Software-Qualitätsmanagement

Kernfach Angewandte Informatik

Sommersemester 2004

Prof. Dr. Hans-Gert Gräbe

Gliederung

1. Qualitätsmanagement und Qualitätssicherung
2. Prinzipien der Software-Qualitätssicherung
3. 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]

Qualitätsmanagement

- Planung: Vorbereitende Maßnahmen
- Lenkung: Konstruktive Maßnahmen
- Sicherung: Analytische Maßnahmen
- Verbesserung: Strukturelle Maßnahmen

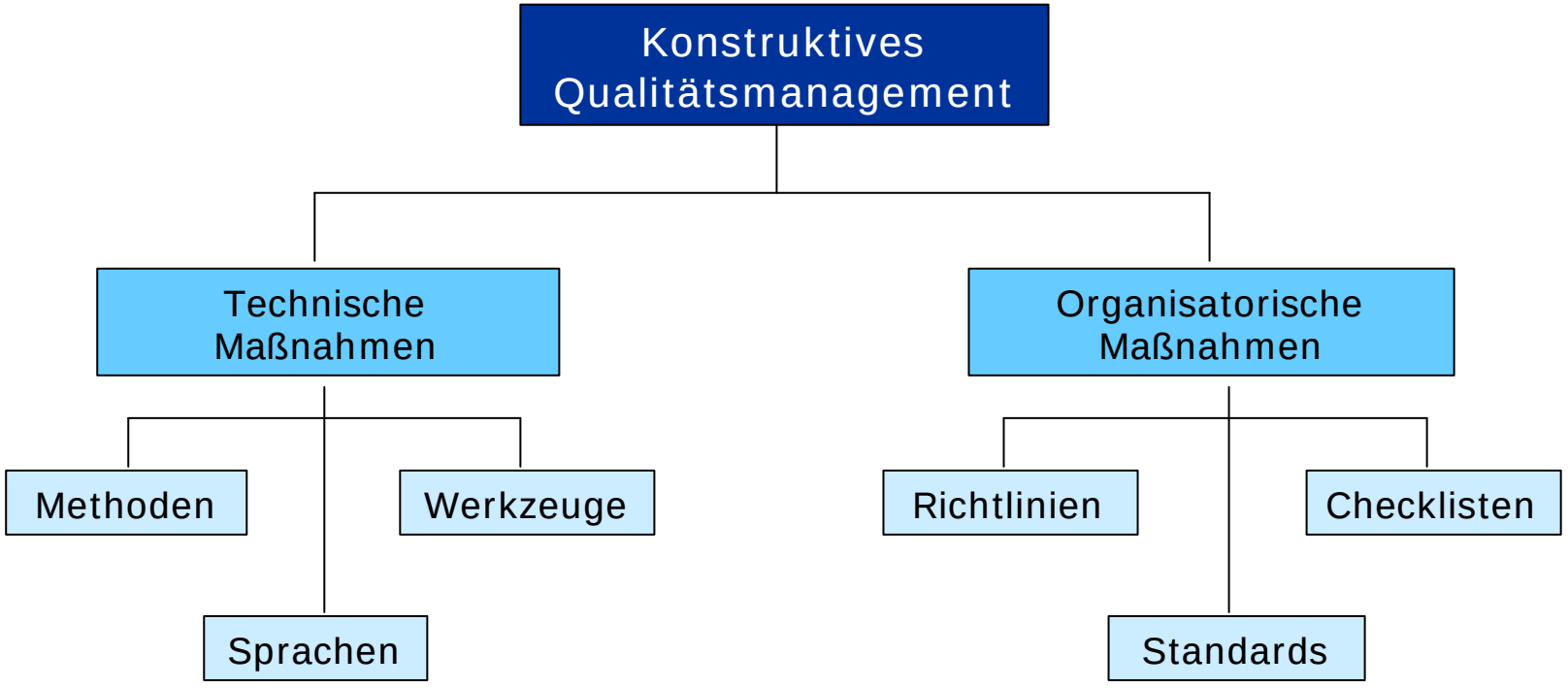
- **Qualitätsmanagement** (QM) = managementbezogene Aktivitäten
  - Oberbegriff, eher prozessorientiert
- **Qualitätssicherung** (QS) = technikorientierte Aktivitäten
  - Teil des gesamten QM-Systems, eher produkt- oder ergebnis-orientiert.
- QM-Systeme aus traditionellen Ingenieurbereichen, die auf Produktqualität ausgerichtet sind, lassen sich nicht auf SW übertragbar
  - Software ist in diesem Sinn kein Produkt, sondern vergleichbar mit einem Produkt-Prototyp

- **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
  - Qualität kann durch Zertifikat (Prüfung durch unabhängige Seite) bestätigt werden

- **Prozessorientiertes Q.-Management**
  - Gerichtet auf den Erstellungsprozess der Software selbst
    - eher für Firmen, die anwenderspezifische Spezialsoftware herstellen, mit variierenden Q.-Anforderungen und dynamischem Qualitätsoptimum
  - Mitarbeiter müssen ein Qualitätsbewusstsein entwickeln

Konstruktive QM-Maßnahmen

- **Konstruktive QM-Maßnahmen**
  - sorgen von vornherein für ein gewisses Maß an Qualität



