

Teil II – DO

20. Qualitätssicherung

Prof. Dr. Uwe Aßmann
Lehrstuhl Softwaretechnologie
Fakultät Informatik
TU Dresden
Version 11-0-2,22.06.11

- 1) Fehler – Warum man QS braucht
- 2) Qualitätsbegriff
- 3) Konstruktives Qualitätsmanagement
- 4) Analytisches QM

- 1) Analyseverfahren
- 2) Testverfahren



Softwaremanagement, © Prof. Uwe Aßmann

Referenzierte Literatur

- ▶ [Wallmüller] Wallmüller, E.: Software-Qualitätssicherung in der Praxis; Hanser Verlag 1990 sowie 2. Auflage erschienen 2001
- ▶ [Trauboth] Trauboth; H.: SW-Qualitätssicherung; Oldenbourg Verlag 1996
- ▶ [Balzert2] Balzert, H. : Lehrbuch der SW-Technik; Bd 2 Spektrum- Verlag 2001, abgelöst durch:
- ▶ [BalzertSM] Balzert, H.: Lehrbuch der Softwaretechnik – Softwaremanagement Spektrum Verlag 2008
- ▶ American Society for Quality <http://www.asq.org/>



20.1 Fehler – Warum man Qualitätssicherung braucht



Softwaremanagement, © Prof. Uwe Alßmann

Fehleranzahl und -kosten



(Empirische Daten)

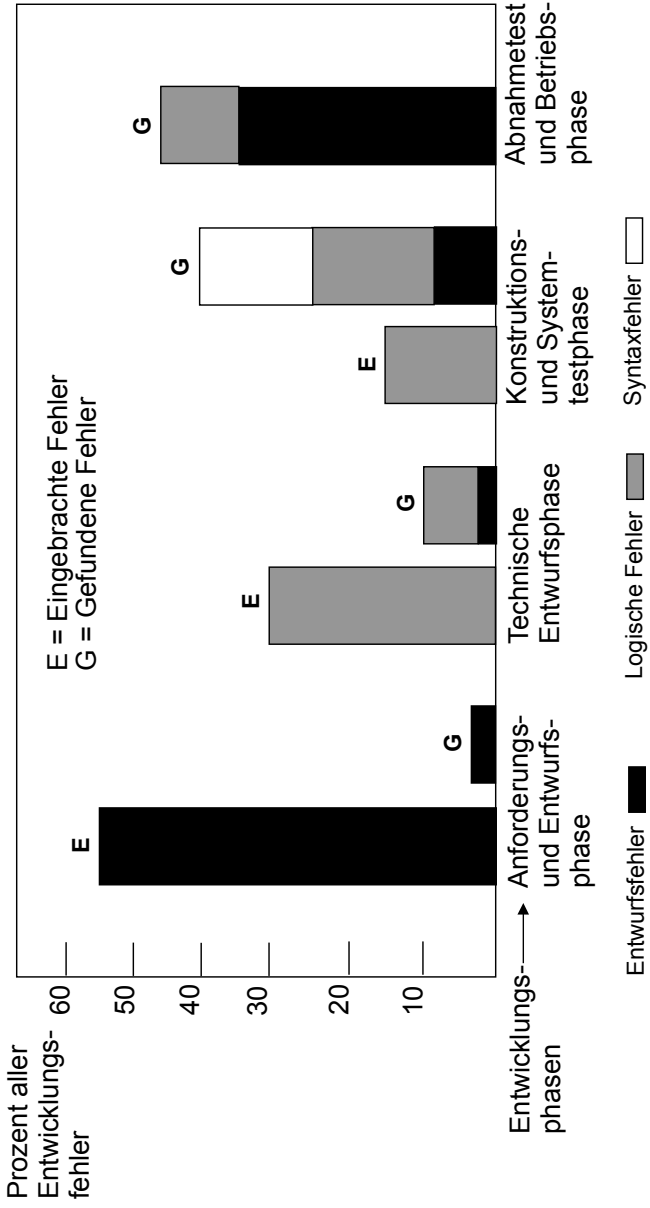
Relative Anzahl der entstandenen Fehler	10%	40%	50%				
Relative Anzahl der erkannten Fehler	3%	5%	7%	25%	50%	10%	
Kosten Pro Fehler-Korrektur (DM)	500	500	500	2.000	6.000	25.000	Feld
	Analyse	Entwurf	Codierung	Entwickler-test	Systemtest	Feld	

Quelle: Liggesmeyer u. a.: Qualitätssicherung software-basierter technischer Systeme; Informatik-Spektrum 21(1998) S. 249 - 258



Fehlerbeseitigungskosten

5



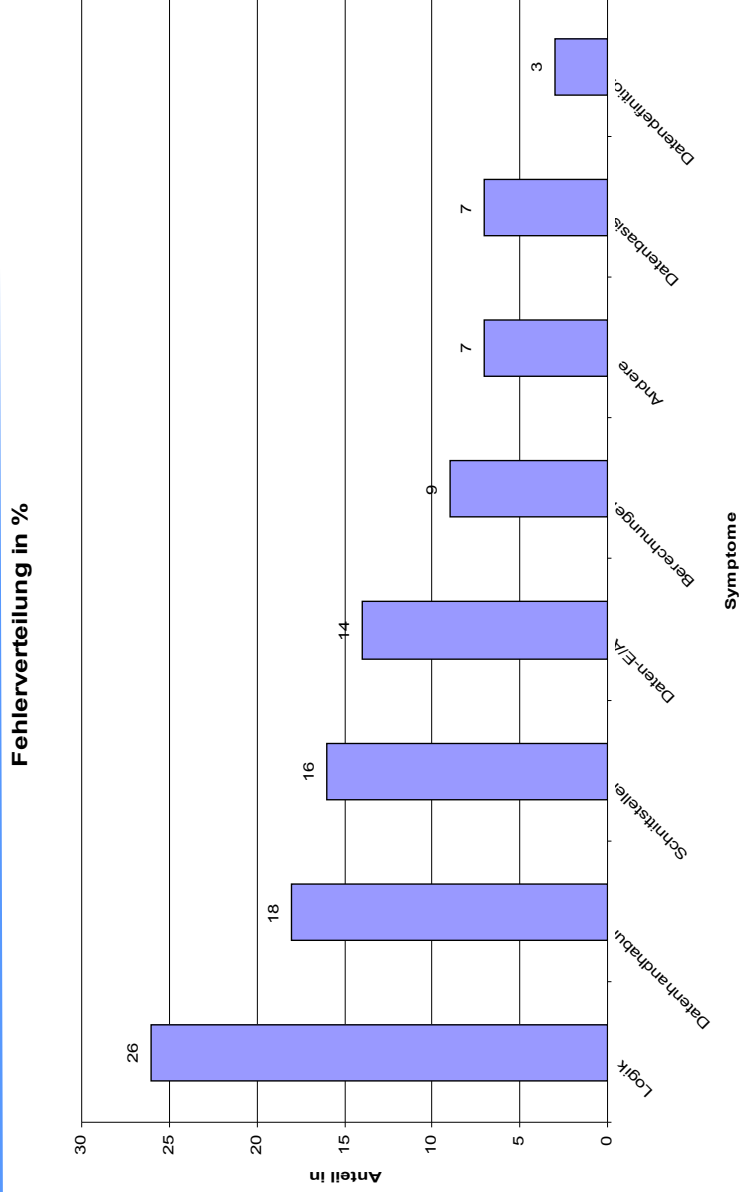
55% aller Fehler entstehen in der Anforderungs- und Entwurfsphase

