

Software- Qualitätsmanagement

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

Sommersemester 2014

Prof. Dr. Hans-Gert Gräbe

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

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

Ältere Qualitätsmodelle

- Zuordnung von Qualitätsmerkmalen zu Sichten
 - **Produkt-Einsatz** (Nutzbarkeit, Integrität, Effizienz, Sicherheit, Zuverlässigkeit)
 - **Produkt-Revision** (Wartbarkeit, Testbarkeit, Flexibilität)
 - **Produkt-Weiterentwicklung** (Wiederverwendbarkeit, Portierbarkeit, Interoperabilität)
 - Sichten als Ausdruck der Relativität des Qualitätsbegriffs.
- Untergliederung der Qualitätsmerkmale in die Bereiche Anwendernutzung, Portabilität und Wartbarkeit
- FURPS (Functionality-Usability-Reliability-Performance-Supportability)
 - 1985 von der Firma HP entwickelt, um die Qualität ihrer Produkte zu verbessern. Führt (nach eigenen Angaben) zu einer Reduzierung der Fehler und einer gleichzeitigen Kostensenkung bei der Firma HP.
- DIN 66272 – Qualitätsmerkmale für Software und Leitfaden zu deren Verwendung (1994, im Mai 2006 ersatzlos zurückgezogen)

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

Darauf kommen wir im zweiten Teil der Vorlesung zurück.

Strukturmodelle

- FCM-Ansatz (factor – criteria – metrics)
- Qualität wird durch ein **Qualitätsmodell** beschrieben.
- Ein Qualitätsmodell operationalisiert den allgemeinen Qualitätsbegriff strukturell durch Ableitung von Unterbegriffen: **Qualitäts-Merkmale** 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.

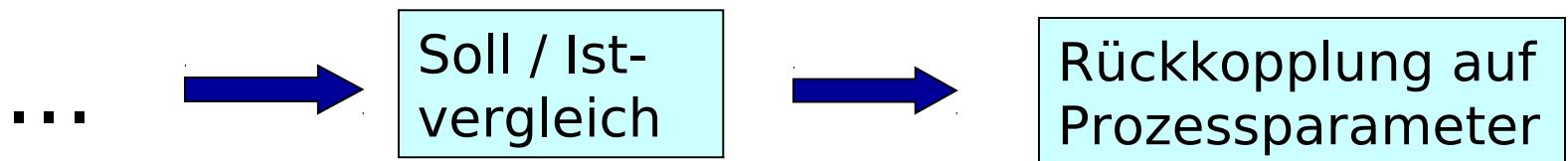
FCM – definatorischer Ansatz (Welche Produktqualität?)

Ausgerichtet auf Bewertung der Qualität eines Produkts



Erweiterung – prozessualer Ansatz (Welche Prozessqualität?)

schließt auch das Prozessmanagement ein.



Kann weniger formalisiert sein.

Vorgehensmodelle

Um Produktqualität zu erreichen, müssen die Qualitätsmaßstäbe aus einem Strukturmodell in einem Vorgehensmodell operationalisiert werden.

Umfassender methodischer Zugang:
Goal-Question-Metric-Ansatz (GQM)

1. Definiere Auswertungsziele
2. Leite alle Fragenstellungen ab
3. Leite alle Maße ab
4. Erfasse die Messwerte
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

Wichtung von Qualitätsmerkmalen: 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

Wichtung von Qualitätszielen: Kritikalität

Kritikalität gibt an, welche Bedeutung dem Fehlverhalten einer physischen oder logischen Einheit zugemessen wird.

Hängt vom Einsatzzweck ab und sollte projektspezifisch durch Abschätzung der Auswirkungen direkten oder indirekten Fehlverhaltens erfolgen.