Analytische QM-Maßnahmen

- **Analytische QM-Maßnahmen**
  - diagnostische Maßnahmen, bringen keine Qualität per se
  - sind zur Messung der Qualität der End- bzw. Zwischenprodukte

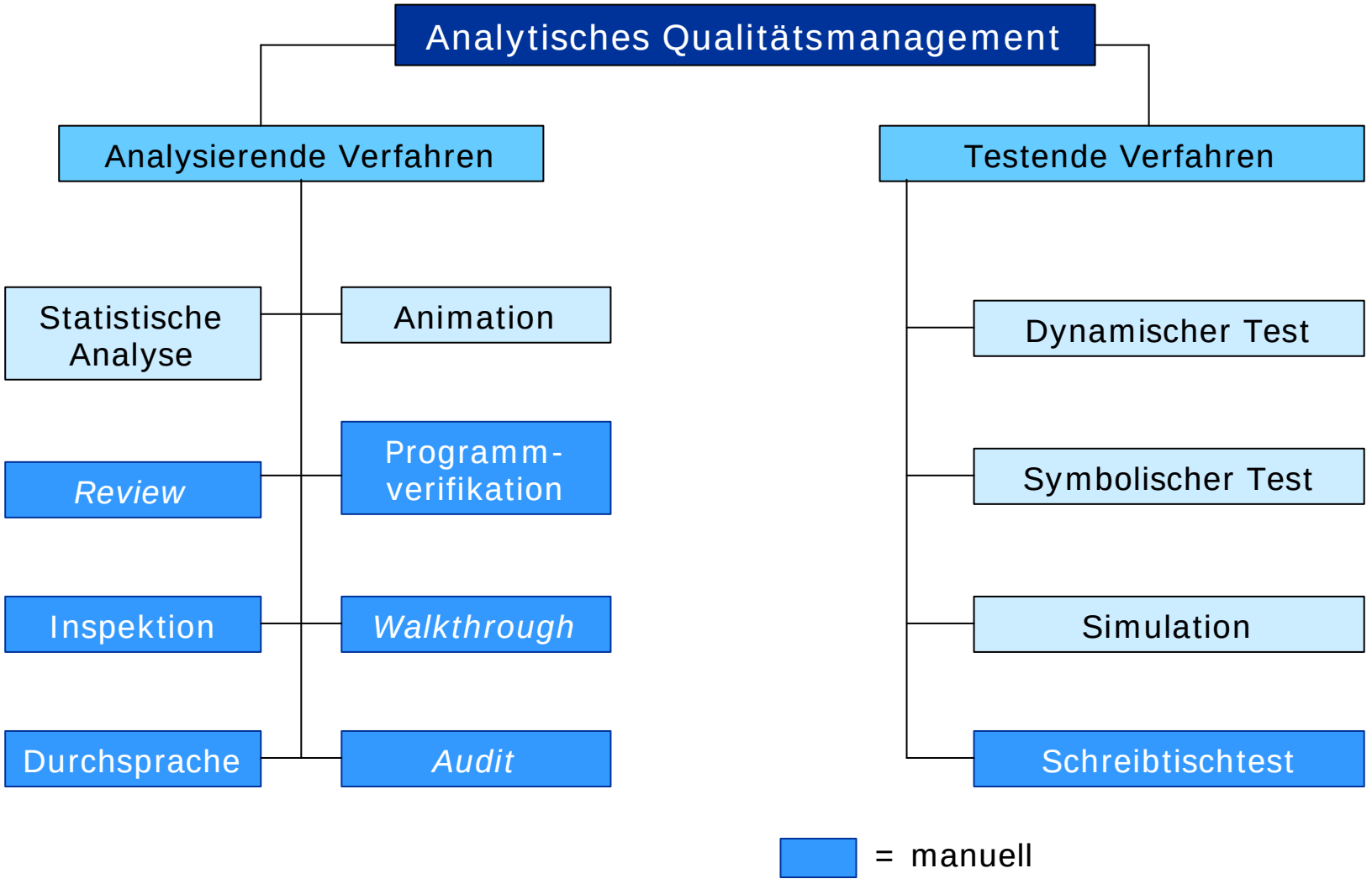
- Gliederung nach verschiedenen Gesichtspunkten:
  - Bezug der Prüfung (Produkt oder Prozess)
  - Automatisierungsgrad der Prüfung (manuell / mit Werkzeug)
  - Nachvollziehbarkeit der Prüfung (Selbstprüfung / Nachweis)
  - Einsatzbereich der Prüfung (in welcher Phase des SW-Zyklus)

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

- **Testende Verfahren** führen den Prüfling mit Eingaben aus.

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

Analytisches Qualitätsmanagement



Aktivitäten im Qualitätsmanagement

- **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
  - Mängel- und Fehleranalyse (Verbesserung der Prozessqualität)

Qualitätssicherungsplan

- Ergebnisse der Qualitätsplanung werden in einem **Qualitätssicherungsplan** dokumentiert.
  - **Was muss gesichert werden?**
    Identifizierung der relevanten Qualitätsmerkmale, ihre relative Bedeutung und ihre Quantifizierung in Form von Metriken
  - **Wann muss gesichert werden?**
    Festlegung der Zeitpunkte für die den gesamten Entwicklungsprozess begleitende Datenerfassung
  - **Wie muss gesichert werden?**
    Auswahl der zur Datenerfassung und Qualitätsprüfung geeigneten Techniken und Methoden
  - **Von wem muss gesichert werden?**
    Festlegung der Verantwortlichkeiten für die Qualitätsprüfung und -lenkung.

