

Software- Qualitätsmanagement

Kernfach Angewandte Informatik

Sommersemester 2005

Prof. Dr. Hans-Gert Gräbe



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 Verantwortungen festlegt sowie diese durch Mittel wie Qualitätsplanung, Qualitätslenkung, Qualitätssicherung und Qualitätsverbesserung im Rahmen des Qualitätsmanagementsystems verwirklichen.

[DIN ISO 8402]

also

- **Q.-Planung:** Vorbereitende Maßnahmen
 - **Q.-Lenkung:** Begleitende administrative Maßnahmen
 - **Q.-Sicherung:** Begleitende diagnostische Maßnahmen
 - **Q.-Prüfung:** Abschließende analytische 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

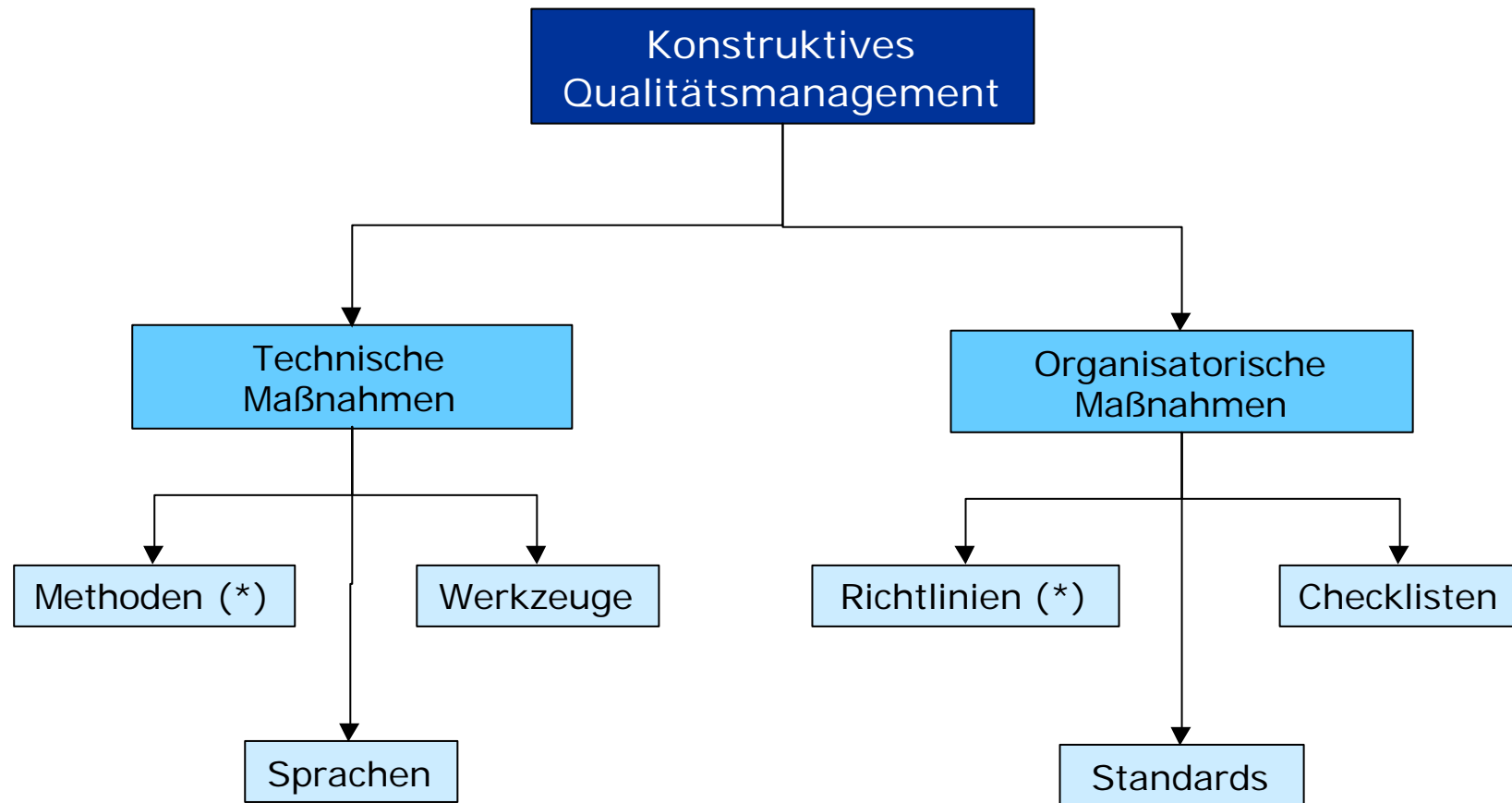
- Gerichtet auf den Erstellungsprozess der Software selbst
 - eher für Firmen, die anwenderspezifische Spezialsoftware herstellen, mit variierenden Q.-Anforderungen und dynamischem Qualitätsoptimum
 - Ziel ist die Herausbildung eines Qualitätsbewusstseins bei den Mitarbeitern
- **Grundansatz:** Qualität durch den Erstellungsprozess selbst
- **Faktoren:** Planbarkeit, Effizienz (im Kosten/Nutzen-Sinn), Produktqualität
- **Kontext:** Prozesszertifizierung, Prozessverbesserung

