

# **Software- Qualitätsmanagement**

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

Sommersemester 2011

Prof. Dr. Hans-Gert Gräbe

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

### Qualitätszielbestimmung

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

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

Methodischer Zugang: *Goal-Question-Metric-Ansatz (GQM)*

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

## 4. Qualitätszielbestimmung

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/ Handlungsanweisungen/ 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.		



## 4. Qualitätszielbestimmung

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

### 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)

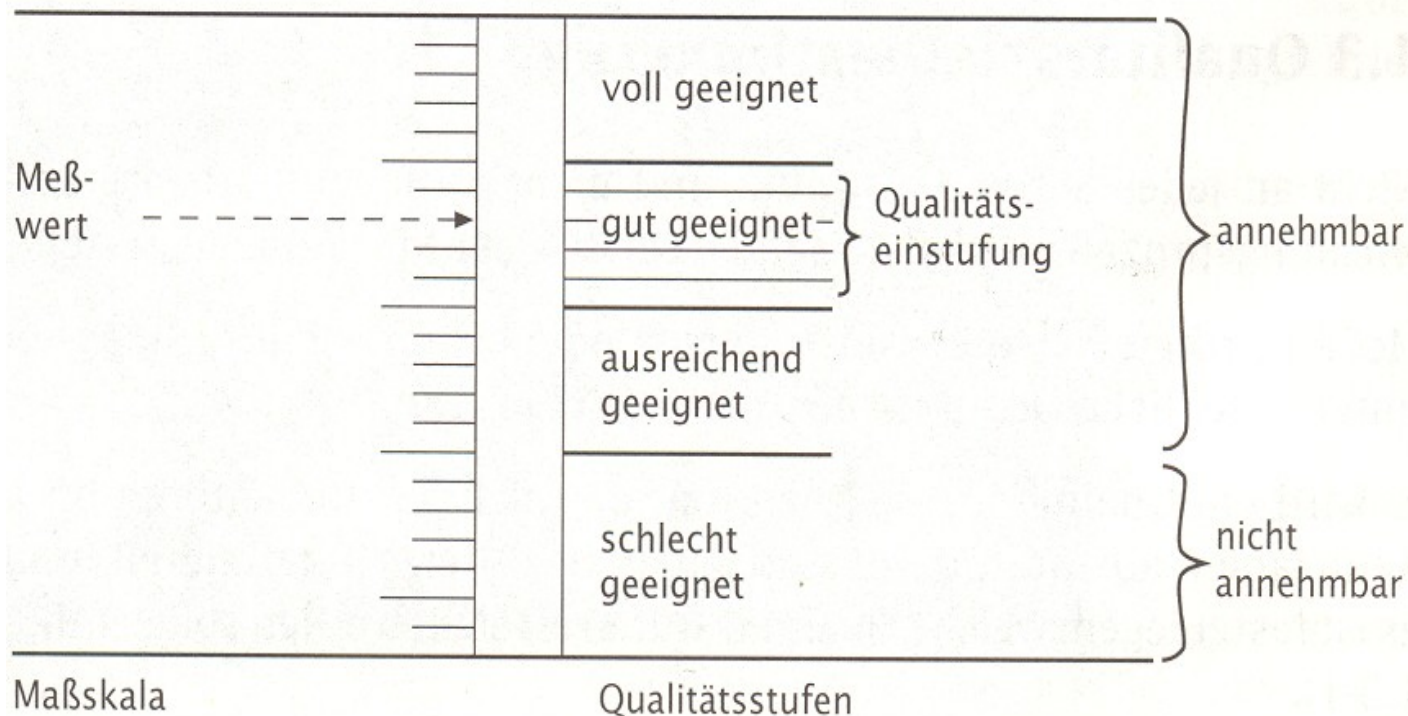
bei Realzeitanwendungen (Flugsicherung)

- fehlerhafte Positionsangaben der Flugsicherung (hoch)
- Ausfall von Plandaten, die zu Abflugverzögerungen führen (niedrig)



## 4. 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.



## 4. Qualitätszielbestimmung

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.