- Gliederungsschema für Qualitätssicherungspläne vom IEEE-Standard 730/84 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 Qualitäts-anforderungen 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 vorausblickende konstruktive Maßnahmen.

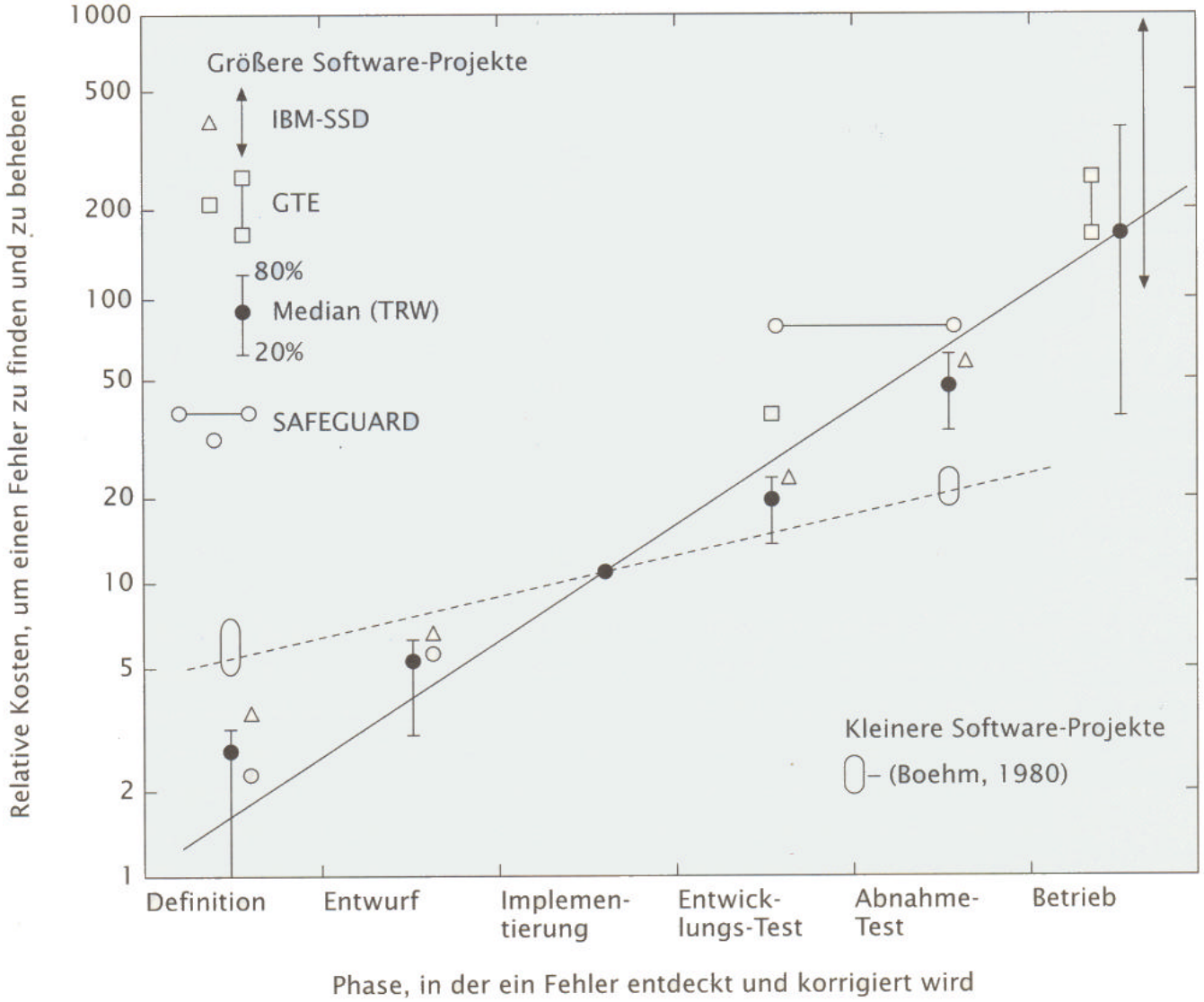
- **Vorteile:**
  - direkte Verbesserung der Produktivität
  - Reduktion analytischer Maßnahmen
  - Voraussetzung für analytische Maßnahmen
  - Vermeidung von Fehlern

