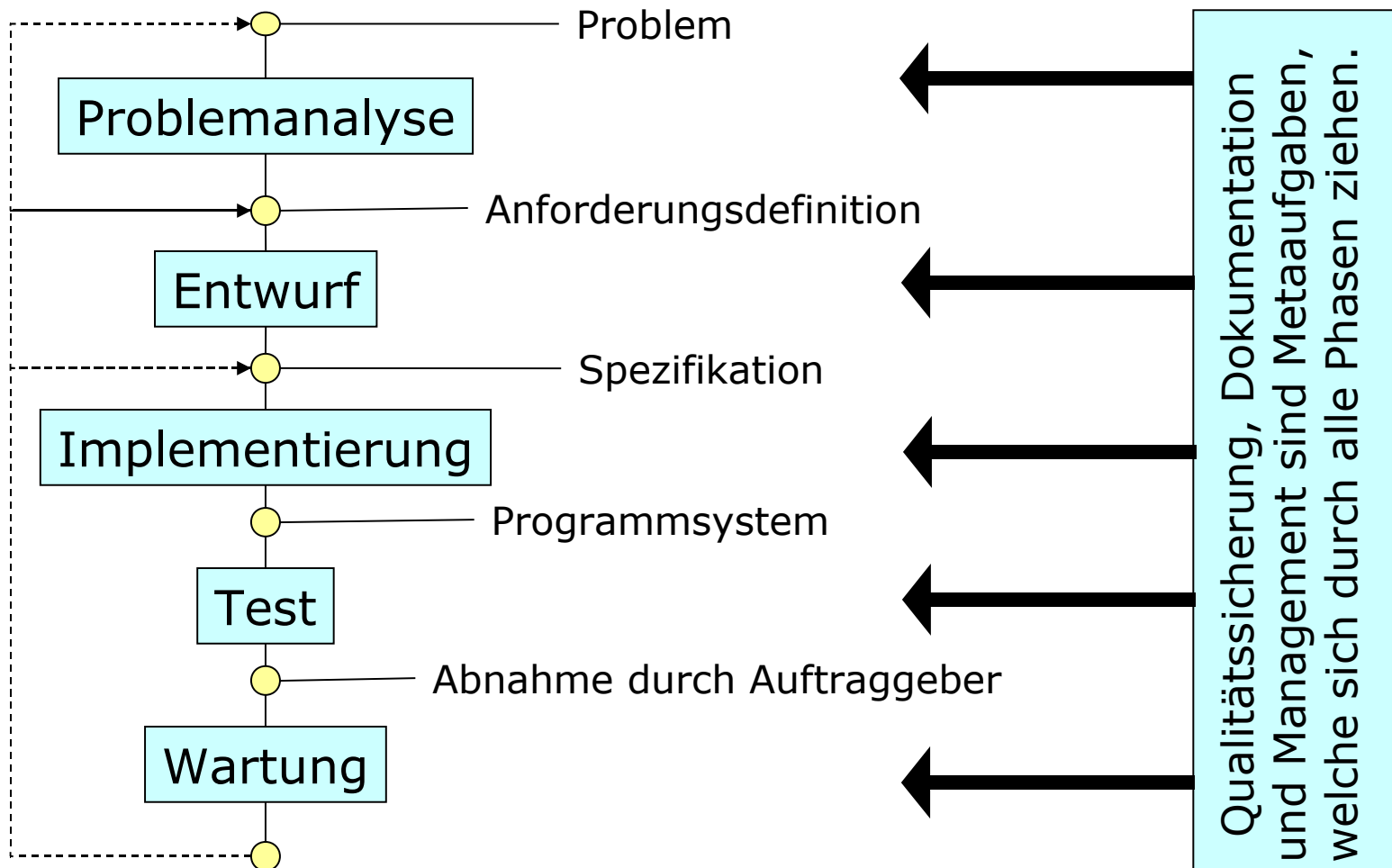
Softwarequalitätsmanagement ähnelt damit am meisten dem Entwicklungsprozess-Qualitätsmanagement, mit der Besonderheit, dass das „Entwicklerteam“ (Entwurf, Installation, Wartung) im Vergleich zu klassischen ingenieurtechnischen Prozessen deutlich weniger eng gekoppelt ist.

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?

- Relevante Norm ist die ISO/IEC 9126 „Softwarequalität“
- Sie definiert 6 **Merkmale** für die Qualität von Software-Produkten:
 - **Funktionalität**: Vorhandensein von Funktionen mit festgelegten Eigenschaften, welche die definierten Anforderungen erfüllen.
 - Richtigkeit, Angemessenheit, Interoperabilität, Konformität, Sicherheit
 - **Zuverlässigkeit**: Fähigkeit der Software, ihr Leistungsniveau unter festgelegten Bedingungen in einem festgelegtem Zeitraum zu entfalten.
 - Reife, Fehlertoleranz, Robustheit, 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, Attraktivität

- **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, Koexistenzfähigkeit