Der Geltungsbereich von Qualitätszielen kann sich erstrecken auf:

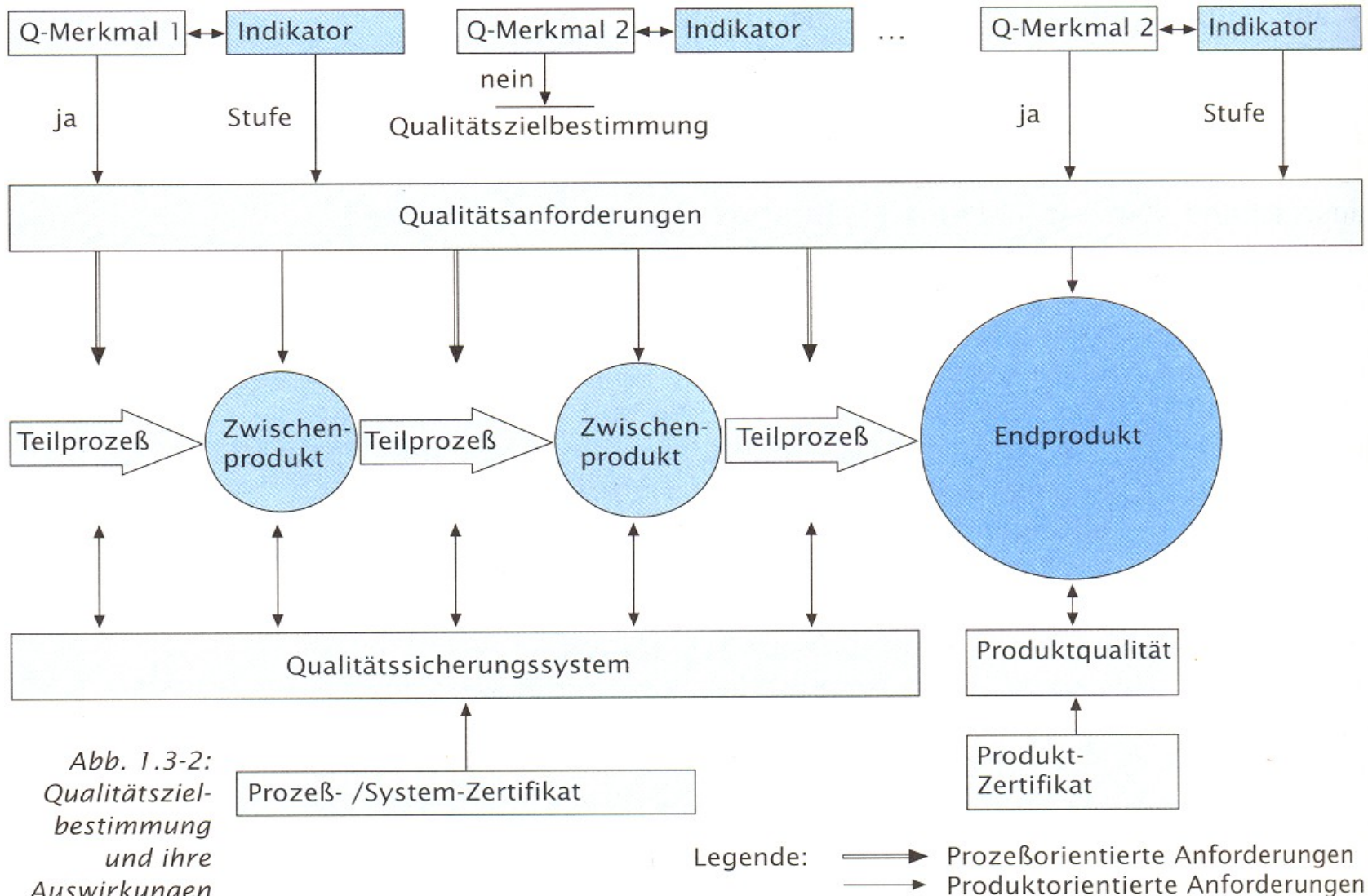
- eine Software produzierende Einheit,
- auf Teilprodukte eines Software-Produkts,
- auf den gesamten Software-Erstellungsprozess,
- auf Teile des Software-Erstellungsprozesses.