Prinzip der frühzeitigen Fehlerentdeckung

Prinzip der frühzeitigen Fehlerentdeckung und –behebung

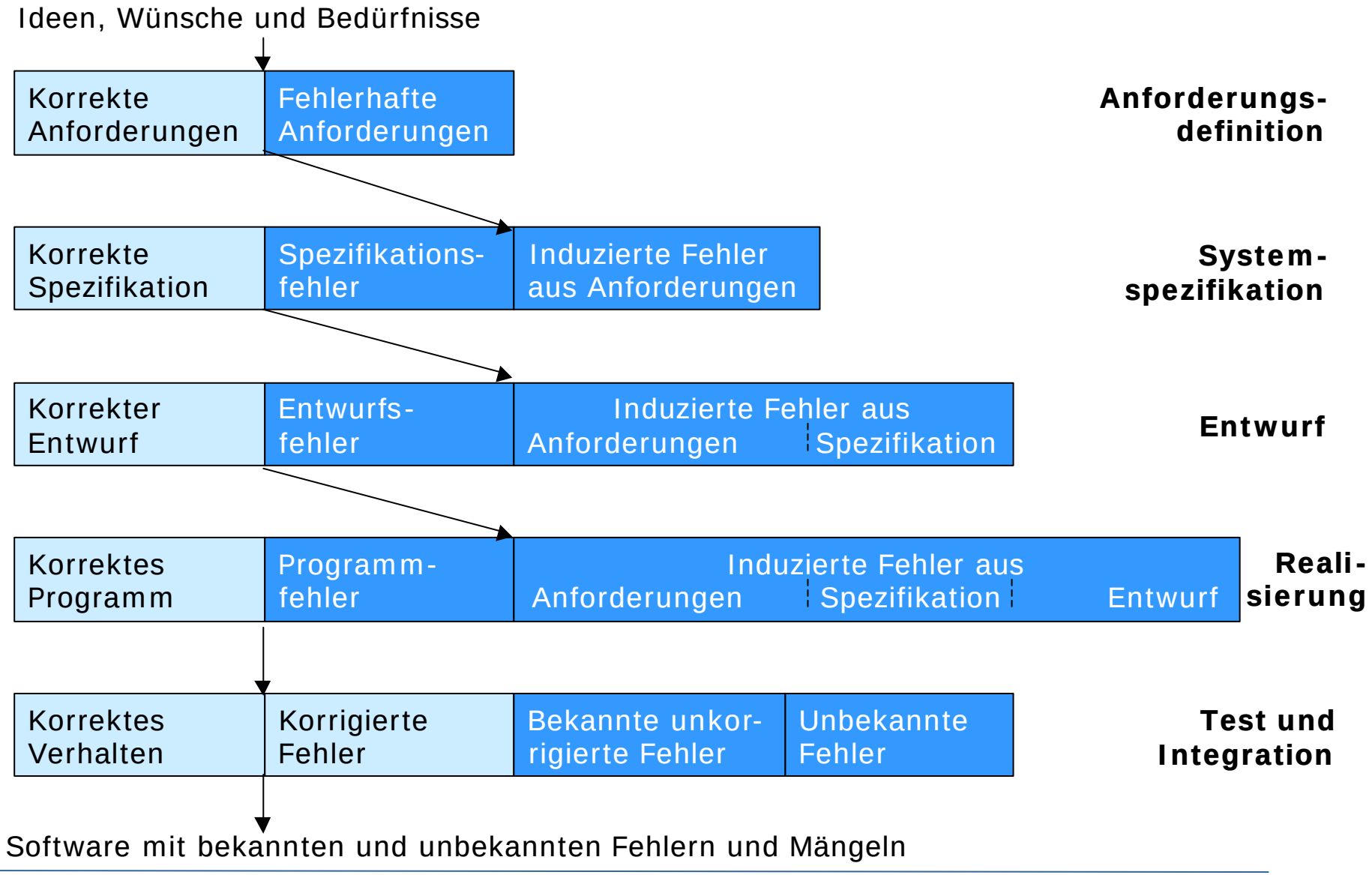
- Fehler ist:
  - Abweichung von den Anforderungen des Auftraggebers
  - Inkonsistenz in den Anforderungen
- Ziel ist es, Fehler gar nicht erst zu machen (konstr. Maßnahmen) oder zum frühestmöglichen Zeitpunkt zu erkennen und zu beheben
- **Vorteile:**
  - Fehler in späteren Phasen werden vermieden
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