Konstruktive Maßnahmen

- Konstruktive Maßnahmen sorgen von vornherein für ein gewisses Maß an Qualität durch 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 %

2. Qualitätsmanagement

2. Konstruktive und analytische Maßnahmen



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

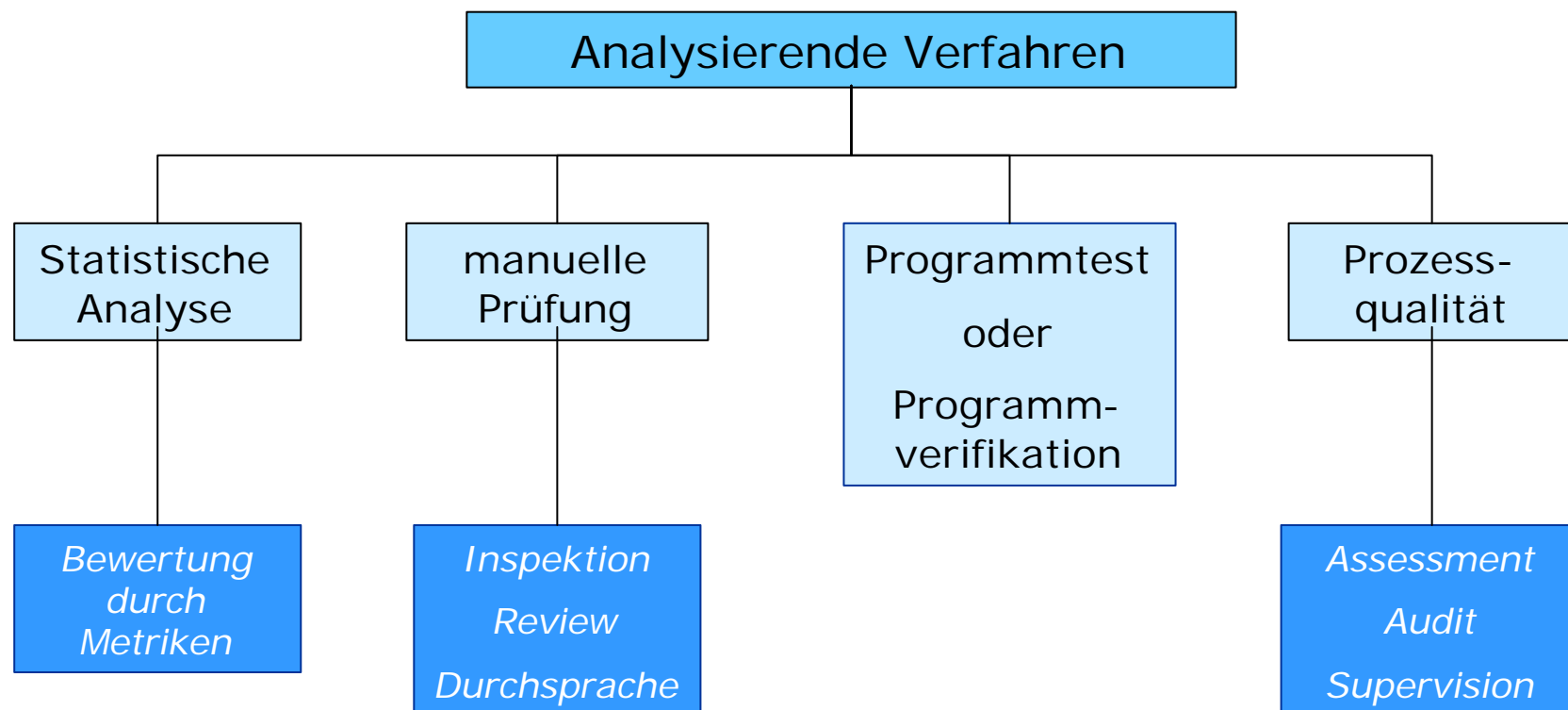
- Ziel: Produkteigenschaften a-priori festlegen
- Technik:
 - Vorgabe von Checklisten, Schablonen
 - Überprüfung der Richtlinien
- Beispiele:
 - Strukturierung der Analyse durch SCR-Tabellen
 - Anwendung von Design Pattern
 - Einsatz von Coding Standards
- Vorteile:
 - Erfahrungen mit Richtlinien werden 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 und konstruktive QM-Maßnahmen beeinflussen sich gegenseitig: Vorausschauende konstruktive Planung erspart analytischen Aufwand