Quelle: [Balzert, S. 487]



Fehlerverteilung

6



Quelle: [Trauboth]



Die acht Grundsätze des Qualitätsmanagements

1. Kundenorientierung

(Bedürfnisse erfüllen, übertreffen, vorwegnehmen)

2. Führung

(Leiten durch Vorbild, Beachtung von Interessengruppen, Entwickeln einer Vision)

3. Einbeziehung der Menschen

(Problemlösungskompetenz entwickeln, Initiative zu Verbesserungen)

4. Prozessorientierter Ansatz

(Tätigkeiten und Ressourcen als Prozess darstellen, effiziente Prozesse)

5. Systemorientierter Managementansatz

(Wechselwirkungen zwischen Einzelprozessen, Koordination von Zuständigkeiten)

6. Ständige Verbesserung

(„Wer aufhört besser zu werden, hat aufgehört gut zu sein“)

7. Sachlicher Ansatz zur Entscheidungsfindung

(Analysen, Mitarbeiter-Umfragen, Vorschläge)

8. Lieferantenbeziehungen zum gegenseitigen Nutzen

(transparente Kommunikation, Verständigung über gemeinsame Ziele)

Quelle: DIN EN ISO 9000:2000-01

Stand: Februar 2000 - .DQS

Experiment von Weinberg

5 Gruppen entwickeln ein Programm mit identischen funktionalen Anforderungen u. einer zusätzlichen, für alle Gruppen unterschiedlichen nicht-funktionalen Anforderung. Die erzielte Qualität wird auf einer Skala von 1 (sehr gut) bis 5 (schlecht) gemessen.

Ziel:

Optimiere ...	Erstellungsaufwand	Anzahl Anweisungen	Speicherbedarf	Klarheit d. Programms	Klarheit d. Ausgaben	Anzahl Anweisungen	Speicherbedarf	Klarheit des Programms	Klarheit der Ausgaben
Erstellungsaufwand	1	4	4	5	3	5	3	5	3
Anzahl Anweisungen	2-3	1	2	3	5	3	5	3	5
Speicherbedarf	5	2	1	4	3	4	4	4	4
Klarheit d. Programms	4	3	3	2	1	2	2	2	2
Klarheit d. Ausgaben	2-3	5	5	1	1	1	1	1	1

- Qualität muss quantifizierbar sein, damit Anforderungserfüllung gemessen werden kann
- auch nicht funktionale Anforderungen sind erreichbar

20.2 Qualitätsbegriff

9



Softwaremanagement, © Prof. Uwe Alßmann

Begriff

10

Qualität ist die Gesamtheit von Eigenschaften und Merkmale eines Produkts oder einer Tätigkeit, die sich auf deren Eignung zur Erfüllung gegebener Erfordernisse bezieht. [DIN 55350, Teil 11]

Qualität des Produkts
des Entwicklungsprozesses

Unterteilung:

- **Merkmale**
- **Teilmerkmale (Kriterien)**
- **Indikatoren (Metriken)**

Bsp.: **Benutzbarkeit**

Erlernbarkeit **Bedienbarkeit**

Zugänglichkeit **Selbsterklärungsfähigkeit** ...

Quelle: [BalzertSM]

Quality is fitness for use



Merkmalsstufung nach DIN ISO 9126

11

Merkmal

- **Funktionalität**
- **Zuverlässigkeit**
- **Benutzbarkeit**
- **Effizienz**
- **Änderbarkeit**
- **Übertragbarkeit**

Teilmerkmale

Richtigkeit (Korrektheit)
 Angemessenheit
 Interoperabilität
 Ordnungsmäßigkeit (Normen, Bestimmungen)
 Sicherheit
 Reife
 Fehlertoleranz
 Wiederherstellbarkeit
 Verständlichkeit
 Erlernbarkeit
 Bedienbarkeit
 Zeitverhalten
 Verbrauchsverhalten
 Analysierbarkeit
 Modifizierbarkeit
 Prüfbarkeit
 Stabilität
 Anpassbarkeit
 Installierbarkeit
 Austauschbarkeit,
 Konformität (gegenüber Normen)

Merkmalsbeschreibung

Fähigkeit des Systems die geforderten Anforderungen(RE) zu erfüllen
 Einhaltung eines Leistungsniveaus unter festgelegten Bedingungen über einen definierten Zeitraum
 Aufwand zur Benutzung der Software durch unterschiedliche Benutzergruppen
 Benötigte Zeit und Verbrauch an Betriebsmitteln für Aufgabe
 Maß für Möglichkeit der Modifizierung von Software auf Basis interner und externer Einflüsse
 Maß für Offenheit und Portabilität von Software zur Lauffähigkeit auf anderen Soft- und Hardwaresystemen

Qualitätsmerkmale

(DIN ISO 9126)

12

Merkmale

- ♦ **Funktionalität**
- ♦ **Zuverlässigkeit**
- ♦ **Benutzbarkeit**
- ♦ **Effizienz**
- ♦ **Änderbarkeit**
- ♦ **Übertragbarkeit**

Teilmerkmale

Richtigkeit, Angemessenheit, Interoperabilität, Ordnungsmäßigkeit (Normen, Bestimmungen), Sicherheit
 Reife, Fehlertoleranz, Wiederherstellbarkeit
 Verständlichkeit, Erlernbarkeit, Bedienbarkeit
 Zeitverhalten, Verbrauchsverhalten
 Analysierbarkeit, Modifizierbarkeit, Stabilität, Prüfbarkeit
 Anpassbarkeit, Installierbarkeit, Austauschbarkeit, Konformität (gegenüber Normen)

