

# Teil II – DO

## 20. Qualitätssicherung und -management

Prof. Dr. Uwe Aßmann  
Lehrstuhl Softwaretechnologie  
Fakultät Informatik  
TU Dresden  
Version 13-1-1, 13.06.13

1) Fehler – Warum man QS braucht

1) Qualitätsbegriff

2) Konstruktive

Qualitätssicherung

3) Analytische QS

1) Analyseverfahren

2) Testverfahren

4) Zertifizierung

5) Aufgabenmanagement

Softwaremanagement, © Prof. Uwe Aßmann, Technische Universität Dresden, Fakultät Informatik

## 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
- ▶ <http://www.qz-online.de/>
- ▶ American Society for Quality <http://www.asq.org/>
- ▶ M. Gharbi, A. Koschel, A. Rausch, G. Starke: Basiswissen für Softwarearchitekten. dpunkt-Verlag. 2013. Ausbildungsmaterial zum iSAQB-Standard für Zertifizierung von Softwarearchitekten
- ▶ codeBeamer Werkzeug als Beispiel für Requirements- und Qualitätsmanagement

- <http://intland.com/documents/CodeBeamer-Requirement-Management-2013-Low.pdf>

# Standards zur QS

3

Norm	Erläuterung
DIN 55350-11	Definition der Qualitätseigenschaften von Softwareprodukten
ISO/IEC 14598-1	Modell für Erkennen der Qualität, Bewertung
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-Dokumentation
DIN 66270	Bewerten von Softwaredokumenten, Qualitätsmerkmale
ANSI/IEEE 1008	Standard for Unit Testing, Modultest
BS/ISO/IEC 25000	Anforderungen an Software-Produkt-Qualitätsanforderungen und Evaluation
EN ISO/IEC 17024	Konformitätsbewertung – Allgemeine Anforderungen an Stellen, die Personen zertifizieren



## 20.1 Warum man Qualitätssicherung und -management braucht

4



# Fehleranzahl und -kosten

5

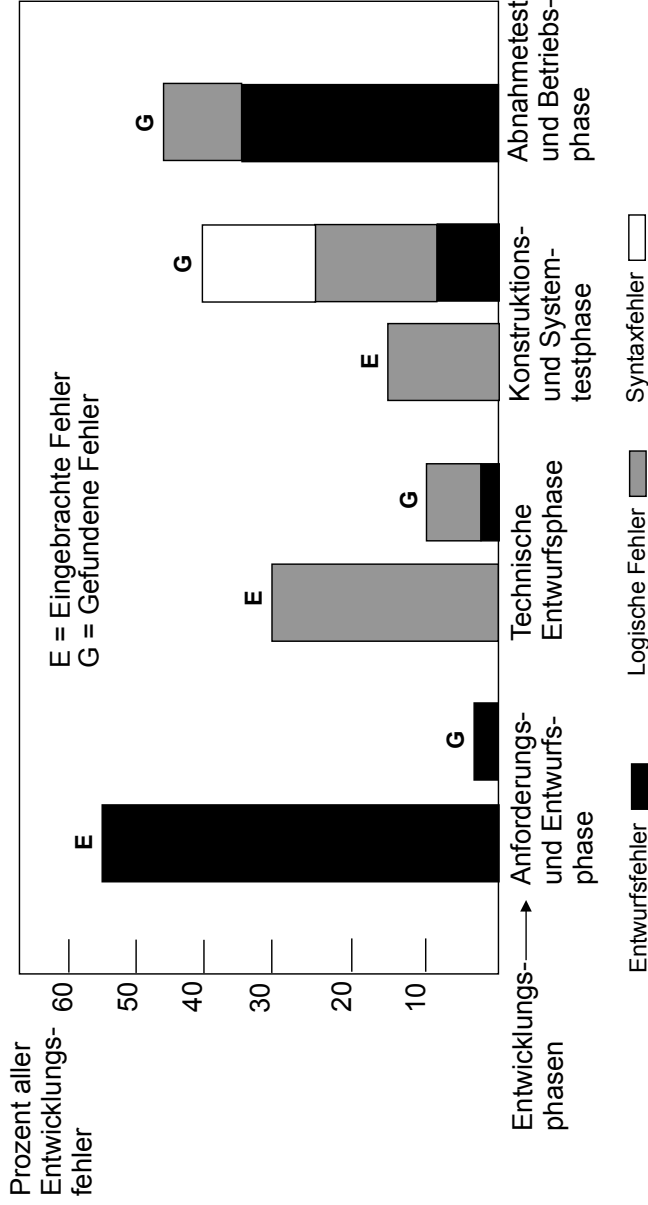
(Empirische Daten)