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

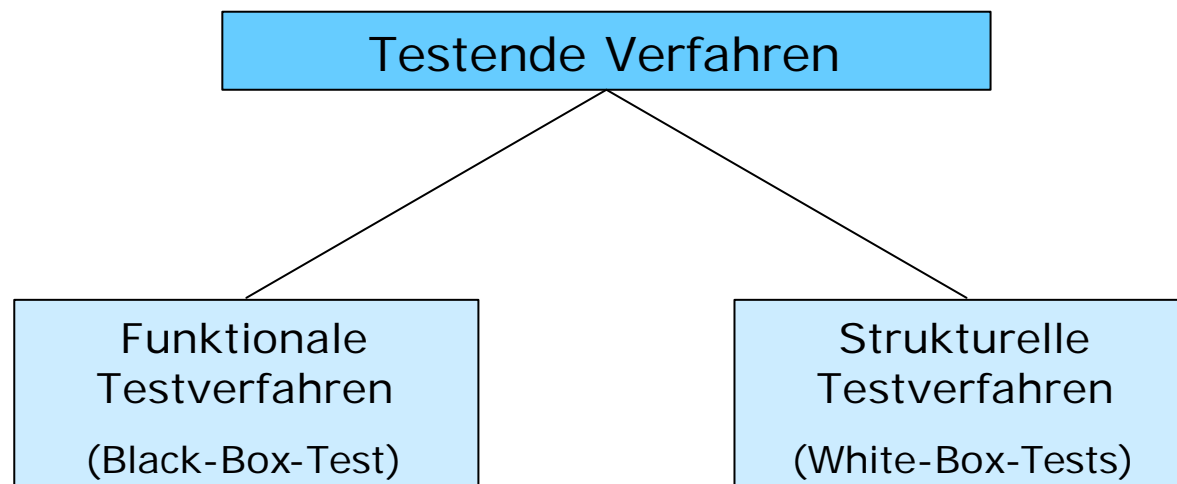
2. Qualitätsmanagement

2. Konstruktive und analytische Maßnahmen

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.



- **Qualitätsplanung:** Festlegung von Qualitätsanforderungen an den Prozess und an das Produkt in überprüfbarer Form.
- **Qualitätslenkung und -sicherung:** 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