Elementarmerkmale

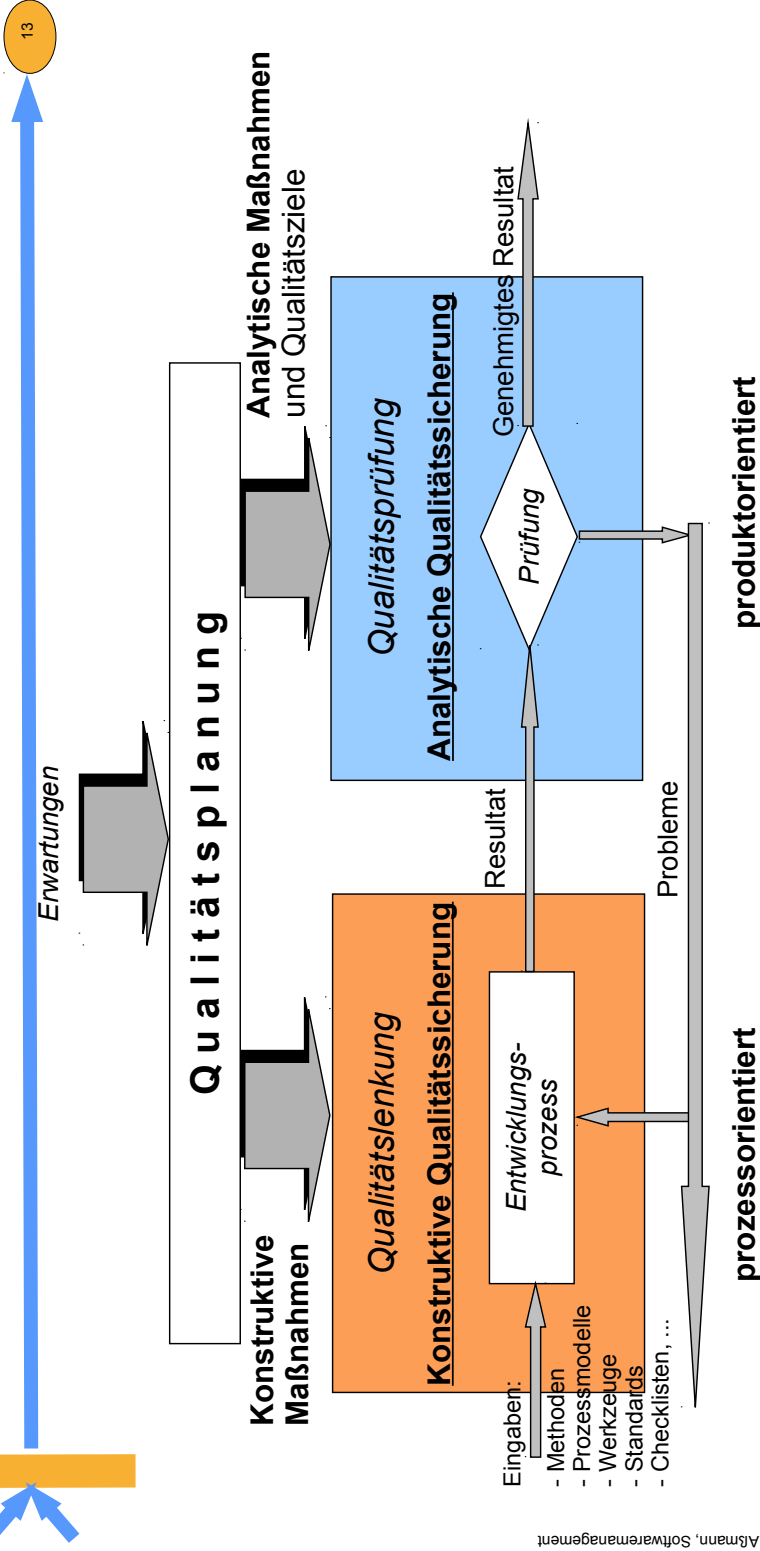
Verständlichkeit
Erlernbarkeit
Bedienbarkeit

Zugänglichkeit
 Selbsterklärungsfähigkeit
 Uniformität
 Einfachheit der E/A

Bsp.: Benutzbarkeit

jeweils Zeit und Zufriedenheit

Qualitätssicherungs-System



Bestandteile des QS-Systems

- ▶ **Qualitätsplanung**
 - Festlegung aller Anforderungen und Ziele an das System und den Projektabwicklungsprozess
 - Bestimmen, Klassifizieren und Wichten aller Qualitätsmerkmale
 - Zugrundelegung von Normen für die Qualitätsplanung
- ▶ **Qualitätslenkung** durch konstruktive Maßnahmen
 - **konstruktive** Maßnahmen bis hin zum Einsatz von SE-Methoden, Werkzeugen
 - **organisatorische** Maßnahmen wie Einsatz von Vorgehensmodellen, Richtlinien, Standards, Checklisten und Dokumentationsvorschriften
- ▶ **Qualitätsprüfung** durch analytische Maßnahmen
 - **statische** Prüfungen (Prüfung der Entwurfsdokumente)
 - **dynamische** Prüfungen (Ausführung des Prüfobjekts, Testen)
 - Analyse und Auswertung des Entwicklungsprozesses nach den häufigsten und gravierendsten Qualitätsmängeln