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

6



**55% aller Fehler entstehen in der Anforderungs- und Entwurfsphase**

Quelle: [ Balzert, S. 487 ]

# 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

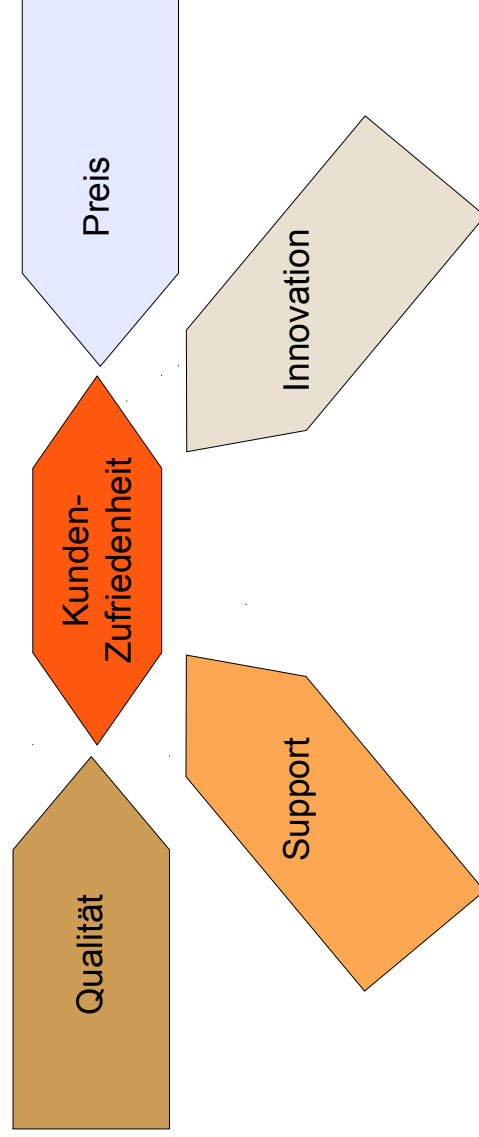
(transparente Kommunikation, Verständigung über gemeinsame Ziele)

Quelle: DIN EN ISO 9000:2000-01

Stand: Februar 2000 - .DQS

# Stammkundengeschäft

Der wesentliche Erfolgsfaktor, einen Stammkunden zu erwerben und zu halten, ist eine hohe Qualität der ausgelieferten Software und Dienstleistung.



# Total Quality Management (TQM) (Ständige Qualitätsverbesserung)

- ▶ 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

## Konsequenzen<sup>1)</sup>

- 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**



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

## 20.1.2 Qualitätsbegriff

13



# Was ist Qualität?

14

**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, Norm für Qualitätsmanagement]

**Quality** is fitness for use.

- ▶ **Qualität**
  - **des Produkts (Produktqualität)**
  - **des Entwicklungsprozesses (Prozessqualität)**
  - **der Beteiligten (Personenqualität)**
- ▶ **Qualitätsmerkmale:**
  - **Teilmerkmale (Kriterien)**
  - **Elementarmerkmale**
  - **Indikatoren (Metriken)**



Quelle: [BalzertSM]