Beispiele:

bei administrativen Systemen

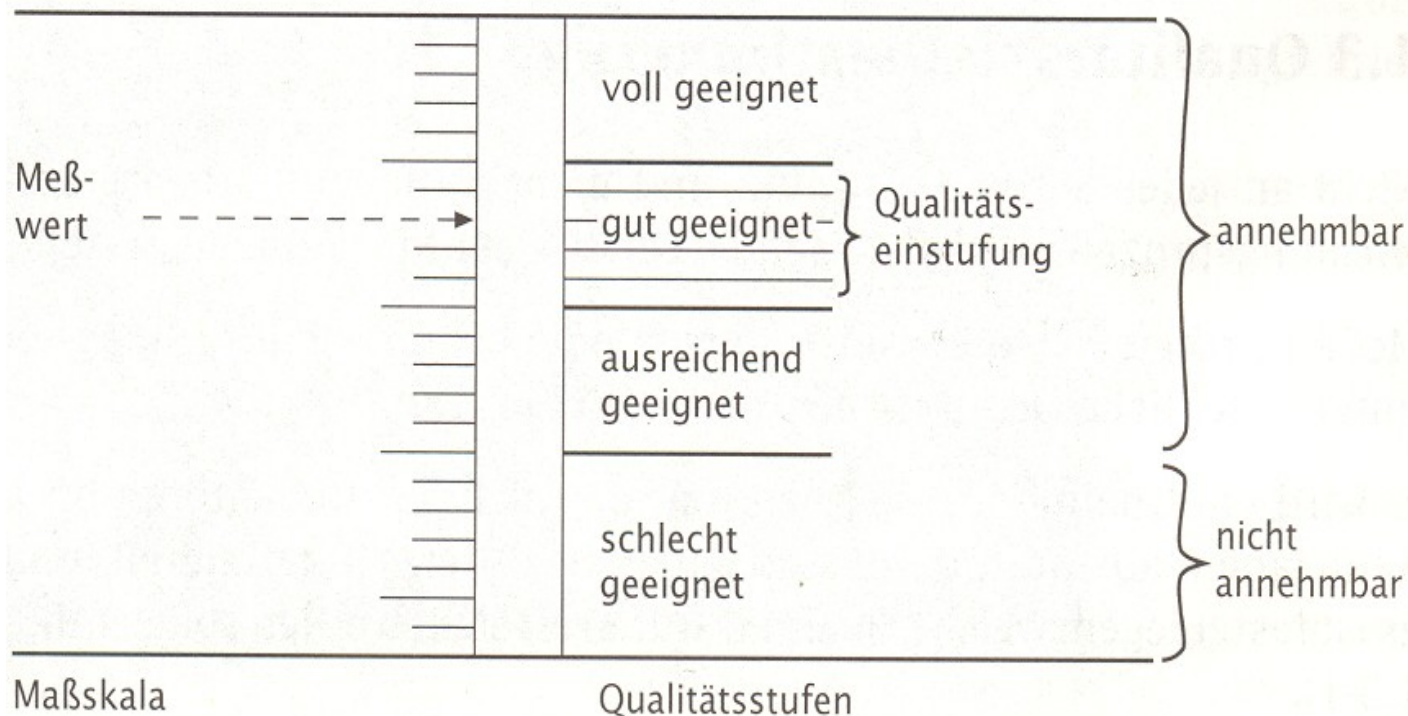
- sensitive Daten werden für unberechtigte Personen zugänglich (hoch)
- verhindert Zugang zu regelmäßig benötigten Daten (niedrig)

bei technischen Systemen

- Verlust von Menschenleben möglich (hoch)
- keine Gefährdung von Gesundheit oder Sachgütern (keine)
- bei Realzeitanwendungen (Flugsicherung)
- fehlerhafte Positionsangaben der Flugsicherung (hoch)
- Ausfall von Plandaten, die zu Abflugverzögerungen führen (niedrig)

3. Qualitätszielbestimmung

- Für Indikatoren, die nur qualitativ erfasst werden können oder zur vereinfachten Handhabung sind **Qualitätsstufen** zu definieren, und es ist festzulegen, welche Stufen erreicht werden sollen.
- Eine Qualitätsstufe ist ein Wertebereich auf einer Skala, dem eine bestimmte Qualitätsforderung zugeordnet ist.