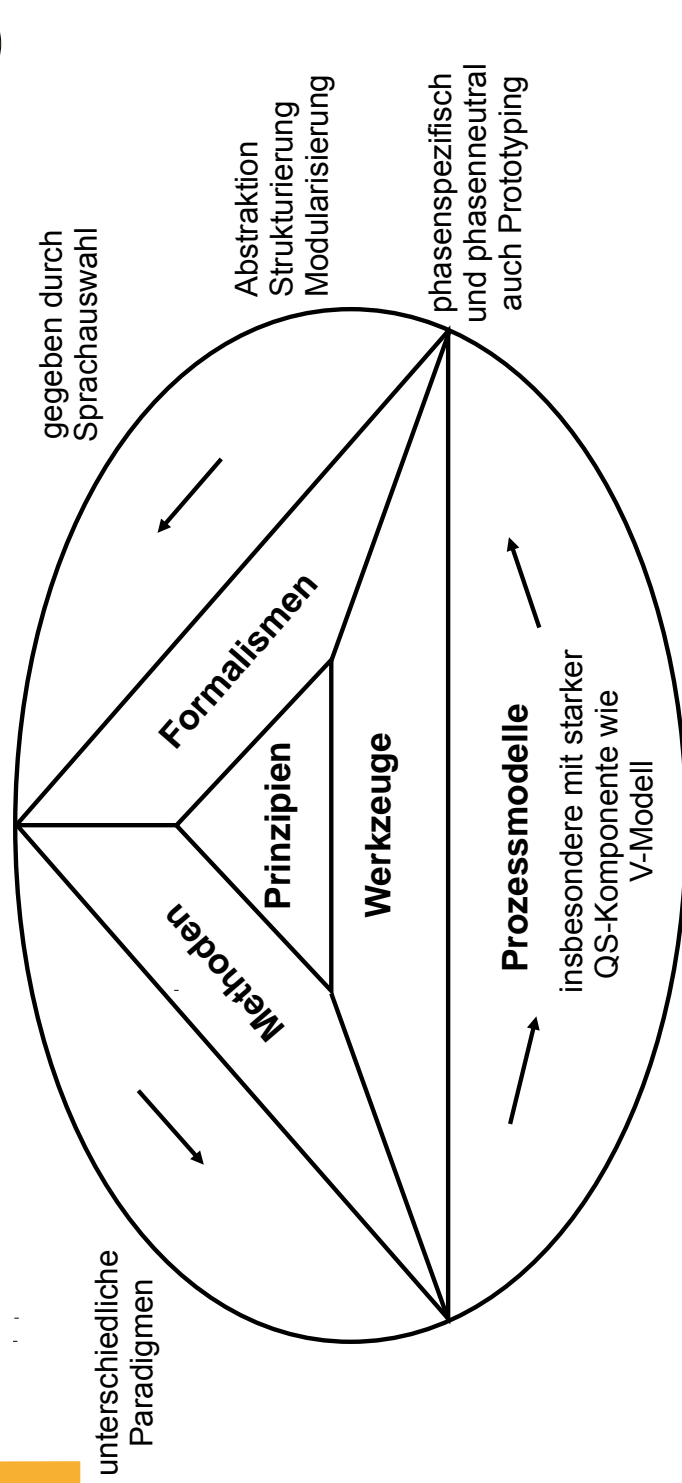
20.3 Konstruktives Qualitätsmanagement

- Konstruktives QM verbessert die Konstruktionsprozess des Produkts durch Qualitätslenkung

15

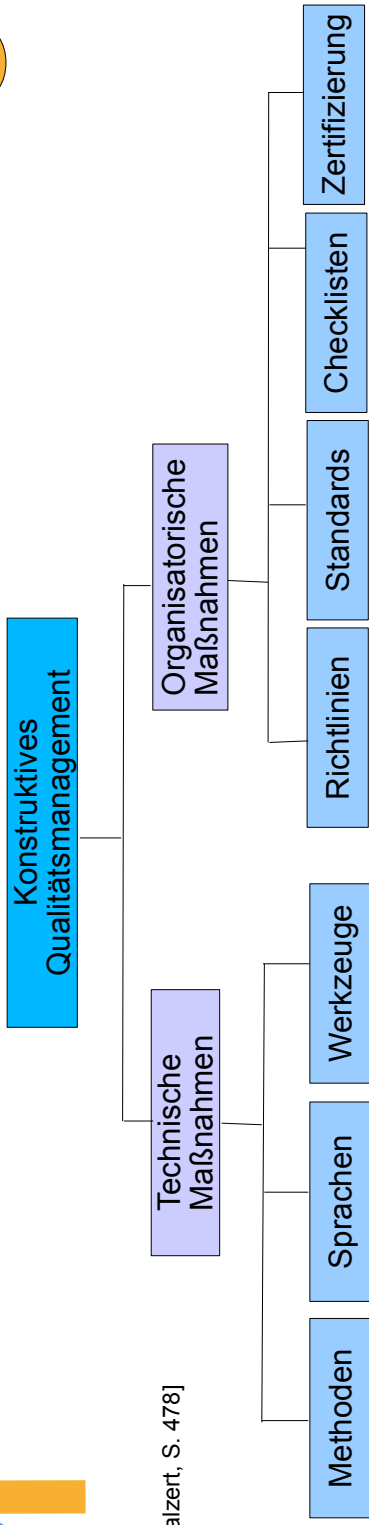
Konstruktive Elemente der Softwaretechnik

16



Maßnahmen zum konstruktiven Q-Management

17



[Balzert, S. 478]

Methoden:

- Einsatz einer **Schätzmethode**, wie Function Point oder COCOMO.
 - Nutzung des **Requirementmanagements**, um Anforderungsstufenkonzepte aufbauen zu können.
 - Förderung der **Persönlichkeitsbildung**, wie fachliche Fortbildung oder psychologisch-orientierte Maßnahmen.
 - Frühzeitige Prüfung der Entwurfs- und Implementierungsanforderungen durch den Aufbau von **Prototypen**
- Programmiersprache** mit strengem Typkonzept, um auch zur Laufzeit Typprüfungen vornehmen zu können.
- Richtlinien:** Projektbegleitende **Dokumentationsfortschreibung** möglichst nach einem Standard und werkzeuggestützt
- Dokumentenmuster für **Pflichtenheft**, dass eine sichere Erfassung aller Anforderungen gewährleistet.
 - **Software-Konfigurationsmanagement** für eine saubere Verwaltung aller bei der Entwicklung entstehender Produkte



Zertifizierung durch externe Organisationen, z.B. TÜV

Quelle: [Wallmüller]

Bsp: Checkliste für Qualität von Anforderungsspezifikationen

18

- ▶ Sind Anforderungen vollständig und widerspruchsfrei? (CCC)
 - Wurden alle Funktionen spezifiziert?
 - Sind alle Algorithmen für Funktionen spezifiziert?
 - Wurden die Datenströme im Kontextmodell in Form von Menge pro Zeit bzw. in Form einer statistischen Verteilung spezifiziert?
 - Sind alle Hardware-Ressourcen spezifiziert?
 - Sind alle Schnittstellen beschrieben?
 - Ist der Initialzustand des Systems spezifiziert?
- ▶ Sind die spezifischen Antwortzeiten realisierbar? (SMART)
 - Wurden für Software-Qualitätsanforderungen Genauigkeitsangaben (Messbarkeitskala, Schwellwerte) spezifiziert?
 - Gibt es zu jeder Funktion Abnahmekriterien?
 - Gibt es Gültigkeitsprüfungen für Daten?
- ▶ Sind die Anforderungen verständlich für die Entwerfer?
- ▶ Ist an spätere Erweiterungen gedacht?
- ▶ Wurde an die Ausbildung des Bedienpersonals gedacht?