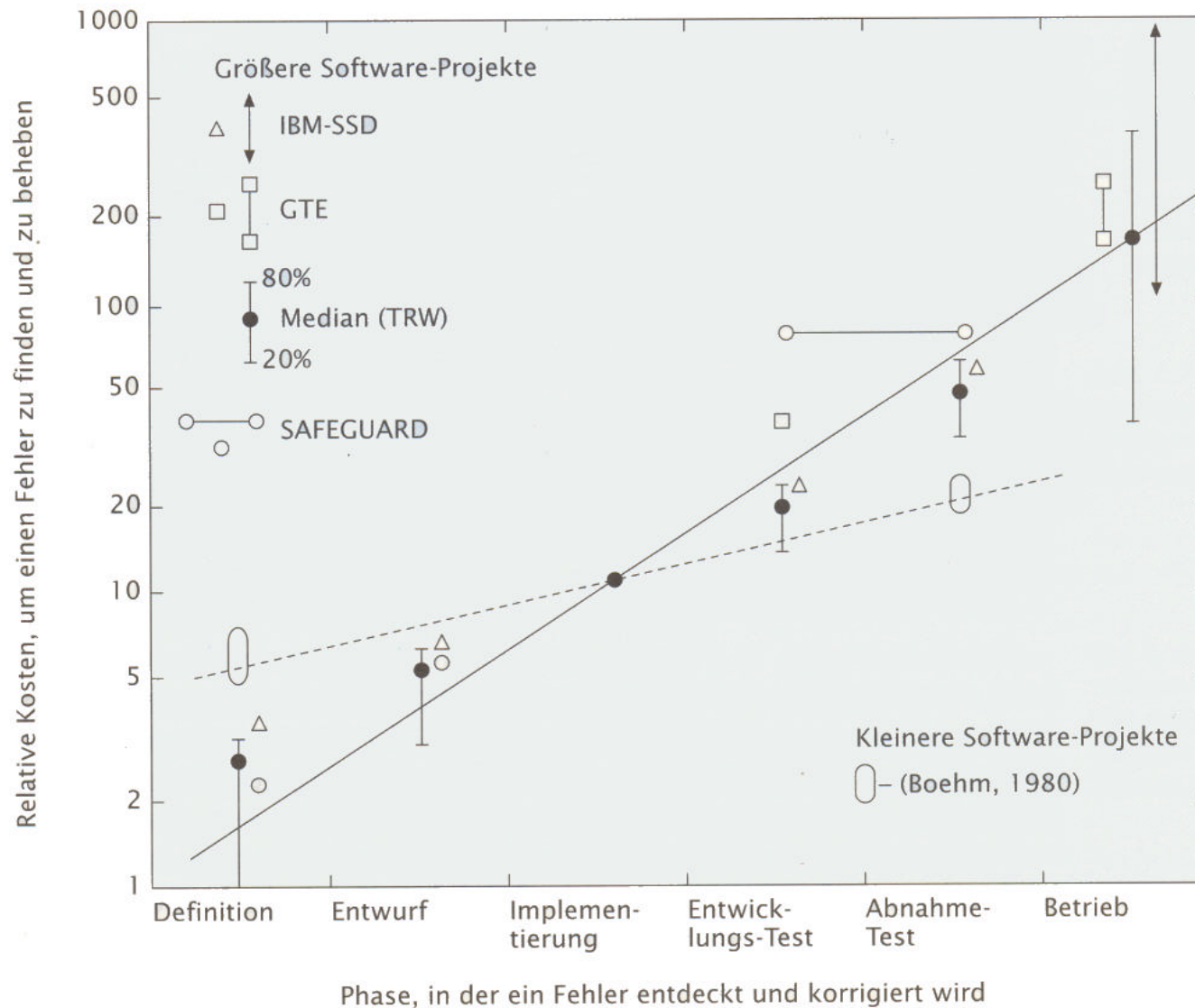
- „Fehler die nicht gemacht werden, brauchen auch nicht behoben werden“ ist das Ziel, das von der maximalen konstruktiven Qualitätssicherung verfolgt wird.
- Reduzierung der analytischen Maßnahmen über Einschränkung der Variationsbreite durch vorausschauende konstruktive Maßnahmen.
- **Vorteile:**
 -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

Prinzip der frühzeitigen Fehlerentdeckung und –behebung

- Fehler ist:
 - Abweichung von den Anforderungen des Auftraggebers
 - Inkonsistenz in den Anforderungen
- Ziel ist es, Fehler gar nicht erst zu machen (konstr. Maßnahmen) oder zum frühestmöglichen Zeitpunkt zu erkennen und zu beheben
- **Vorteile:**
 - Fehler in späteren Phasen werden vermieden
 - Kosten werden reduziert
 - mit höherer Wahrscheinlichkeit werden Fehler richtig korrigiert
 - die Fehlerfortpflanzung wird reduziert
- Folgerung: Viel Aufmerksamkeit den frühen Projektphasen