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.

### Qualitätslenkung

- Reguläre Aktivitäten: Durch entwicklungsbegleitende **Qualitätsprüfungen** sind die Anforderungen sicherzustellen
- Besondere Aktivitäten: Neue, die Qualität betreffende Ergebnisse erfordern eine Wiederholung der Qualitätszielbestimmung.
- Finale Aktivitäten: Sind alle Anforderungen erfüllt, kann bei der Abnahme ein entsprechendes **Produktzertifikat** vergeben werden.



- 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.



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

Produkte und Zwischenergebnisse werden auf vorher festgelegte Qualitätsmerkmale überprüft

- Qualität wird im Nachhinein festgestellt
- Gütebedingungen und Prüfbestimmungen
- 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

**Kontext:** analytische und konstruktive QS-Maßnahmen

#### Prozessorientiertes Q.-Management

Gerichtet auf den Erstellungsprozess der Software selbst

- eher für Firmen, die anwenderspezifische Spezialsoftware herstellen, mit variierenden Q.-Anforderungen und dynamischem Qualitätsoptimum
- Ziel ist die Herausbildung eines Qualitätsbewusstseins bei den Mitarbeitern

**Grundansatz:** Qualität durch den Erstellungsprozess selbst

**Faktoren:** Planbarkeit, Effizienz (im Kosten/Nutzen-Sinn), Produktqualität

**Kontext:** Prozesszertifizierung, Prozessverbesserung

### Konstruktive Maßnahmen

Vorgabe von Konstruktionstechniken und Richtlinien

- strukturiertes Vorgehen
- Werkzeug gestützte Entwicklung
- höhere Programmiersprachen

Vorteile:

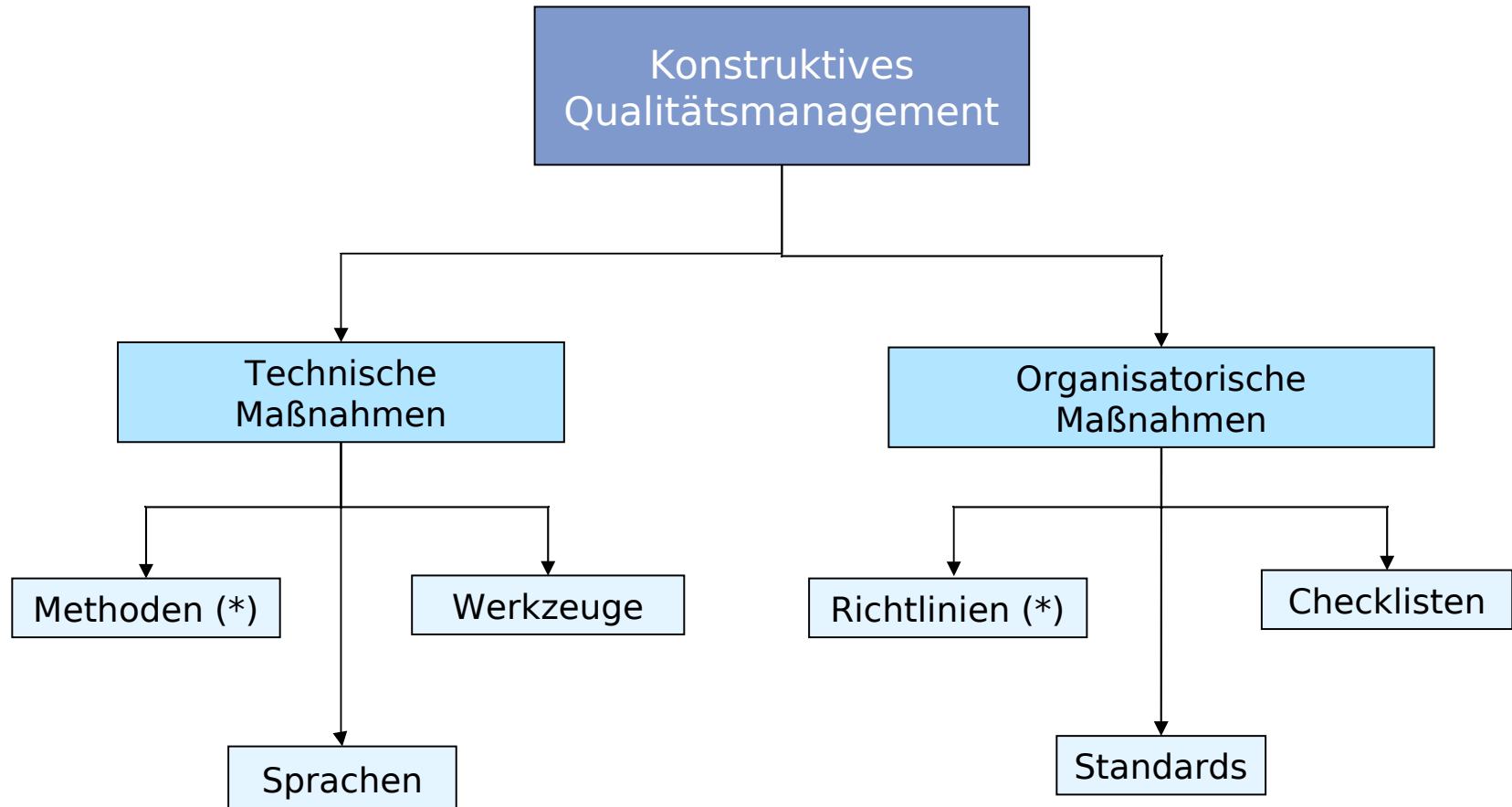
- Erfahrungen projektübergreifend sammeln und nutzen
- Aufwertung der Planungsaktivitäten in frühen Projektphasen
- Werkzeugunterstützung

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.