Quelle: [Wallmüller]

Aufbauorganisation der QS

19

- a) QS durch externe Unternehmen
- in kleineren Unternehmen Akzeptanzproblem der QS-Mitarbeiter („unproduktiv“)
 - Einsatz externer Subunternehmer evtl. auch ökonomischer

- b) QS durch eigenständige Abt. im Org./DV-Bereich

- QS entweder als Linien- oder Stabsstelle unterhalb des Information Management IM



Total Quality Management (TQM) (1)

20

- ▶ zuerst eingeführt in Japan von Deming als Firmenphilosophie, abgeleitet vom PDCA (Deming)
 - **horizontal:** über alle Abteilungen hinweg und
 - **vertikal:** über alle Leitungsebenen
- ▶ Ziel: Kundenzufriedenheit (Qualität, Kosten und Zeit)

Erfahrungswerte

- **Zufriedener Kunde:** erzählt positives Erlebnis 4 - 8 mal weiter
- **Unzufriedener Kunde:** erzählt „Geschichte“ 9 - 16 mal weiter ==> unkontrollierter negativer Multiplikator
- **Neukunden** zu gewinnen ist schwieriger und aufwendiger, nämlich 6 mal teurer als Bestandskunden gut zu betreuen

Konsequenzen¹⁾

- Jeder **unzufriedene Kunde** ist eine Herausforderung an Fähigkeiten des Unternehmens
- **Beschwerde-Management** heißt, aus einem unzufriedenen Kunden einen zufriedenen Kunden zu machen und Kundenbindung zu erzeugen
- **nötig:** - Bewusstsein schaffen
- gezielte Schulung
- konsequente Umsetzung

⇒ **Nur ein zufriedener Kunde bleibt auch ein Kunde**

¹⁾ **Quelle:** Knöll u. a.: Entwicklung und Qualitätssicherung von Anwendungssoftware; Spektrum Verlag 1996

Total Quality Management TQM (2)

21

14 Führungspflichten zur Verbesserung der Qualität und Produktivität (nach Deming):

- ▶ Für Unternehmen
 - Unternehmenspolitik muss Willen zur Verbesserung klar aufzeigen.
 - Prüfung primär, um die Prozesse zu verbessern und die Kosten zu reduzieren.
 - Der Preis eines Produktes allein zählt nicht, sondern das unablässige Verbessern von Prozessen und Produkten.
 - Institutionalisierung des Trainings, insbesondere Training on the job.
 - Leiterschaft ist zu institutionalisieren.
 - Barrieren zwischen Organisationseinheiten abreißen.
 - Das Qualitätsmanagementsystem kontinuierlich verbessern.
 - Leistungsquoten beseitigen.
- ▶ Für Belegschaft
 - Adäquate Einstellung zur Qualität.
 - Angst abbauen, Vertrauen und Klima für Innovation schaffen.
 - Förderung von Ausbildung und Selbstverbesserung für jeden im Unternehmen.
 - Andere am Erfolg teilhaben lassen.

Standards zur QS

22

DIN 55350-11	Definition der Qualitätseigenschaften	ISO/IEC 14598-1	Modell für Erkennen der Qualität, Bewertung von Softwareprodukten
ISO/IEC 9126-1	6 Hauptkategorien für Softwarequalität: u. a. Usability	DIN 66271	Softwarefehler und ihre Beurteilung durch Kunden und Lieferanten
ANSI/IEEE 829	Standard for Software Test Documentation	DIN 66270	Bewerten von Software-dokumenten, Qualitätsmerkmale
ANSI/IEE 1008	Standard for Software Unit Testing, Modultest	BS/ISO/IEC 25000	Anforderungen an Software Produkt Qualitätsanforderungen und Evaluation

20.4 Analytisches Qualitätsmanagement

23

Qualitätsprüfung analysiert die Qualität von Produkt und Prozess und versucht, Verbesserungen vorzuschlagen

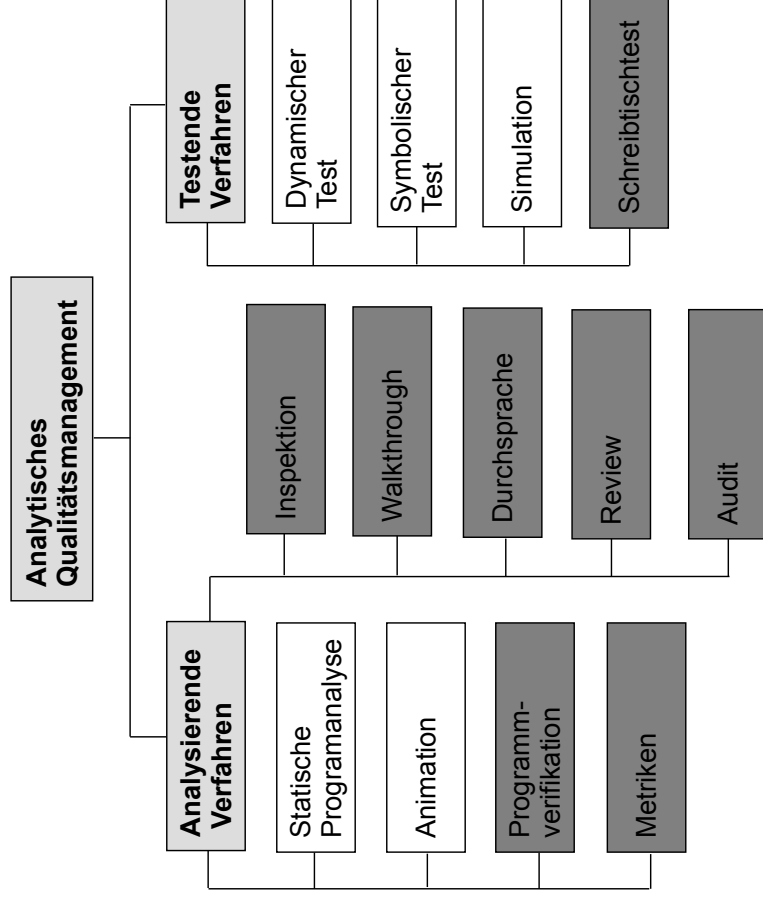
- Analyse
- Test



Softwaremanagement, © Prof. Uwe Alßmann

Analytisches Qualitätsmanagement

24



Quelle: [nach Balzert, S. 479]



Analytische QS-Ziele (1)

Zunächst sollten Qualitätsziele festgelegt werden.

Qualitätszielbestimmung für das Projekt in Form von einfachen Güteklassen:

Produktqualität sehr gut gut normal nicht relevant

Funktionalität	x			
Zuverlässigkeit	x			
Benutzbarkeit	x			
Effizienz			x	
Änderbarkeit			x	
Übertragbarkeit	x			

Quelle: [Balzert]

QS-Ziele (2)

a) Fehlerverhütung

- Ziele des Projektes festlegen (Req.-Katalog, PH)
- systematisches Entwickeln mit SE-Methoden
- Projektmanagement (Q. eines Produkts entsteht aus Q. der Phasenergebnisse ==> Summationseffekt)
- Qualifikation der Mitarbeiter
- Güte der Hilfsmittel/Tools

b) Fehlerentdeckung/Beseitigung (mittels analysierender Verfahren)

- Audits
- Reviews
- Code-Inspektionen, Walkthroughs
- statische Programmanalyse
- Verifikation
- c) Verbesserung der Entwicklung (Entwicklungshilfen)
- Integrationshilfen
- Testfallbibliotheken
- Fehlersuchhilfen

QS-Verifikation

Verifikation und Validation (V & V) (vgl. ISO 9000:2005)