3. Qualitätszielbestimmung

In der Regel ist eine Qualitätszielbestimmung pro Projekt erforderlich und als **Qualitätsanforderung** zu fixieren.

- Legen schriftlich fest, welche Qualitätsziele als relevant betrachtet werden und wie diese erreicht werden sollen.
- Manchmal reicht eine Qualitätszielbestimmung für eine ganze *Klasse* ähnlicher Software-Projekte aus.

Der Geltungsbereich von Qualitätszielen ist nicht nur auf Projektebene relevant, sondern kann sich erstrecken auf:

- eine ganze Software produzierende Einheit (Entwickler-Team),
- auf Teilprojekte innerhalb eines Software-Projekts,
- auf den Software-Erstellungsprozess eines Unternehmens,
- auf Teile dieses unternehmensweiten Software-Erstellungsprozesses.

Die Qualitätsanforderungen sind im Zuge der Projektplanung zu fixieren und z.B. im Pflichtenheft zu dokumentieren, da die zu erreichenden Qualitätsparameter Auswirkungen auf Termin und Kosten haben.

Qualitätslenkung

- Reguläre Aktivitäten: Durch entwicklungsbegleitende **Qualitätsprüfungen** ist die Umsetzung der Qualitätsanforderungen sicherzustellen
- Besondere Aktivitäten: Das Eintreten vorhergesehener Risiken oder neue, die Qualität betreffende Ergebnisse führen zu Abweichungen von der Roadmap des Projekts mit Auswirkungen auf die Qualitätslenkung, die bis zu einer Neuaufnahme der Qualitätszielbestimmung gehen kann.
- Finale Aktivitäten: Im Rahmen einer Endprüfung (Meilenstein) sind die erreichten Qualitätsparameter umfassend zu begutachten. Sind alle Anforderungen erfüllt, kann bei der Abnahme ein entsprechendes **Produktzertifikat** vergeben werden.