Prinzip der frühzeitigen Fehlerentdeckung

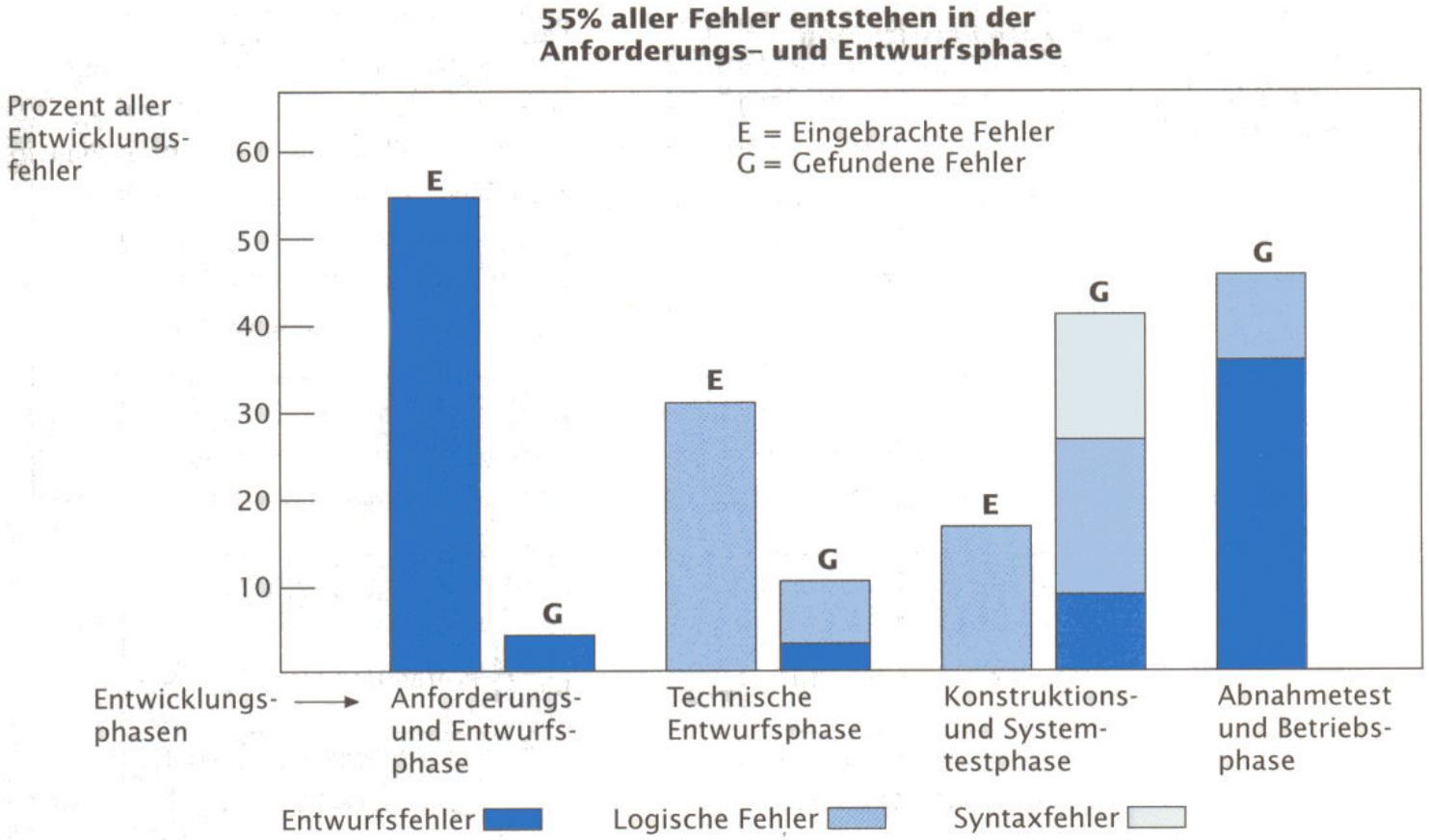


Kosten einer verzögerten Fehlerentdeckung [Boehm 76]

Folgen später Fehlerentdeckung



Fehlerbeseitigungskosten



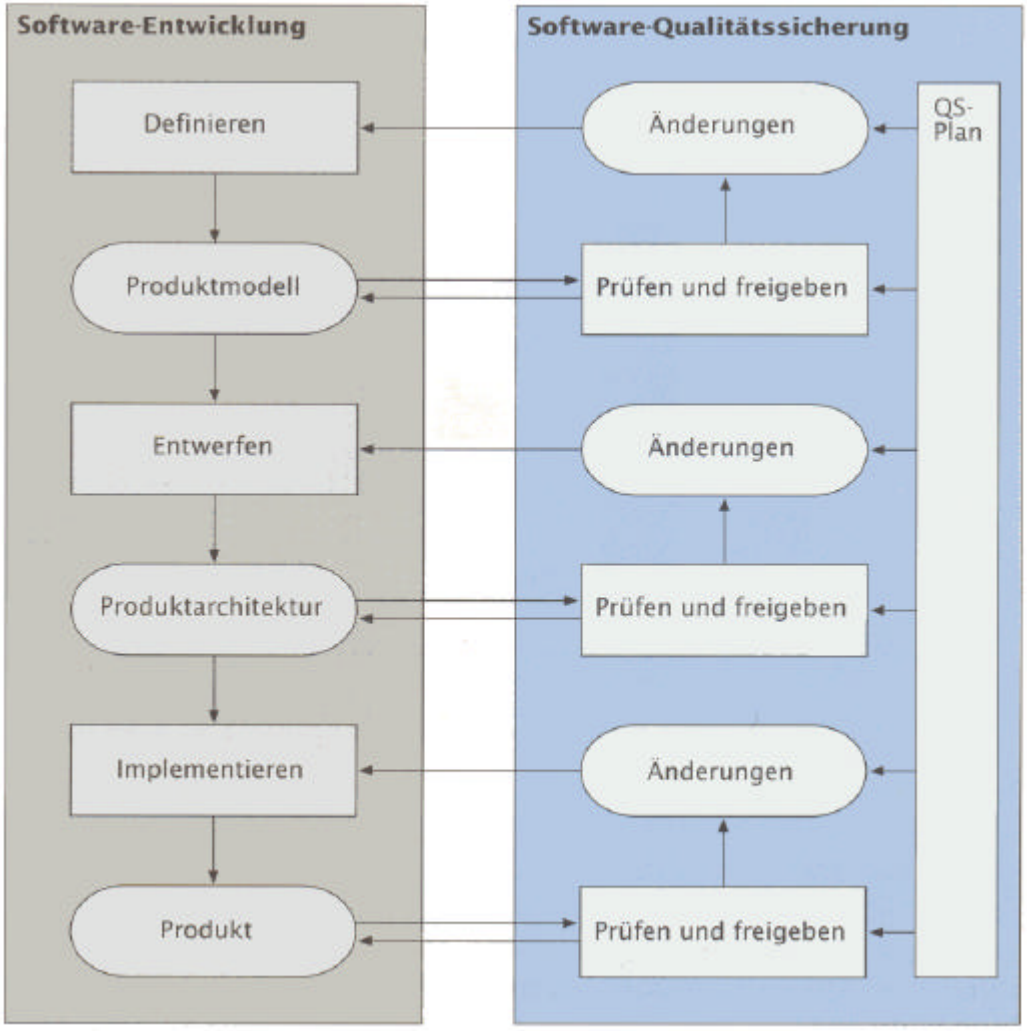
Fehlerbeseitigungskosten (abgeleitet von Alberts 1985)