- ◆ **Verifikation:** „Bestätigung durch einen objektiven Nachweis, dass die Anforderungen für eine bestimmte Anwendung oder einen bestimmten Gebrauch erfüllt sind.“
- **Validation:** „Bestätigung, dass ein Produkt ein vom Kunden erstelltes Lastenheft und damit die Anforderungen an den Gebrauch durch den Kunden erfüllt“.
- „**Verifikation** heißt, die Arbeit richtig durchzuführen, und **Validation** heißt, die richtige Arbeit durchzuführen“ [B. Boehm]

20.4.1. Reine Analytische QS-Verfahren

28

Reviews

29

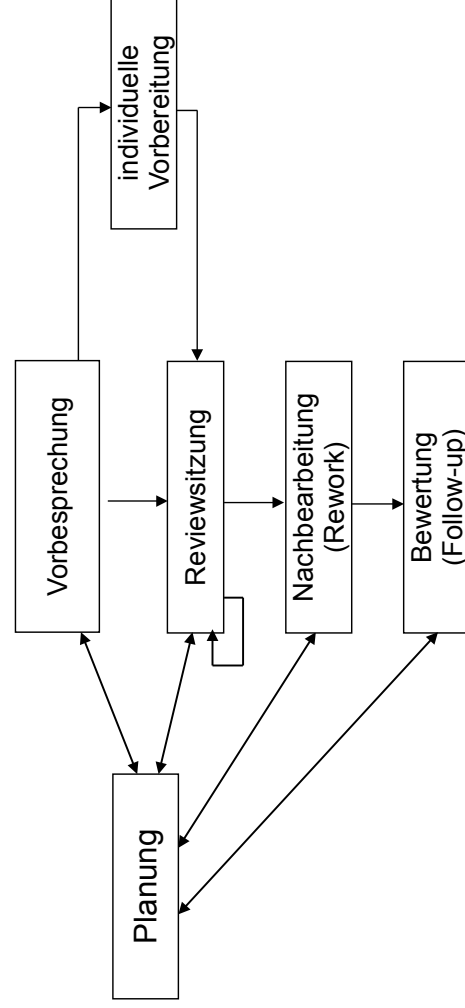
Ein **Review** ist eine manuelle Prüfmethode mit festgelegtem Ablauf, mit der ein bestehender Zustand (z.B. Projektergebnisse) oder die Wirksamkeit eingeführter Maßnahmen einem **Team von Gutachtern** vorgelegt und von diesen kommentiert oder genehmigt werden (Projektplan-Review, Anforderungs-Review, Entwurfs-Review, Code-Review u.a.)

- ▶ Reviews fokussieren sich auf Produktqualität

Quelle: Frühauf u. a.: Software-Prüfung - eine Fibel; vgl. auch IEEE-Norm 729-1983: Standard for Software Reviews and Audits

Normaler Ablauf eines Reviews

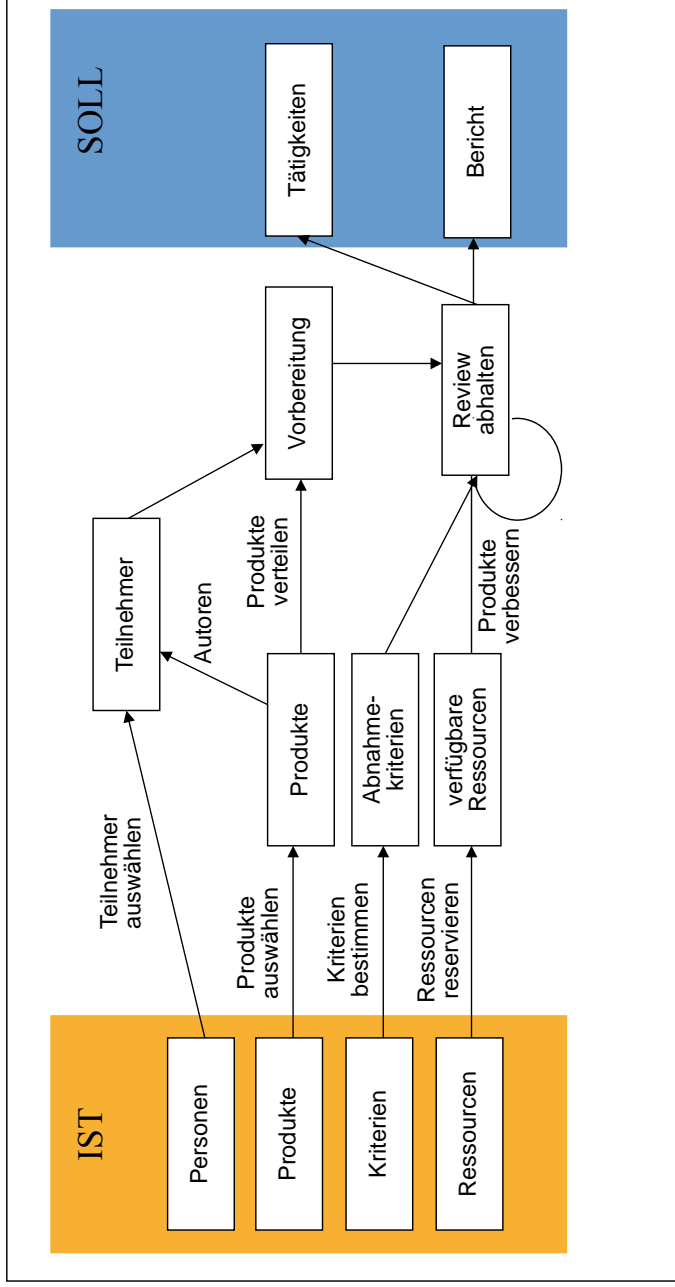
30



Quelle: [Wallmüller]