# Unterteilung der Qualitätsmerkmale nach ISO/IEC 25000, früher DIN ISO 9126

15

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

## Merkmal

## Teilmerkmale

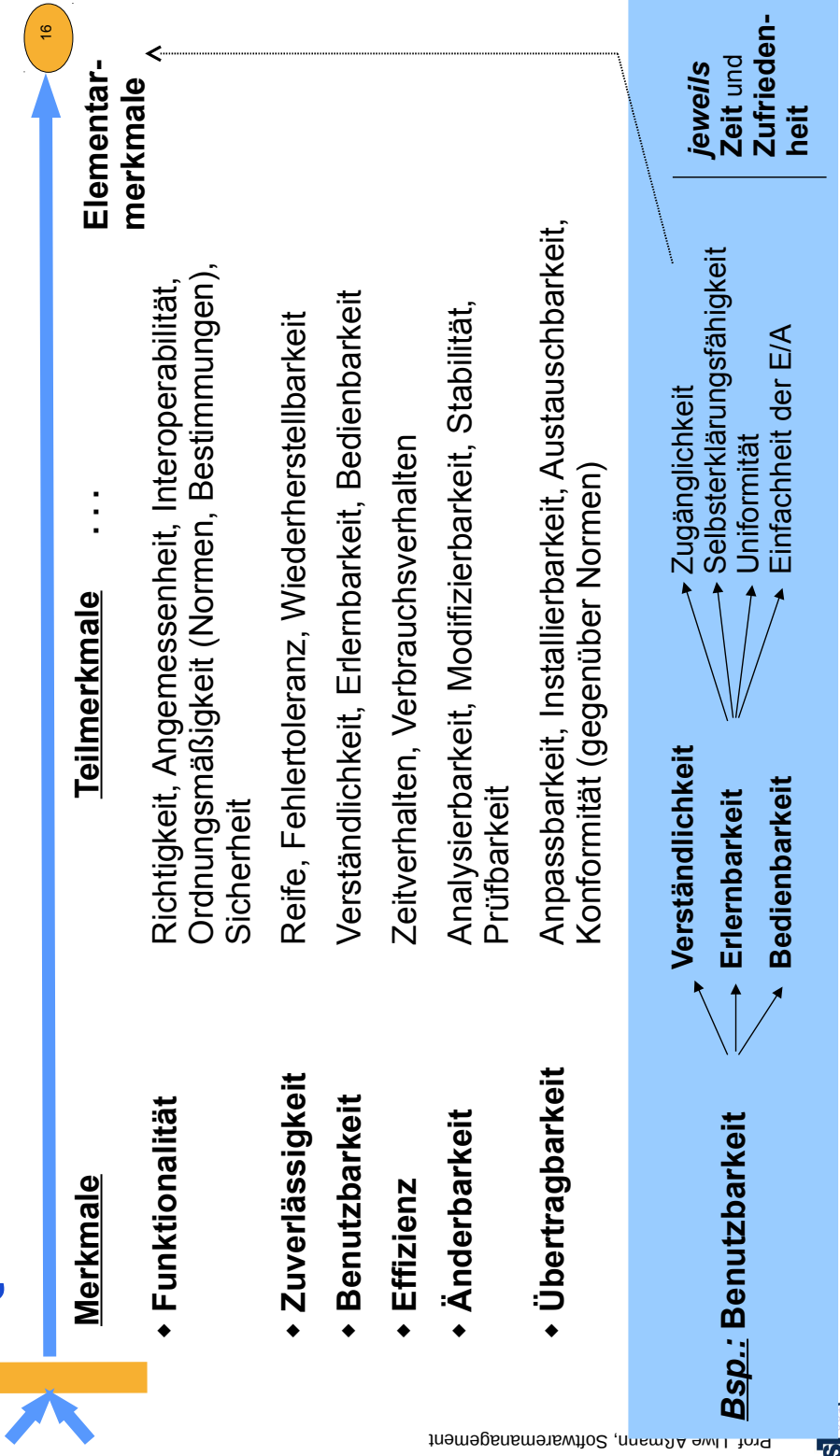
## Merkmalsbeschreibung

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)