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

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 QM-Maßnahmen die Ergebnisse seiner Tätigkeit zu betrachten.
- Entwickler darf aber seine Aufgaben im Bereich Qualitätssicherung nicht vernachlässigen.
- Zwei organisatorische Alternativen:
  - Qualitätssicherung als organisatorisch unabhängiger Teil von der Gestaltung
  - Qualitätssicherung als Teil der Entwicklung

Prinzip der unabhängigen Qualitätssicherung

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

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 den 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

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 Qualitätssicherung

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.

Beispiel: Qualitätssicherung im V-Modell

V-Modell

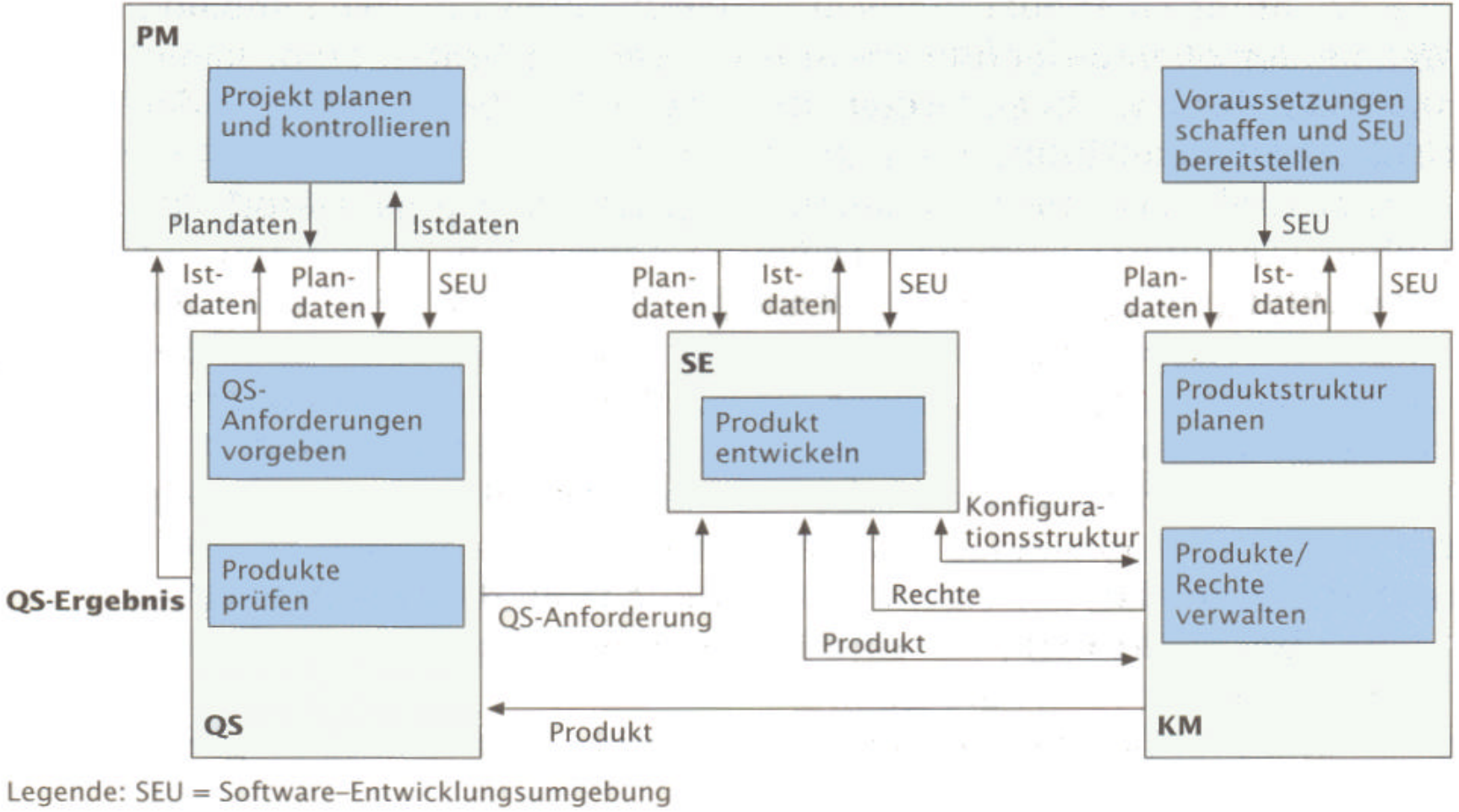
- Einsatz eines **Qualitätssicherungssystems** zur Durchführung des Qualitätsmanagements.

- QS-System besteht aus für die Verwirklichung des Qualitätsmanagements bzw. der Qualitätssicherung erforderlichen Organisationsstruktur, Verfahren, Prozessen und Mitteln.