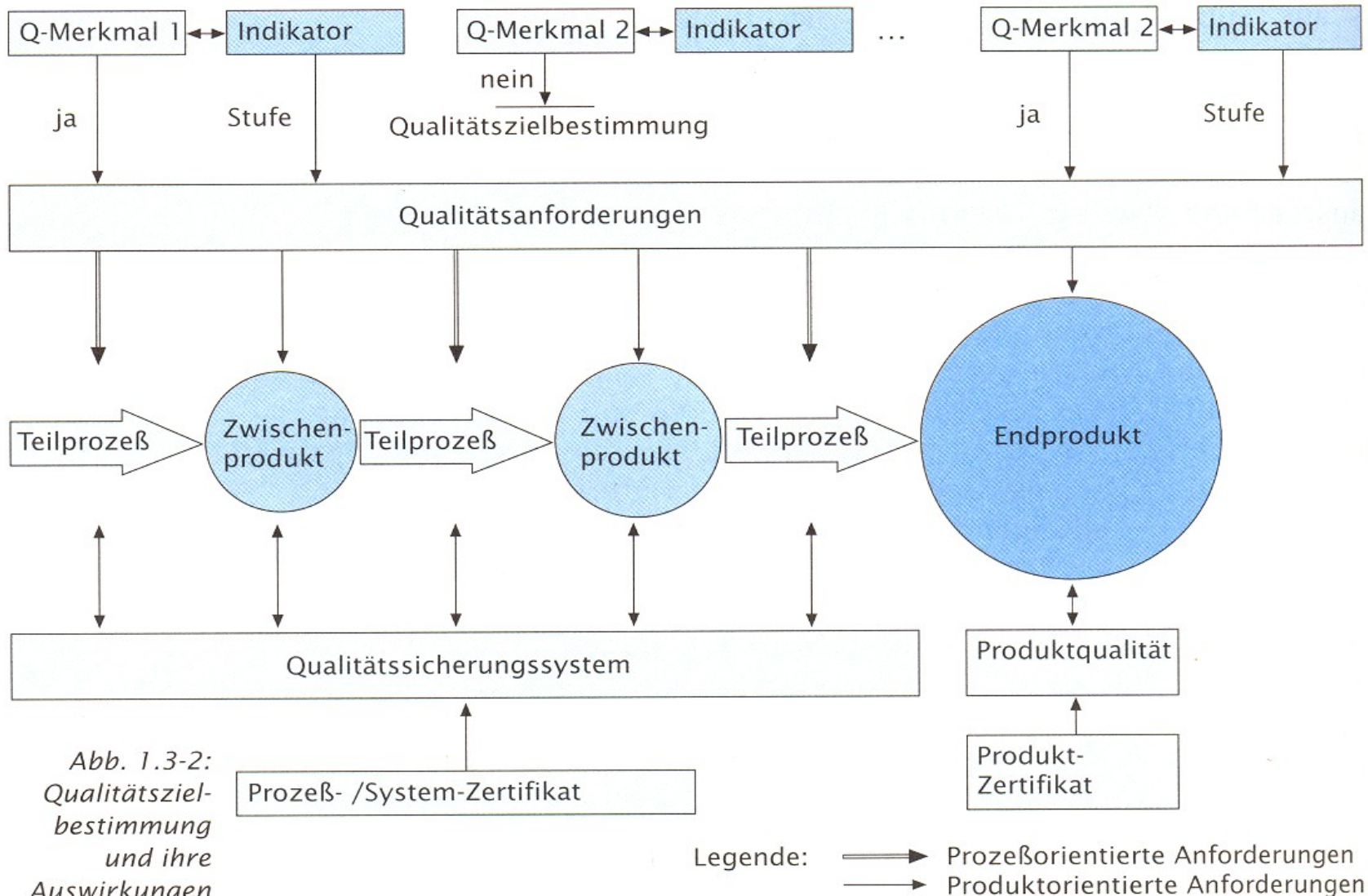


Abb. 1.3-2:
Qualitätsziel-
bestimmung
und ihre
Auswirkungen

- FCM-Modell =

typisches **Strukturmodell**, über welches der Qualitätsbegriff im konkreten Kontext spezifiziert 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 operativen 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.

Ziel ist es, dies zunächst auf der Ebene der einzelnen Software-Projekte durchzusetzen.

In einer weiteren Entwicklungsstufe des Unternehmens kann dann ein **unternehmensweites Qualitätsmodell** entwickelt werden – eine unternehmensweit gültige **Systematik**, nach welcher die Q.-Ziele festgelegt, in Kriterien operationalisiert und mit entsprechenden Indikatoren untersetzt werden.

1. Aufgaben im Qualitätsmanagement
2. Konstruktive und analytische Maßnahmen
3. Qualitätssicherung und Systementwicklung
4. Aktivitäten im Qualitätsmanagement
5. Prinzipien der Software-Qualitätssicherung
6. Beispiel: Qualitätssicherung im V-Modell

1. Aufgaben im Q.-Management

Qualitätsmanagement umfasst alle Tätigkeiten der Gesamtführungsaufgabe, welche die Qualitätspolitik, Ziele und Verantwortlichkeiten festlegt sowie diese durch Mittel wie Qualitätsplanung, Qualitätslenkung, Qualitätssicherung und Qualitätsverbesserung im Rahmen des Qualitätsmanagementsystems verwirklichen.

[DIN ISO 8402]

- **Q.-Planung:** Vorbereitende Maßnahmen
 - **Q.-Sicherung:** Begleitende Maßnahmen mit
 - **Q.-Lenkung:** administrative Maßnahmen
 - **Q.-Prüfung:** diagnostische Maßnahmen
- sowie
- **Q.-Verbesserung:** Prozess-strukturelle Maßnahmen

1. Q.-Management - zwei Ansätze

Produktorientiertes Q.-Management

Fokus: Operationalisierung der Qualität der ausgelieferten Software

- Produkte und Zwischenergebnisse werden auf vorher festgelegte Qualitätsmerkmale überprüft
- Qualität wird im Nachhinein festgestellt
- Relevant: Gütebedingungen und Prüfbestimmungen
- Angewendet eher im Bereich der Komponentensoftware und Standardsoftware mit konstanten Q.-Anforderungen