2. Qualitätsmanagement

Prinzip der frühzeitigen Fehlerentdeckung

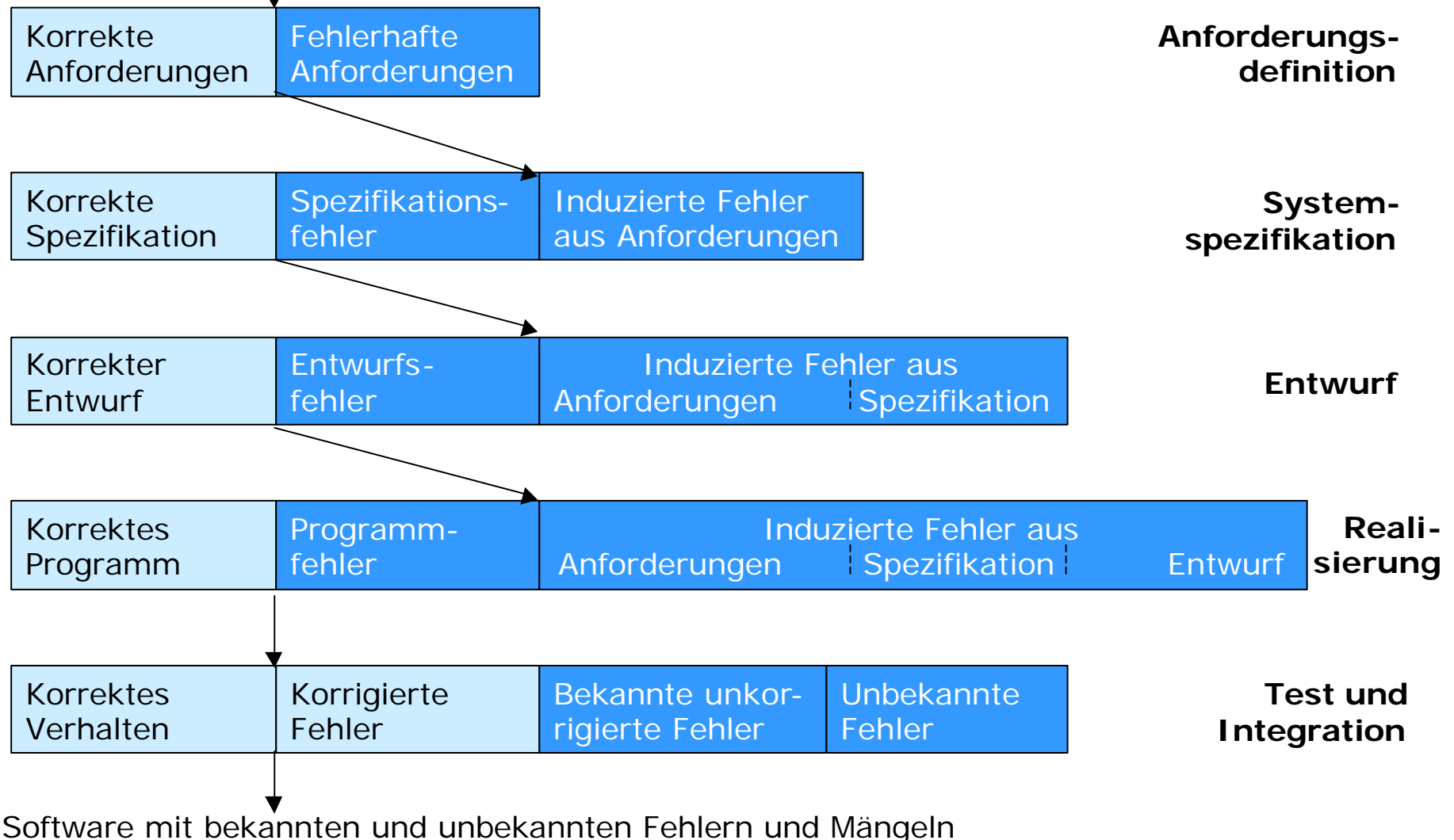


Kosten einer verzögerten Fehlerentdeckung [Boehm 76]