- **V-Modell** ist ein Vorgehensmodell. Es gliedert sich in 4 Submodelle:
  - System-Entwicklung (SE)
  - Qualitätssicherung (QS)
  - Konfigurationsmanagement (KM) und
  - Projektmanagement (PM)

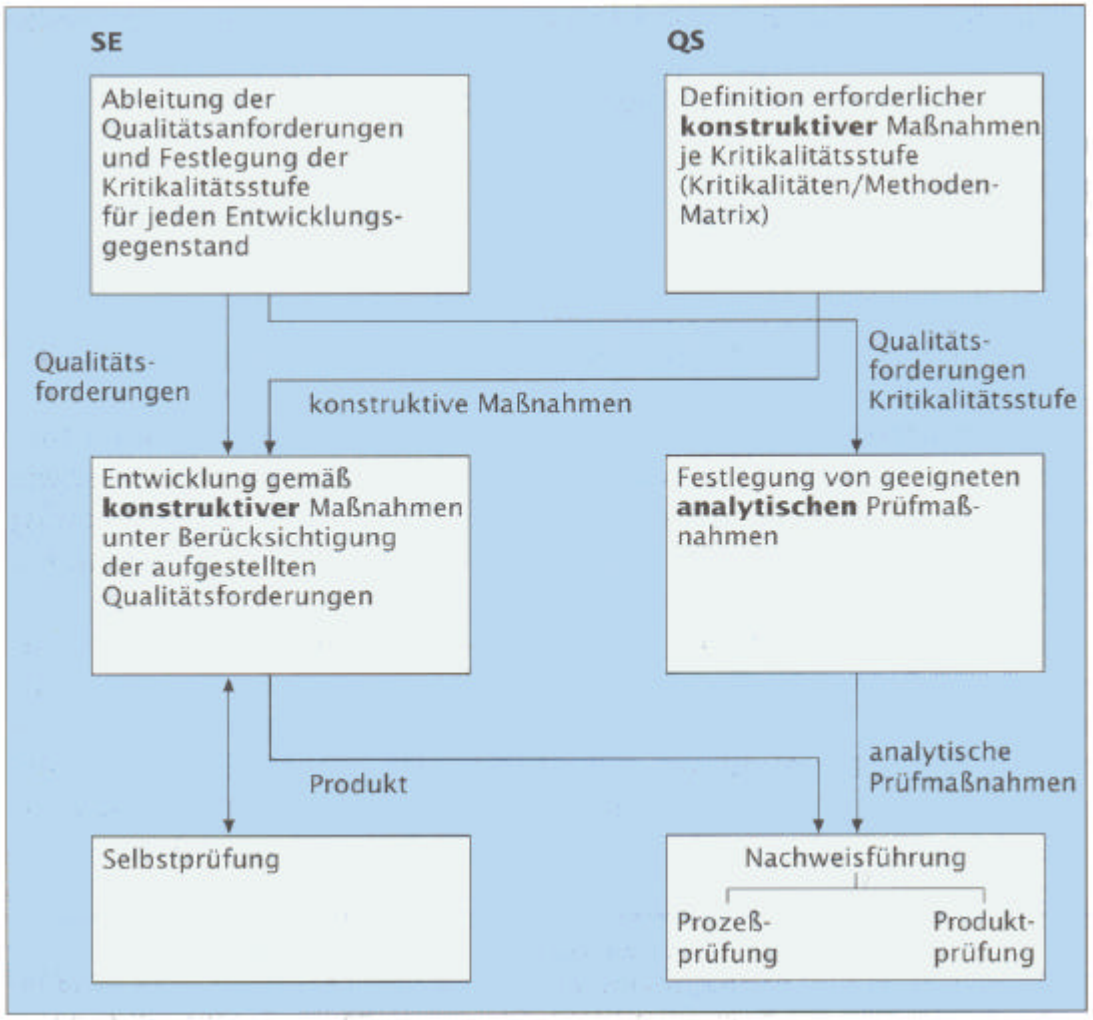
- Für uns ist der Teil **Qualitätssicherung wichtig.**

Zusammenwirken der vier Submodelle



[V-Modell 97]