### 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:

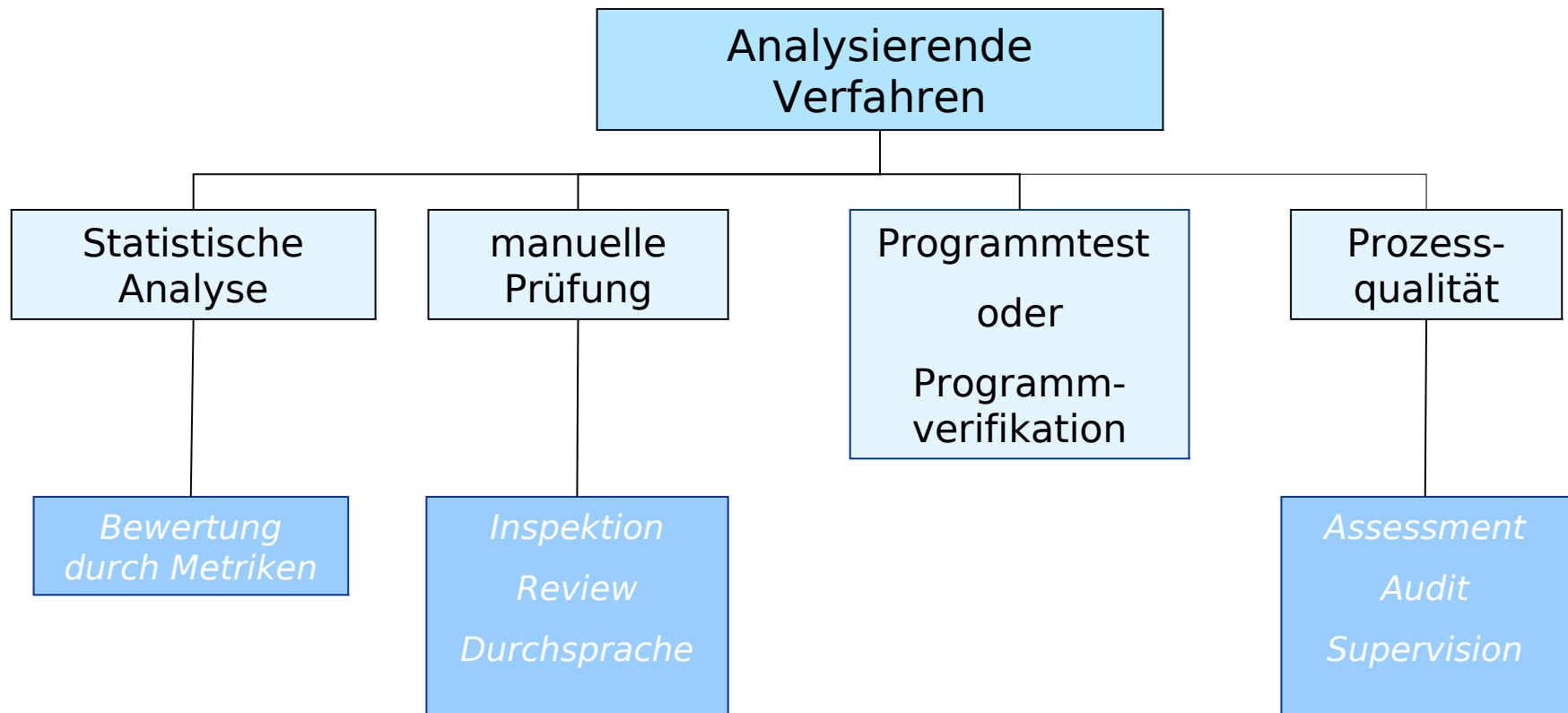
- Erfahrungen mit Richtlinien werden projektübergreifend wirksam
- Unterstützung durch Werkzeuge und Vorlagen