Bsp.: Abhängigkeitsdiagramm(-graph) eines Review als Vorgangspfeil-Netz

31

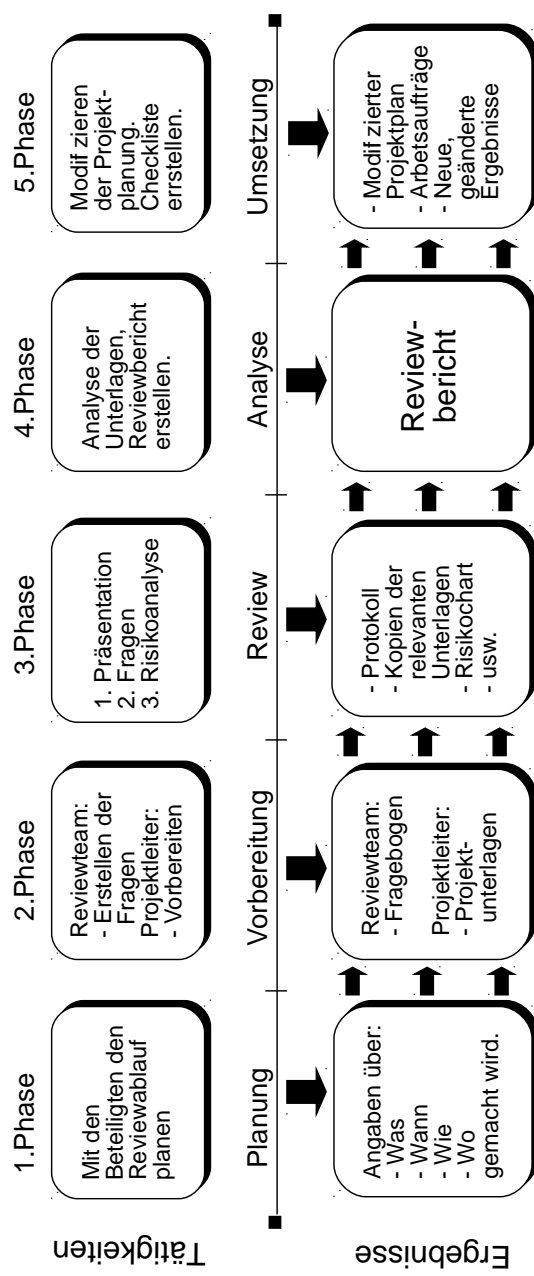


Quelle: [Zuser, W. S. 122]

Projektreview-Ablauf

32

Der Reviewablauf besteht aus fünf Schritten:



Quelle: [1 Jenny]

Arten von Reviews (1)

33

- ▶ **Inspektionen:** Die Dokumentation des Prüfgegenstandes wird Zeile für Zeile gelesen (von den Inspektoren)
 - Teilnehmer (Moderator, Autor, Gutachter, Protokollführer)
 - Inspektionen sind in jeder Phase möglich (Bsp. Code-Inspektion)
 - Vorbereitung (Einladung, Richtlinien, Rollen, Prüflinge)
 - Prüfung anhand von Checklisten
 - Inspektoren tragen Fehler vor
 - Inspektionsprotokoll durch Protokollführer
 - ggf. Freigabe durch Moderator
- ▶ **Walkthroughs:** (abgeschwächtes Review)
 - Die **Funktionalität** des Prüfgegenstandes wird anhand von vorbereiteten Beispielen und Testfällen **durchgespielt**.
 - ohne Moderator, evtl. ohne individuelle Vorbereitung, Autor stellt sein Prüfobjekt vor.

Arten von Reviews (2)

34

- ▶ **Round-Robin-Review:**
 - Die Gutachter sollen in der Vorbereitung nach Argumenten suchen, warum der Prüfling gut ist.
 - In der Sitzung trägt jeder sein Plädoyer vor, die anderen Gutachter intervenieren
 - Argumente für und gegen den Prüfling werden notiert.
- ▶ **Peer Review:** („Späher“)
 - Gutachter werden „eingeschlossen“, untersuchen die Prüflinge und erstellen Gutachten.
 - Ein Moderator leitet das Team.
 - Das Team wird entweder ad hoc zusammengestellt oder existiert als permanente Einrichtung („professionelle Peers“)

Bsp.: Checkliste für Grobentwurfs-Reviews

35

Performance

- ⇒ Gibt es Hinweise auf die Nichterfüllung von Performance-Anforderungen?

Benutzungsschnittstelle

- ⇒ Sind die Layouts der Benutzungsschnittstelle einheitlich?
- ⇒ Sind die Bildschirmmasken mit Informationen nicht überladen?
- ⇒ Sind die Bildschirmausgaben übersichtlich?
- ⇒ Ist die Benutzerführung ausreichend?
- ⇒ Sind die Benutzereingaben auf ein Minimum beschränkt?

Daten

- ⇒ Wurde das Datenmodell geprüft?
- ⇒ Gibt es fehlende oder nicht benutzte Variablen in einem I-, O- oder Update-Modul?
- ⇒ Gibt es falsche oder fehlende Datentypen in einem Input-, Output- oder U.-Modul?

Funktionalität

- ⇒ Ist in einem Verarbeitungsmodul ein Teil nicht vorhanden, überflüssig oder falsch?
- ⇒ Sind in einem V.-modul logische Bed. nicht vorhanden, überflüssig oder falsch?

Außerdem:

Schnittstellen, Dokumentation, Standards, Syntax der Entwurfsbeschr., . . .

Audits (1)

36

Ein **Audit** ist eine *systematische* und *unabhängige* Untersuchung, bei der sowohl die Übereinstimmung mit Spezifikationen, Standards, vertraglichen Vereinbarungen oder anderer Kriterien (Angemessenheit, Einhaltung vorgegebener Vorgehensweisen und Anweisungen als auch deren Wirksamkeit und Sinnhaftigkeit) überprüft werden.

- ▶ **Audit der Produktqualität:** quantitative Bewertung der Konformität des Produktes mit den geforderten Produktmerkmalen lt. Pflichtenheft
- ▶ **Audit der Prozessqualität:** Überprüfung der Elemente eines Prozesses auf Vollständigkeit und Wirksamkeit z. B. im Vergleich zu einem Vorgehens- oder Prozessmodell
- ▶ **Audit des QS-Systems:** Prüfung, ob vorhandene Elemente des QS-Systems entsprendend den Anforderungen vollständig, dokumentiert und wirksam sind.
- ▶ Audit des Finanzmanagements
- ▶ Audit des Entwicklungs- und Managementprozesses:
 - z. B.: - Produktivität des Projektteams , Einhaltung vorgegebener Standards
- ⇒ Während eines größeren Projekts sollten mehrere Audits durchgeführt werden

Audits (2)

37

- ▶ Systematische und unabhängige Untersuchung mit formalem Charakter
- ▶ Validation der Systeme, Prozesse, Produkte mit den Vorgaben (Spezifikationen) durch Dritte, meist spezialisierte Audit-Firmen
- ▶ Audits werden durch ausgebildete Auditoren nach einem definierten Ablauf durchgeführt:
 - Vorbereitung: Die Auditoren benötigen zur Vorbereitung ausgewählte Untersuchungs- und/ oder Prüfdokumente (z.B. Projektplan, Vorgehensmodell, zugrunde liegende Vorgaben, Metriken u.a.).
 - Durchführung: erfolgt durch Interviews mit Prozessverantwortlichem und paralleler Dokumentensichtung
 - Abschluss: zum Abschluss des Audits erfolgt ein vorläufiges Feedback an alle Beteiligten
- ▶ Ergebnisse werden in einem ausführlichen **Audit-Bericht** dokumentiert

Quelle: [Kollektiv]

Audits (3)

38

Schritte für den Ablauf eines Audits sind:

- Ziele der Überprüfung definieren
- Umfang und Anwendungsbereich definieren
- Initiierung
- Überblick gewinnen und Daten sammeln
- Analyse und Auswertung der gesammelten Daten
- Lösungs- und Verbesserungsvorschlag erarbeiten
- Erstellen und Präsentieren des Ergebnisberichts

Der angetroffene Ist-Zustand wird mit den Zielsetzungen und dem Erreichungsgrad verglichen.