Arbeitsteilung zwischen SE und QS

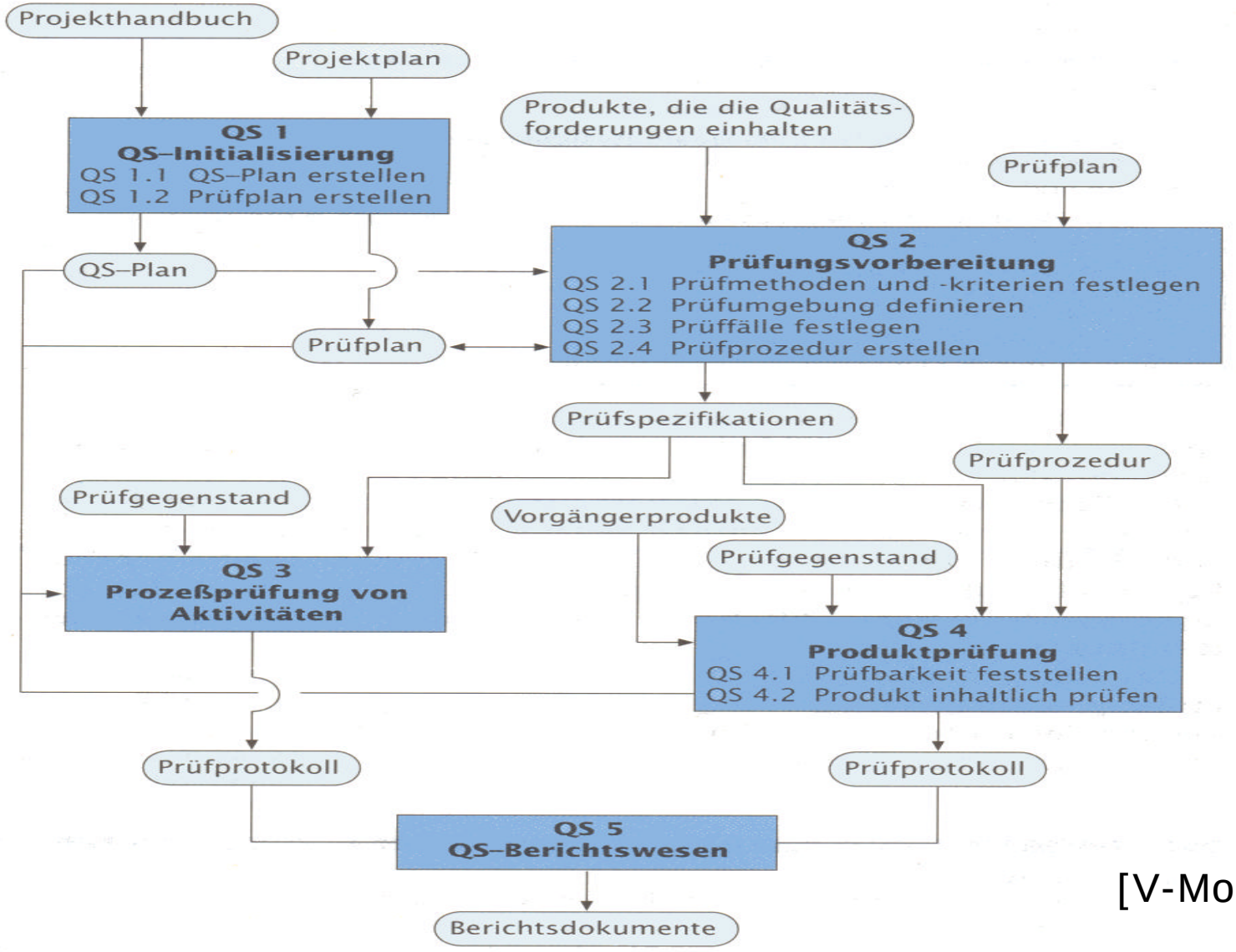


[V-Modell 97]

Submodell QS im V-Modell

- Qualitätsstufen werden über **Kritikalität** definiert, d.h. welche Bedeutung das Fehlverhalten einer Einheit hat.
  - ist projektspezifisch und hängt vom Einsatzzweck ab
- Im QS-Teil sind folgende Hauptaktivitäten durchzuführen:
  - Planungsaktivitäten, Prüfaktivitäten und Lenkungsaktivitäten
- und werden folgende Produkte erstellt:
  - QS-Plan, Prüfplan, Prüfspezifikation, Prüfprozedur, Prüfprotokoll.

Funktionsüberblick Submodell QS



[V-Modell 97]

Zusammenfassung

- **Qualitätsmanagement** umfasst
  - Q.-**Planung**
    - Festlegung von Q.-Anforderungen in überprüfbarer Form
  - Q.-**Lenkung** und -**Sicherung**
    - Überwachung und Steuerung des Entwicklungsprozesses mit dem Ziel, die vorgegebenen Q.-Anforderungen zu erfüllen
    - Nachweisführung über Erfüllungsstand von Q.-Anforderungen
  - und Q.-**Prüfung**
    - Erfassung der Ist-Parameter der Q.-Indikatoren entsprechend der Q.-Planung sowie Kontrolle der Einhaltung der konstruktiven QM
- **konstruktive QM** sorgen dafür, dass das Produkt gewisse Eigenschaften a priori besitzt
- **analytische QM** messen das existierende Q.-Niveau und identifizieren Ausmaß und Ort von Defekten
- Alle Maßnahmen des **QM** werden in einem **QS-Plan** fixiert
- QS von Softwareprojekten sollte sich an den oben formulierten **sechs Grundprinzipien** orientieren.

Literatur

[Boehm 76] Boehm B.W., *Software-Engineering*, in: IEEE Transactions on Computers, Vol. C-25, Dec. 1976

[Mizumo 83] Mizumo Y., *Software Quality Improvement*, in: IEEE Computer, March 1983

[Myers 79] Myers G.J., *The Art of Software Testing*, New York: John Wiley & Sons 1979

[Rombach 93] Rombach H.D., *Software-Qualität und – Qualitätssicherung*, in: Informatik-Spektrum 1993

[Spillner, Liggesmeyer 94] Spillner A., Liggesmeyer P., *Software-Qualitätssicherung in der Praxis*, in: Informatik-Spektrum 1994

[V-Modell 97] *Entwicklungsstandard für IT-Systeme des Bundes, Vorgehensmodell*, Teil 1: Regelungsteil, Allgemeiner Umdruck Nr. 250/1, BWB IT 15, Koblenz, Juni 1997