### 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.

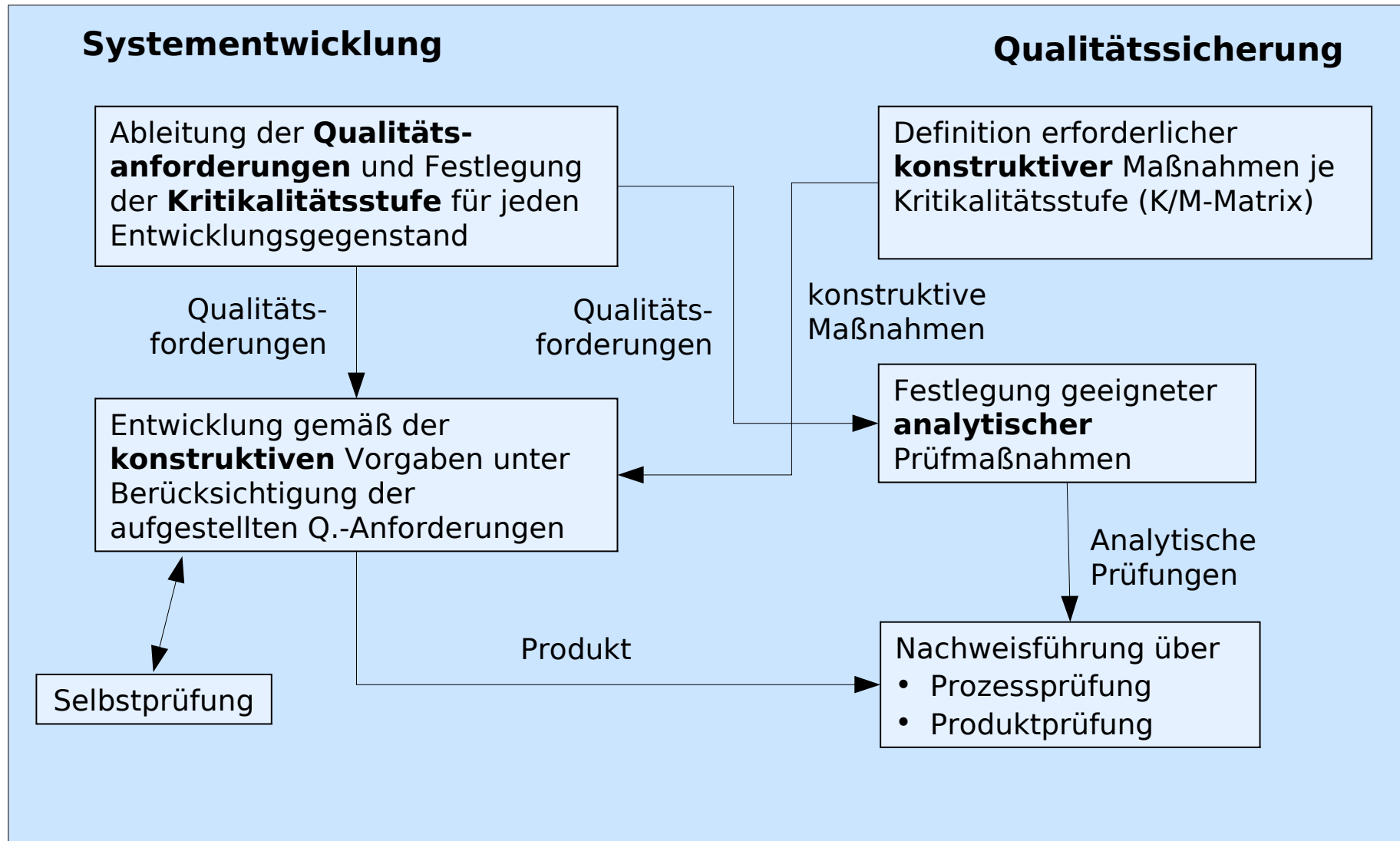




### Qualitätssicherung und Systementwicklung

- konstruktive Maßnahmen sind im Rahmen der Anforderungsanalyse und vor Beginn des Entwicklungsprozesses zu fixieren
- analytische Maßnahmen werden Entwicklungsprozess begleitend oder zu Meilensteinen, mit denen Entwicklungsprozess-Etappen abgeschlossen werden, wirksam
- analytische Maßnahmen können konstruktive Vorgaben erfordern, um die Produktion der zu analysierenden Daten zu initiieren.

Analytische und konstruktive QM-Maßnahmen beeinflussen sich gegenseitig: Vorausschauende konstruktive Planung erspart analytischen Aufwand



## Qualitätsplanung und Qualitätssicherung

Qualitätsmanagement besteht aus den Phasen

- **Qualitätsplanung**
  - vorbereitende Aktivitäten zur Festlegung von Standards, zu erreichender Parameter und von Verantwortlichkeiten
  - Ergebnis: Qualitätssicherungsplan und Prüfplan
- **Qualitätssicherung**
  - Systementwicklung begleitende Aktivitäten zur Umsetzung und Dokumentierung der erreichten Qualitätsparameter
  - Ergebnis: Protokolle, Zertifikate, Vorschläge für die Projektsteuerung
  - **Qualitätslenkung** und **Qualitätsprüfung**