2. Qualitätsmanagement

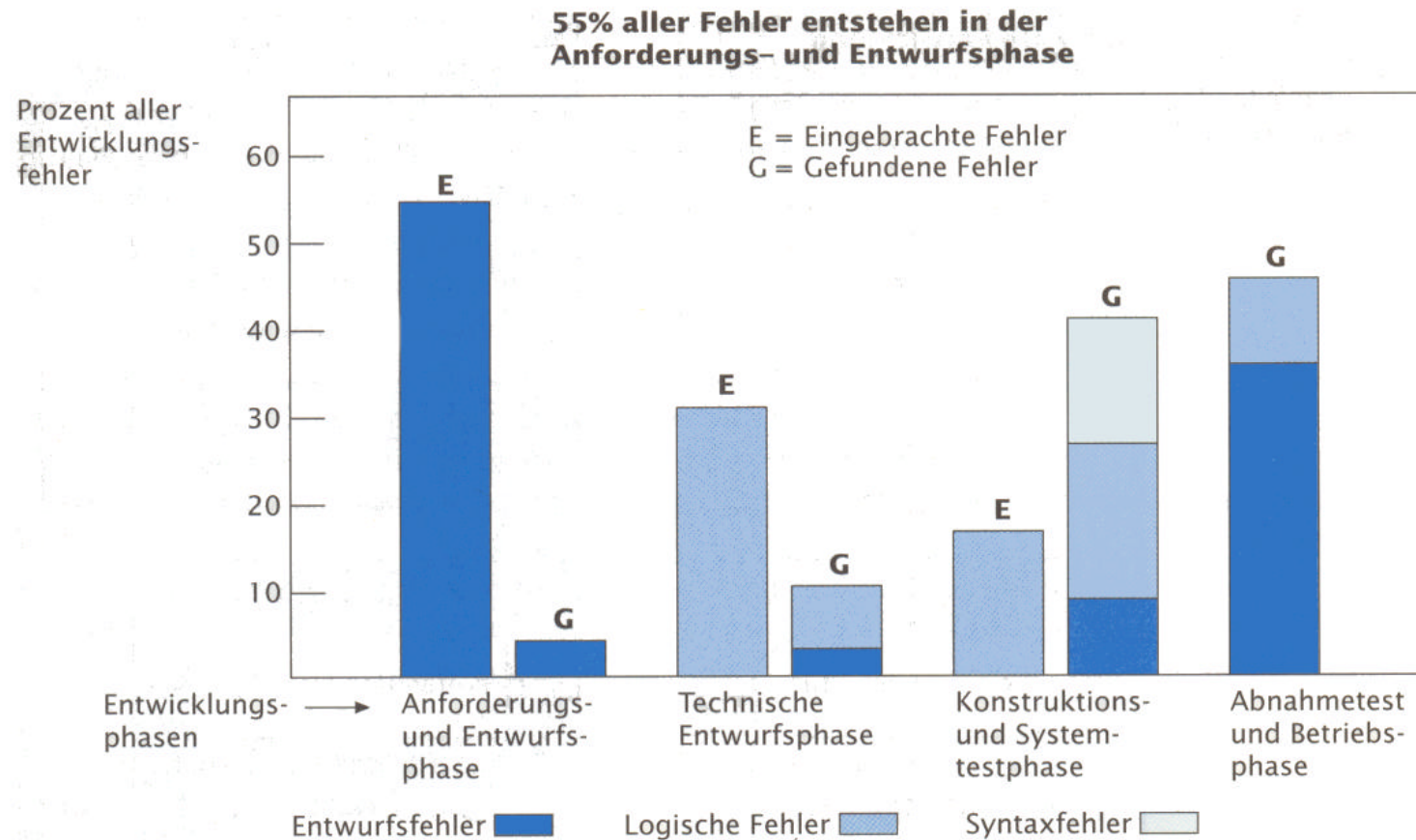
Folgen später Fehlerentdeckung

Ideen, Wünsche und Bedürfnisse



2. Qualitätsmanagement

Fehlerbeseitigungskosten



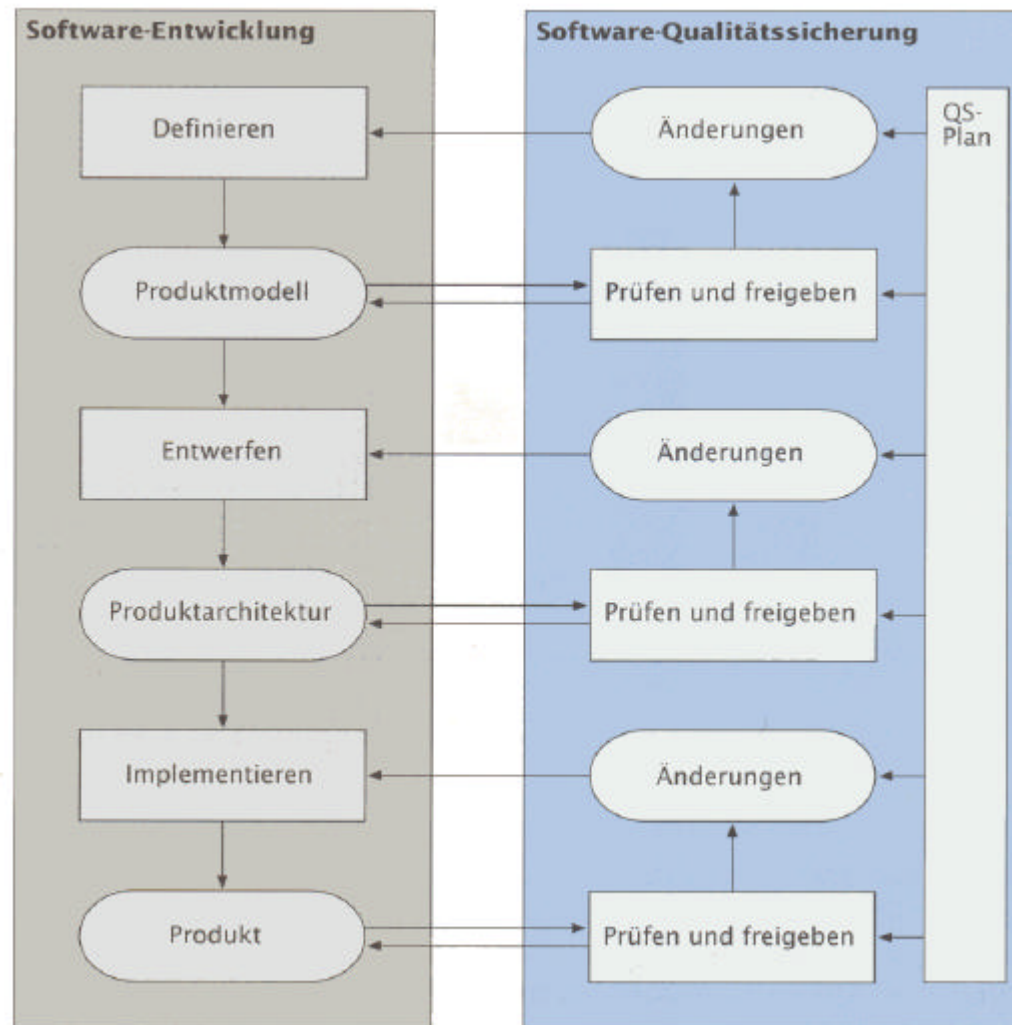
Fehlerbeseitigungskosten (abgeleitet von Alberts 1985)

Prinzip der entwicklungsbegleitenden, integrierten Qualitätssicherung

- Um das Prinzip der frühzeitigen Fehlerentdeckung zu realisieren, ist Softwareentwicklung begleitende und in den Entwicklungsprozess integrierte Qualitätssicherung nötig.
- **Vorteile:**
 - Einbettung der Qualitätssicherung in das organisatorische Ablaufmodell der Software-Entwicklung
 - Qualitätssicherung findet zu dem Zeitpunkt statt, zu dem sie im Entwicklungsprozess angebracht ist
 - Qualitätssicherung wird nicht als Fremdkörper empfunden, sondern gehört per se zur Software-Erstellung

2. Qualitätsmanagement

Prinzip der integrierten Qualitätssicherung



Prinzip der unabhängigen Qualitätssicherung

„... Testing is a **destructive** process, even a sadistic process ...“ [Myers 79]