Grundansatz: Qualität als messbare Größe des *Produkts*

- Qualität kann durch Zertifikat (Prüfung durch unabhängige Seite) bestätigt werden
- Relevante Bestimmungen: ISO 9126

Methodik: analytische und konstruktive QS-Maßnahmen

1. Q.-Management - zwei Ansätze

Prozessorientiertes Q.-Management

Fokus: Operationalisierung der Qualität des Erstellungsprozesses der Software selbst

- Eher in Firmen, die anwenderspezifische Spezialsoftware herstellen, mit variierenden Q.-Anforderungen und dynamischem Qualitätsoptimum, wo die Qualität eher durch qualifiziert tätiges Personal bestimmt wird.
- Ziel ist die Herausbildung eines Qualitätsbewusstseins bei den Mitarbeitern

Grundansatz: Qualität durch den Erstellungsprozess selbst

Faktoren: Planbarkeit von Prozessen, Effizienz (im Kosten/Nutzen-Sinn), Systematik der Qualität der Arbeitsprodukte

Methodik: Prozesszertifizierung, Prozessverbesserung

Konstruktive Maßnahmen

Vorgabe von Konstruktionstechniken und Richtlinien

- strukturiertes Vorgehen
- Werkzeug gestützte Entwicklung
- höhere Programmiersprachen

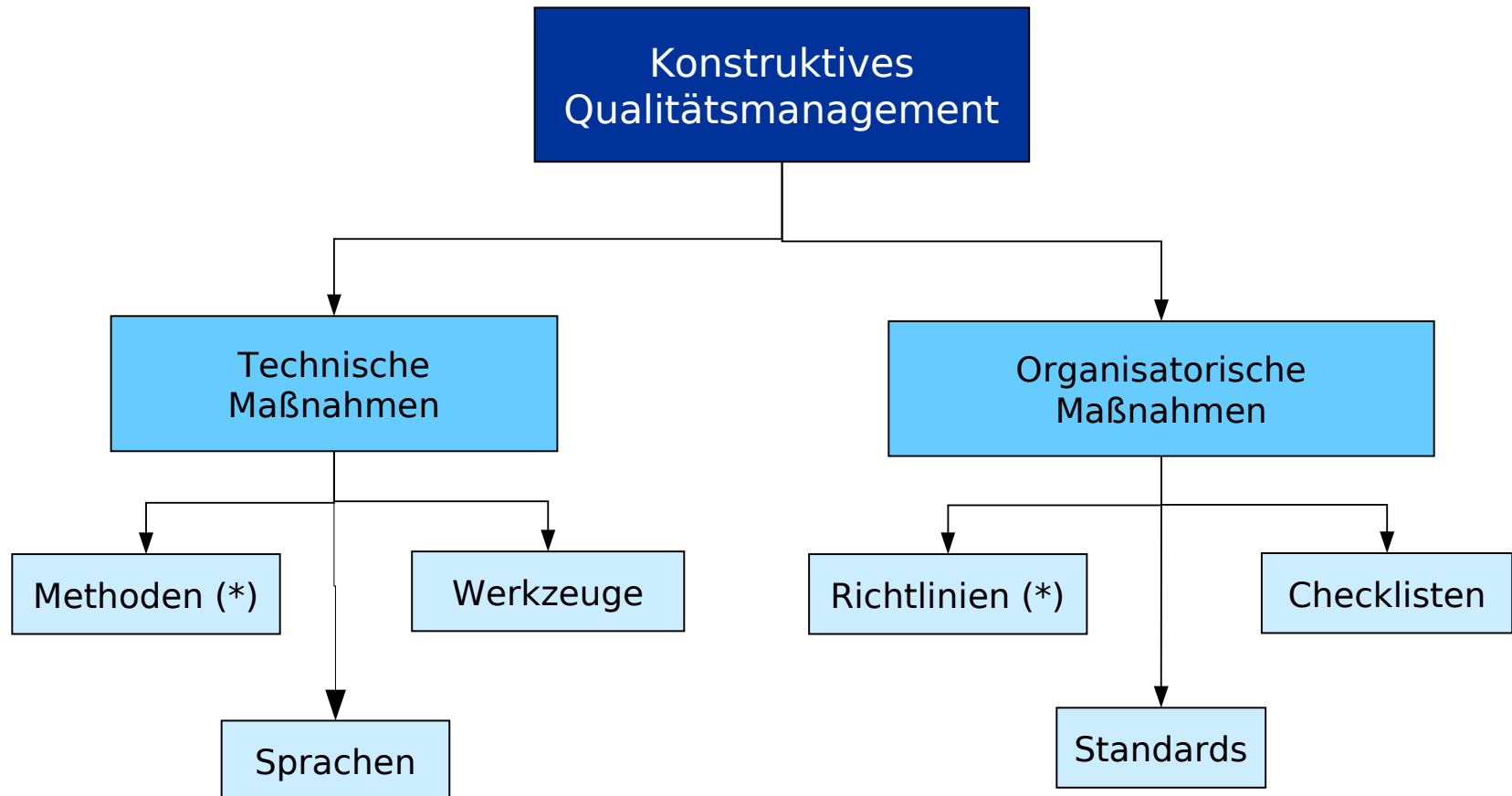
Vorteile:

- Erfahrungen projektübergreifend sammeln und nutzen
- Aufwertung der Planungsaktivitäten in frühen Projektphasen
- Werkzeugunterstützung

Nutzen:

- Steigerung der Qualität um bis zu 50 %
- Steigerung der Produktivität um bis zu 30 %

Konstruktive Maßnahmen sorgen durch Einschränkung der Variabilität in der Systementwicklung von vornherein dafür, dass gewisse Fehler nicht auftreten können und damit ein gewisses Maß an Qualität per se erreicht wird.



(*) - dazu Beispiele auf den nächsten Folien

Beispiel für konstruktive Verfahren: Methoden

Ziel: strukturierte Vorgehensweise

Technik: Vorgabe von Zwischenprodukten

- Vorgabe von Modellen (Bsp.: objektorientiert)
- Vorgabe von Einzelschritten (Bsp.: Anwendungsfall-Modellierung)
- Vorgabe von Erstellungsmitteln (Bsp.: Klassendiagramm, Anwendungsfall-Diagramm)

Vorteile:

- Strukturierung unterstützt gute Granularität, Änderbarkeit
- Werkzeugunterstützung

Beispiel für konstruktive Verfahren: Richtlinien

Ziel: Produkteigenschaften a-priori festlegen

Technik:

- Vorgabe von Checklisten, Schablonen
- Überprüfung der Richtlinien

Beispiele:

- Anwendung von Design Pattern
- Einsatz von Coding Standards

Vorteile:

- Erfahrungen mit Richtlinien werden projektübergreifend wirksam
- Unterstützung durch Werkzeuge und Vorlagen

Analytische Maßnahmen

- diagnostische Maßnahmen, bringen keine Qualität per se
- sind zur Messung der Qualität der End- bzw. Zwischenprodukte
- Gliederung nach verschiedenen Gesichtspunkten:
 - Bezug der Prüfung (Produkt oder Prozess)
 - Automatisierungsgrad der Prüfung (manuell / mit Werkzeug)
 - Nachvollziehbarkeit der Prüfung (Selbstprüfung / Nachweis)
 - Einsatzbereich der Prüfung (in welcher Phase des SW-Zyklus)

Analytische Maßnahmen dienen zur Datenerhebung, um Ist- und Soll-Zustand zu vergleichen und so den Grad der erreichten Qualität im Nachhinein festzustellen.

Analysierende Verfahren sammeln gezielt Informationen über den Prüfling mit analytischen Mitteln.

