

# **Software- Qualitätsmanagement**

**Kernfach Angewandte Informatik**

Sommersemester 2007

Prof. Dr. Hans-Gert Gräbe

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

- 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. Aktivitäten im Qualitätsmanagement
4. Prinzipien der Software-Qualitätssicherung
5. Beispiel: Qualitätssicherung im V-Modell

**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

## 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

- werkzeuggestützte Entwicklung

- höhere Programmiersprachen

Vorteile:

- Erfahrungen projektübergreifend sammeln und nutzen

- Aufwertung der Planungsaktivitäten in frühen Projektphasen

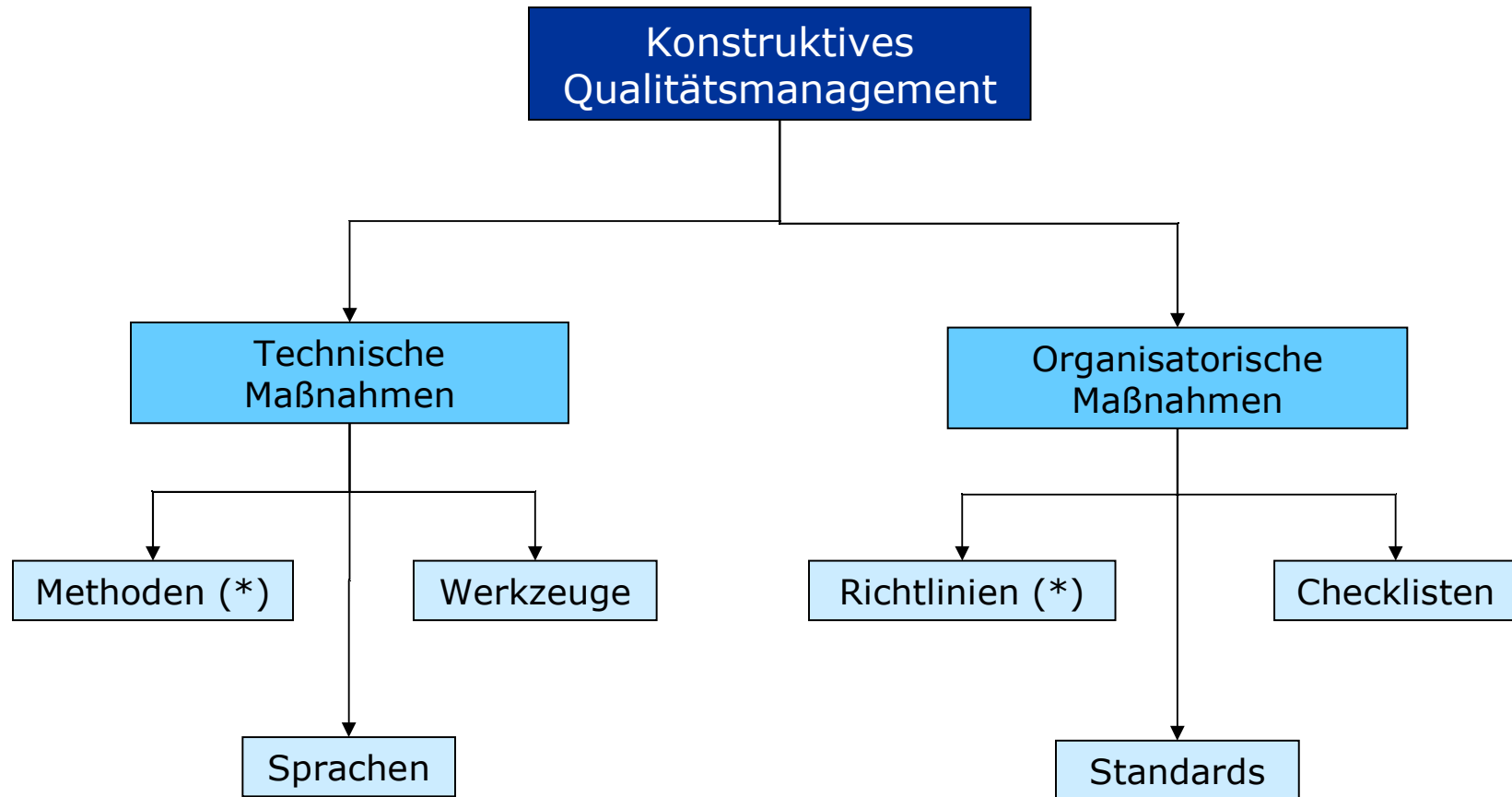
- Werkzeugunterstützung

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.