- Der Entwickler eines Produkts ist am schlechtesten geeignet, um durch Anwendung analytischer QS-Maßnahmen die Ergebnisse seiner Tätigkeit zu betrachten.
- Entwickler darf aber seine Aufgaben im Bereich QS nicht vernachlässigen.
- Zwei organisatorische Alternativen:
 - Qualitätssicherung als organisatorisch unabhängiger Teil von der Gestaltung
 - Qualitätssicherung als Teil der Entwicklung

Prinzip der unabhängigen Qualitätssicherung - Qualitätssicherung als organisatorisch unabhängiger Teil

- **Vorteile:**

- Entwicklung übt keinen „Druck“ auf die Qualitätssicherung aus
- Neutralität
- klare Budgetaufteilung
- Betonung der Qualitätssicherung

- **Nachteile:**

- Gefahr der Isolierung der Qualitätssicherung von der Entwicklung
- gleichmäßige Personalauslastung ist unter Umständen sicherzustellen

Prinzip der unabhängigen Qualitätssicherung - Qualitätssicherung als Teil der Entwicklung

- **Vorteile:**

- flexiblere Einsetzung des Personals
- Qualitätssicherung „bekommt alles mit“
- Erleichterung der Teamarbeit
- vertrauensvolle Zusammenarbeit

- **Nachteile:**

- Entwicklungsmanagement kann „Druck“ auf die Qualitätssicherung ausüben
- Budgetmittel können zugunsten der Entwicklung umverteilt werden

Prinzip der unabhängigen Qualitätssicherung

Personalalternativen

- 3 Möglichkeiten für die Personalausstattung der Qualitätssicherung:
 - Personal arbeitet nur in der Qualitätssicherung
Ermöglicht die Einstellung von Mitarbeitern mit einem hohen Spezialisierungsgrad, aber diese bekommen nie Erfahrung mit der Entwicklung von Software.
 - Jeder Mitarbeiter rotiert in festgelegten Abständen zwischen der Qualitätssicherung und der Entwicklung
Ermöglicht einen systematischen Wissenstransfer, aber Mitarbeiter müssen Tätigkeiten durchführen, zu denen sie keine „Lust“ haben.
 - Jeder Mitarbeiter arbeitet sowohl an der Qualitätssicherung als auch an der Entwicklung (in der Praxis üblich)
Ermöglicht einen flexiblen Personaleinsatz, aber die Vermischung der Entwicklung und Qualitätssicherung könnte dazu führen, dass keine dieser Arbeiten richtig durchgeführt werden.

Prinzip der unabhängigen Qualitätssicherung - Verhältnis zwischen Entwicklung und Qualitätssicherung

- Die Entwicklung erstellt Produkte und die Qualitätssicherung ist für die Überprüfung zuständig.
 - Entwickler kann sich auf die konstruktiven Aspekte konzentrieren
 - Wird aber nicht zu Sorgfalt angehalten
 - Realisierung im Ansatz des Pair Programming
- Die Entwicklung ist für einen definierten Qualitätszustand ihrer Produkte selbst zuständig. Erst danach setzt die externe Q.-Sicherung ein.
 - klar definierte, transparente Verantwortlichkeiten
 - Eigenverantwortlichkeit der Entwickler
 - erfordert messbare Qualitätsstufen und Nachweis, dass sie erreicht wurden

Prinzip der unabhängigen QS im Lichte der quantitativen QS

- Wird das Prinzip der quantitativen Qualitätssicherung befolgt, so werden viele Prüfparameter instrumentell erfasst.
- Personal der Qualitätssicherung kann sich auf Fragen der Interpretation der Messwerte konzentrieren.
- In der Qualitätssicherung sind folgende zusätzliche Maßnahmen erforderlich:
 - Sammlung von Daten,
 - Validierung dieser Daten und
 - Einrichten einer quantitativ orientierten Datenbank.
- Vorteile einer unabhängigen Qualitätssicherung sind:
 - objektive, unabhängige Qualitätssicherung,
 - heilsame Wirkung auf Entwicklung und
 - Qualitätsvergleiche werden möglich.