#### Qualitätsplanung

- **Qualitätszielbestimmung:** Festlegung von Qualitätsanforderungen an den Prozess und an das Produkt in überprüfbarer Form.
- **Planung der Qualitätslenkung:** Planung der Umsetzung, Steuerung, Überwachung und Korrektur des Entwicklungsprozesses mit dem Ziel, die vorgegebenen Anforderungen zu erfüllen.
- **Planung der Qualitätsprüfung:** Planung der Durchführung der im Rahmen der Qualitätsplanung festgelegten Maßnahmen zur
  - Erfassung von Istwerten der Qualitäts-Indikatoren
  - Überwachung der Umsetzung der konstruktiven Maßnahmen
  - Tests, Reviews, Audits, Inspektionen
- **Planung der Qualitätsverbesserung:** Planung der Auswertung der Qualitätssicherungs-Ergebnisse und Prozessverbesserung.
  - Mängel- und Fehleranalyse (Verbesserung der Prozessqualität)

#### Qualitätssicherungsplan und Prüfplan

- Ergebnisse der Qualitätsplanung werden in einem **Qualitäts-Sicherungsplan** dokumentiert (prozess-orientiert), die begleitenden Maßnahmen in einem **Prüfplan** festgelegt (produkt-orientiert).
  - **Festlegung der Aufgaben**
    - Was ist zu tun?
    - Identifizierung der zu sichernden *Produkte*
    - Identifizierung der relevanten *Qualitätsmerkmale*, ihre relative *Bedeutung* und ihre *Quantifizierung* in Form von Metriken
  - **Festlegung der Vorgaben und Hilfsmittel**
    - Wie ist es zu tun?
    - Auswahl der zur Datenerfassung und Qualitätsprüfung geeigneten *Techniken* und *Methoden*
    - konstruktive Vorgaben (etwa Richtlinien, Vorlagen)
    - analytische Vorgaben (Verfahren, Werkzeuge)



#### Qualitätssicherungsplan und Prüfplan

- **Festlegung der Termine**
  - (Bis) wann ist es zu tun?
  - Festlegung der *Zeitpunkte* für die den gesamten Entwicklungsprozess begleitende *Datenerfassung*
  - Einordnung des Prüfplans in den Projektplan
- **Festlegung der Verantwortlichkeiten**
  - Wer hat es zu tun?
  - Festlegung der *Verantwortlichkeiten* für die Qualitätsprüfung und -lenkung.
  - Definition und Besetzung von *Rollen* (Q-Manager, Prüfer, Autor, Gutachter)
- Der IEEE-Standard 730 für den *Software Quality Assurance Plan* beschreibt den Aufbau eines solchen Qualitätssicherungsplans.

#### **Grundsätze für die Qualitätssicherung in der Software-Entwicklung**

- produkt- und prozessabhängige Qualitätszielbestimmung
- quantitative Qualitätssicherung
- maximale konstruktive Qualitätssicherung
- frühzeitige Fehlerentdeckung und -behebung
- entwicklungsbegleitende, integrierte Qualitätssicherung
- unabhängige Qualitätssicherung

#### Prinzip der produkt- und prozessabhängigen Qualitätszielbestimmung

Nur 50% der Betriebe legen überhaupt Qualitätsmerkmale fest

Explizite und transparente Qualitätszielbestimmung ist vor Beginn des Entwicklungsprozesses aber äußerst hilfreich.

- Die in der Qualitätszielbestimmung festgelegten Qualitätsanforderungen werden vom Auftraggeber für den Abnahmetest verwendet.
- Für den Software-Lieferanten ergeben sich aus den Qualitätsanforderungen die Maßnahmen für den Entwicklungsprozess und die Qualitätsprüfung.

**Vorteil:** Prinzip der produkt- und prozessabhängigen Qualitätszielbestimmung vor Beginn des Entwicklungsprozesses bringt Planungs- und Kalkulationssicherheit.

**Nachteil:** Qualitätssicherung beansprucht zusätzliche Ressourcen