## 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

## **Konstruktive Verfahren: Richtlinien**

Ziel: Produkteigenschaften a-priori festlegen

Technik:

- Vorgabe von Checklisten, Schablonen

- Überprüfung der Richtlinien

Beispiele:

- Strukturierung der Analyse durch SCR-Tabellen

- Anwendung von Design Pattern

- Einsatz von Coding Standards

Vorteile:

- Erfahrungen mit Richtlinien werden projektübergreifend wirksam

- Unterstützung durch Werkzeuge und Vorlagen

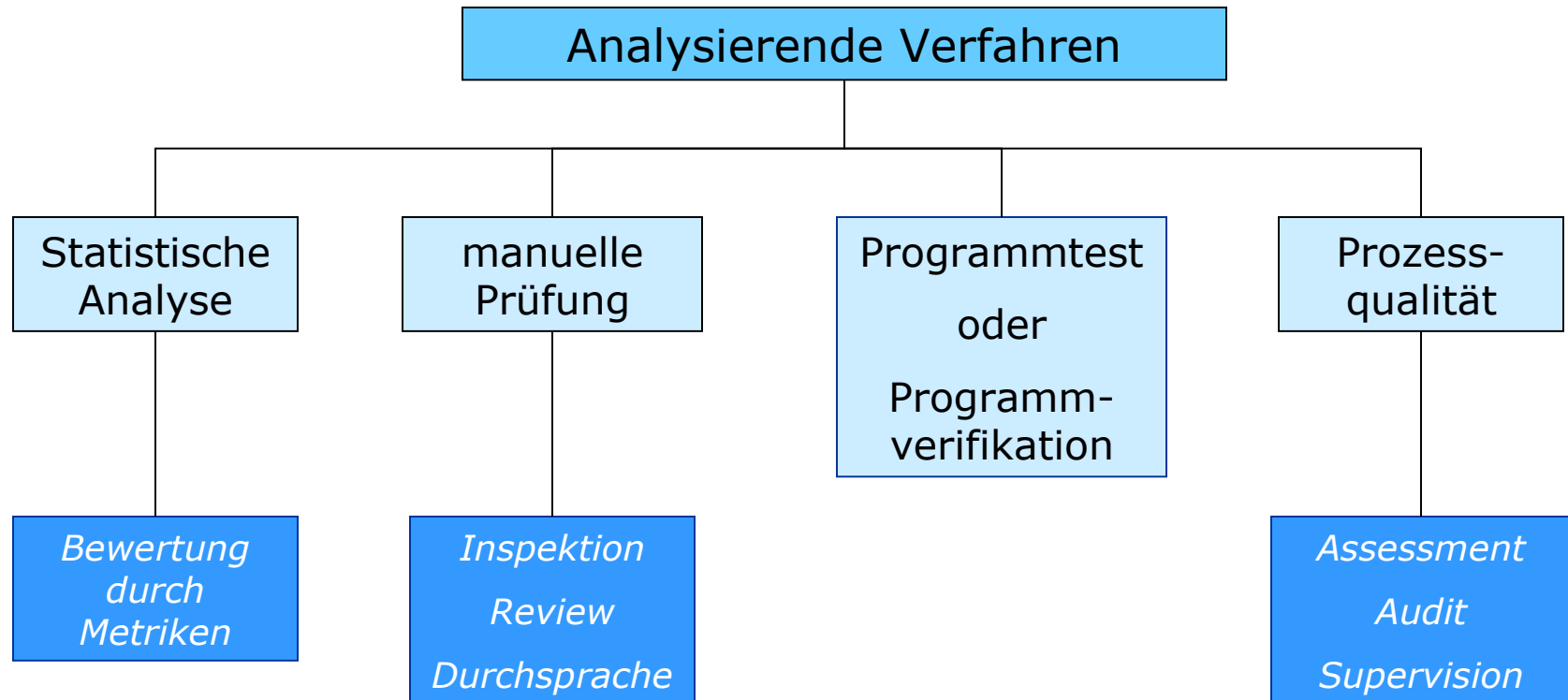
### **Analytische Maßnahmen**

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.