Im Ergebnisbericht sind alle Maßnahmen aufzuführen, die zur Einhaltung des angestrebten Projektverlaufs eingeleitet werden müssen.

Quelle: ANSI-Norm N45.2.10-1973

Statische Programmanalyse

39

- ▶ mit der Hilfe von Werkzeugen ==> Vorlesung “Software-Werkzeuge”
- ▶ **Abstrakte Interpretation** interpretiert das Programm statisch mit abstrakten Werten, die Fehlerwerte entdecken lassen
 - Sicherheitsprüfungen (z.B. Buffer overflow analysis, driver protocol analysis)
- ▶ **Vertragsprüfung** mit Werkzeugen
 - Theorembeweiser, gute Übersetzer für Programmiersprachen mit Verträgen wie Eiffel

Statische Analyse mit Softwarewerkzeugen

40

Bei der statischen Analyse von Programmen kann man folgende Aussagen werkzeuggestützt erhalten:

- **syntaktische Informationen**
 - z. B. Komplexitätsgrade, Aufrufgraphen, Strukturbäume, Typkonflikte, undefinierte Variable, Endlosschleifen, Aufrufe nicht existierender Prozeduren, unerlaubte Verschachtelung von Schleifen und Verzweigungen
- **semantische Informationen**
 - z. B. Steuerflusssanomalien, Datenflussunverträglichkeiten, deklarierte aber nicht verwendete Variable, nicht initialisierte Variable, falsche Verwendung globaler und lokaler Variablen
- **lexikalische Informationen**
 - z. B. Länge und Häufigkeit von Programmeelementen, unerreichbarer Code, falsche bzw. nicht referenzierte Sprungmarken, falsche Parameterübergaben

20.4.2. Testende QS-Verfahren

41



Softwaremanagement, © Prof. Uwe Alßmann

SW-Testmethoden (1)

42

- ▶ **Statische Prüfungen:** Siehe auch Vorlesung Softwarewerkzeuge (WS)
- ▶ **Dynamische Prüfungen:**
 - Datenbezogenes Testen mit Testdaten: Datenstrukturen, Referenz- oder Betriebsdaten (bei großen Programmen lassen sich kaum alle Datenkombinationen erproben)
 - Funktionsbezogenes Testen: abschnittsweiser Vergleich des Codes incl. E/A-Verhalten mit der Spezifikation
 - Ablaufbezogenes Testen: werden alle Schleifen, Verzweigungen durchlaufen?
 - (Kontrollfussorientierter Test - „Durchspielen“ aller Fälle; ==> aufwendig)
 - vergessene Funktionen werden nicht gefunden !!



SW-Testmethoden – Datenbezogener Test

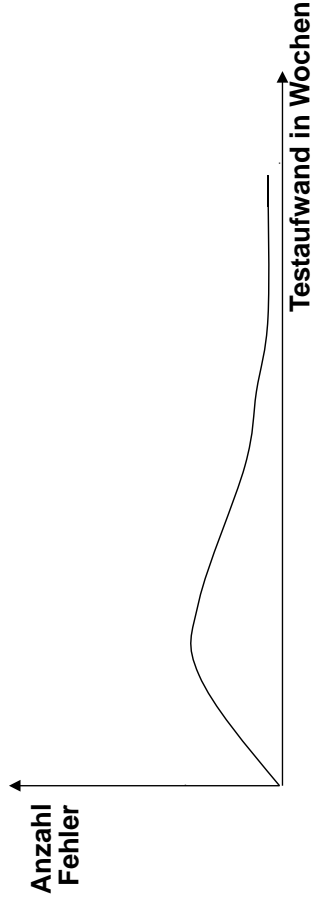
43

▶ **Regressionstest:** Vergleich zweier Versionen des gleichen Programms

Siehe Vorlesung Softwaretechnologie-II

▶ Test-Endkriterien

- a) aus Erfahrung: in 5% aller Module stecken 95% der Fehler
==> Stichproben; Ende, wenn 5% fehlerhafte Moduln gefunden
- b) nach Fehlerrate:



SW-Testmethoden (Black Box-Test)

44

Ziel: Feststellung von Abweichungen gegenüber Anforderungen bzw. Spezifikation (innere Struktur ist nicht von Interesse)

Methoden:

- **Äquivalenzklassenbildung**
 - Einteilung der E/A-Daten in Äquivalenzklassen (gütige und ungütige)
- **Grenzwertanalyse**
 - Testfälle an den Grenzen der Wertebereiche
- **Intuitive Testfallermittlung** (kein eigentliches Verfahren)
 - zusätzliche Testfälle durch Intuition (Liste möglicher Fehler aus Erfahrung, Standardfehler)
- **Funktionsabdeckung**
 - Testfälle für Normal- und Ausnahmeverhalten
Vermeidung von Redundanz durch Testfallmatrix

⇨ Testfälle für SW-Module, -Komponenten, ...,

SW-Testmethoden (White Box-Test)

45

Ziel: Entdeckung von Fehlern durch ablauforientierte Testfälle
(interne Struktur / Quelltext interessiert)

Methoden:

- ◆ **Pfadabdeckung**
(wenigstens eine Mindestzahl von Pfaden prüfen)
- ◆ **Anweisungsabdeckung**
(entsprechend Spezifikation, alle oder Auswahl)
- ◆ **Bedingungsabdeckung**
- ◆ **Zweig-/Bedingungsabdeckung**
- ◆ **Abdeckung aller Mehrfachbedingungen**

QS: Anforderungsdefinition und Abnahmekriterien

46

- **Abnahmekriterien bereits während der Anforderungsdefinition** (Pflichtenheft)
 - Aufdeckung von Lücken, Überschneidungen, Widersprüchen
(hat oft die Überarbeitung von Anforderungen zur Folge)
 - Grundlage für den Nachweis des Erfüllungsgrades
 - ein oder mehrere Abnahmekriterien zu genau einer Anforderung

Den Anwender interessieren vorrangig ergebnisorientierte (Black-Box-) Abnahmekriterien

- Funktionsabdeckung
 - Äquivalenzklassenbildung
 - Grenzwertanalyse
 - Intuitive Testfallermittlung (Ergänzung der o. g. aus Erfahrung)
- **ablauforientierte (White-Box-) Abnahmekriterien**
(nicht nur Prüfung, ob Funktionen ausgeführt werden, sondern ob auch die Reihenfolge richtig)
 - Zweigabdeckung
 - Bedingungsabdeckung
 - Pfadabdeckung

The End



47