### Produktqualität: Statistische Analyse

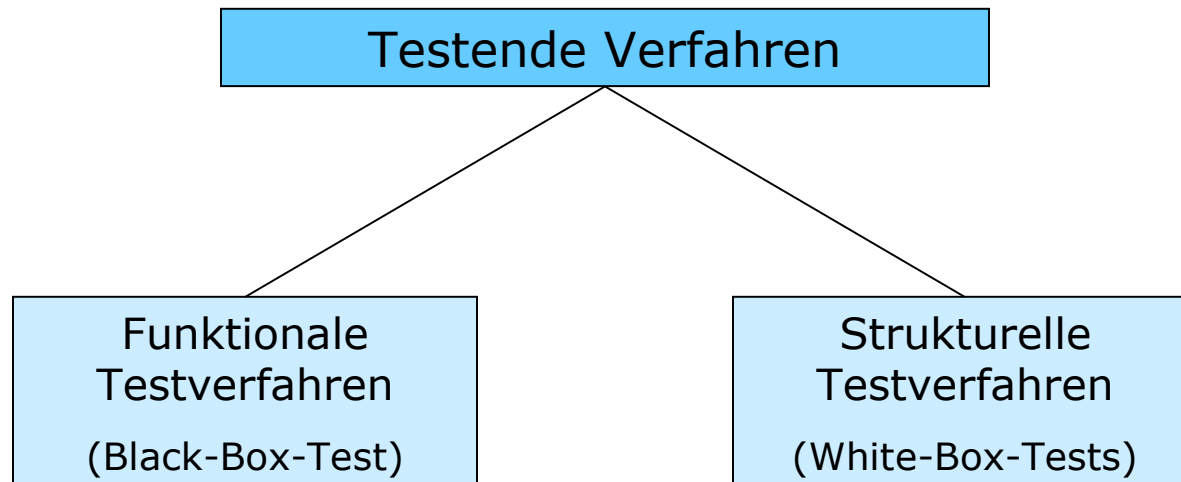
- **Ziel: Bewertung** eines Produkts (Entwurfsdokuments, Grob/Fein-Entwurf, Code, Designdokumentation usw.) mittels **Metriken**
- **Schwerpunkt:** Zuverlässigkeit, Änderbarkeit

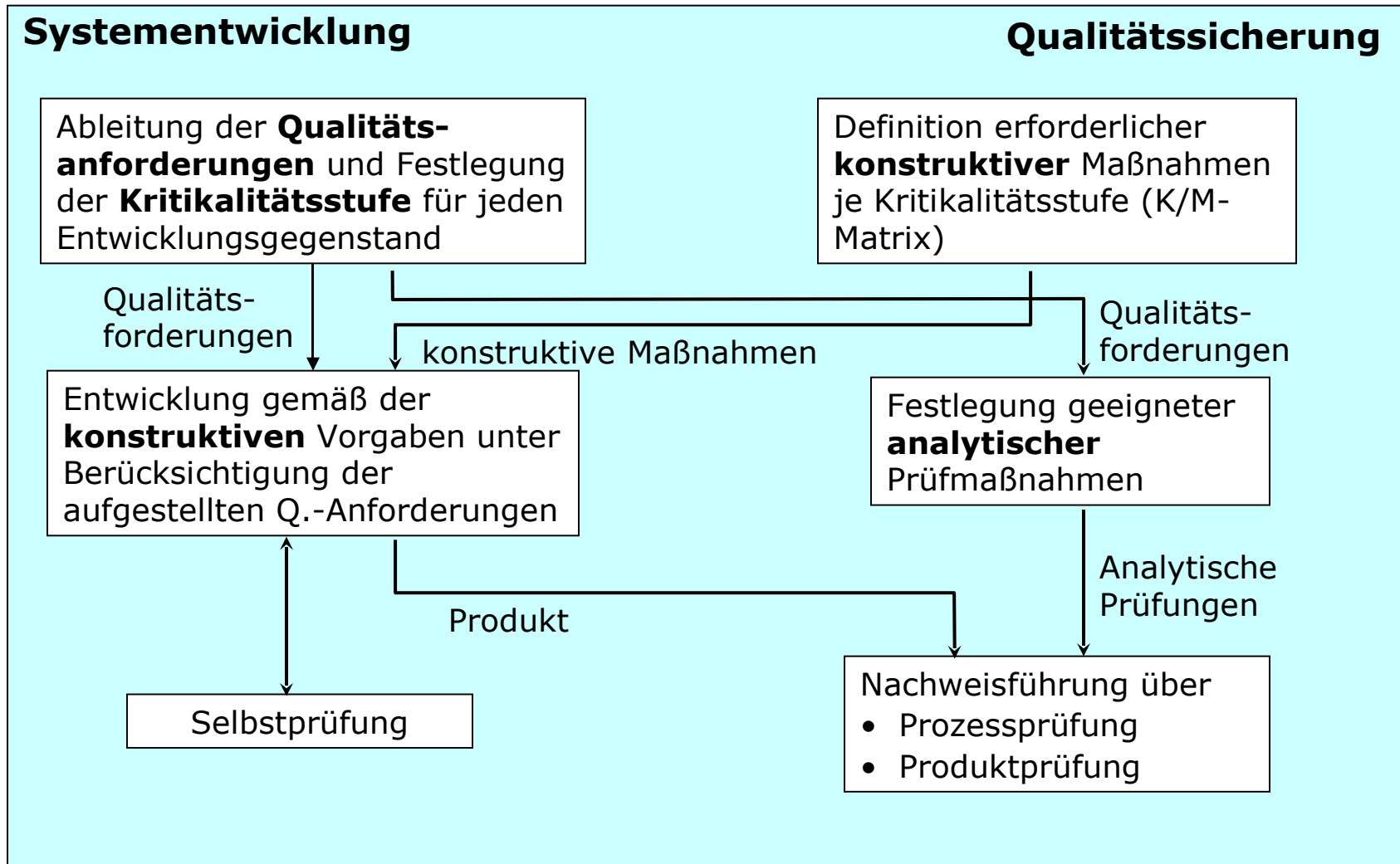
Einsatzgebiet	Kriterium	Metrik
Komponenten-analyse	Umfang	lines of code
	innere Struktur	Kontrollfluss-komplexität
	Schnittstelle	# Methoden pro Klasse Schnittstellenbreite

Einsatzgebiet	Kriterium	Metrik
Systemanalyse	Umfang	lines of code
	Kopplung	# Aufrufe in/aus Komponenten
	OO-Strukturierung	OO-Metriken
Prozessanalyse	Aufwandsoptimierung	Zeiterfassung
	Dokumentenqualität	entdeckte Fehler pro Seite
	Prüfprozessqualität	# vorab gefundener Fehler / # in der Sitzung gefundener Fehler

## Produktqualität: Testende Verfahren

Diese prüfen das Verhalten des Prüfling bei konkreten Eingaben.







## 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.  
**Qualitätslenkung:** Umsetzung, Steuerung, Überwachung und Korrektur des Entwicklungsprozesses mit dem Ziel, die vorgegebenen Anforderungen zu erfüllen.  
**Qualitätsprüfung:** 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  
**Qualitätsverbesserung:** 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)

- Gliederungsschema für Qualitätssicherungspläne ist im IEEE-Standard 730-1984 festgelegt.

### **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 Qualitätsmerkmale fest [Spillner et al. 94].
  - Qualitätsmerkmale mit hoher Priorität sind Robustheit, Verständlichkeit, Wartbarkeit und Laufzeiteffizienz.
  - Erst in zweiter Linie werden Korrektheit, Vollständigkeit und Benutzungsfreundlichkeit festgeschrieben.
- explizite und transparente Qualitätszielbestimmung ist vor Beginn des Entwicklungsprozesses 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.
- Prinzip der produkt- und prozessabhängigen Qualitätszielbestimmung vor Beginn des Entwicklungsprozesses bringt Planungs- und Kalkulations-sicherheit.

## Prinzip der quantitativen Qualitätssicherung

- Schwierigkeiten bei der Quantifizierung von Soll- und Istwerten:
  - Metriken sind ziel- und kontextabhängig
  - Anzahl der Variationsparameter ist um ein Vielfaches höher als bei traditionellen Produktionsprozessen
  - kreativer Charakter vieler Aspekte der Software-Entwicklung
  - unkontrollierte Variabilität von Entwicklungsprozessen
- **Vorteile:**
  - Messen ist geeignet
    - o zum besseren Verständnis unterschiedlicher Qualitätsmerkmale
    - o zur besseren Planung und Sicherung von Qualitätsmerkmalen
    - o zur Verbesserung von Entwicklungsansätzen
  - Methoden und Werkzeuge zur Planung und Durchführung der Datenerfassung sind schon vorhanden.
  - Auch zur Auswertung und Präsentation von Messdaten können vorhandene Werkzeuge verwendet werden.



### **Prinzip der maximalen konstruktiven Qualitätssicherung**

- „Fehler die nicht gemacht werden, brauchen auch nicht behoben werden“  
ist das Ziel, das von der maximalen konstruktiven Qualitätssicherung verfolgt wird.
- Reduzierung der analytischen Maßnahmen über Einschränkung der Variationsbreite durch voraussblickende konstruktive Maßnahmen.
- **Vorteile:**
  - direkte Verbesserung der Produktivität
  - Reduktion analytischer Maßnahmen
  - Voraussetzung für analytische Maßnahmen
  - Vermeidung von Fehlern