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

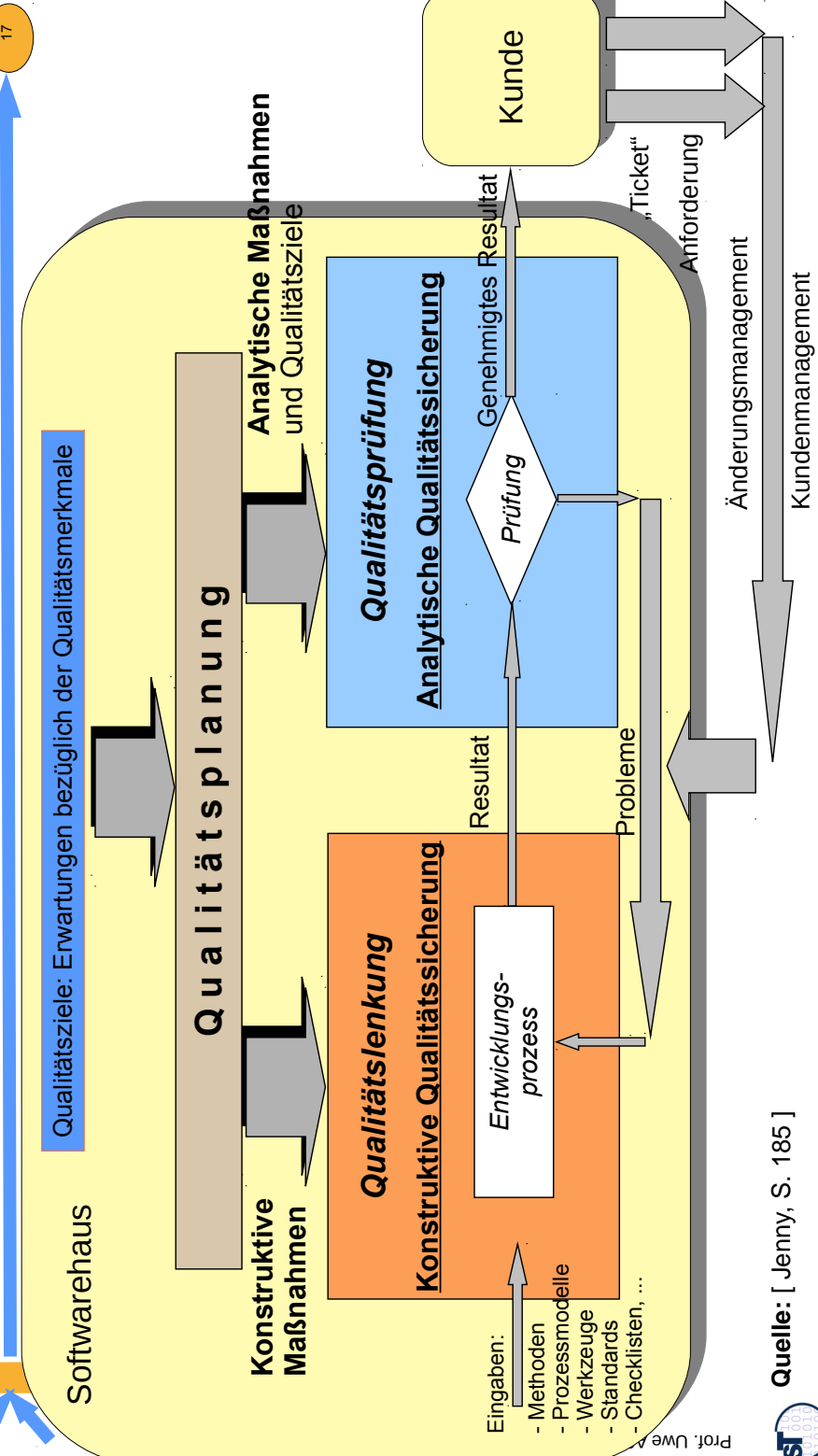


# Qualitätsmerkmale ISO/IEC 25000



Prof. Uwe Armann, Softwaremanagement

# Qualitätsmanagement-System (im Application Lifecycle Management ALM)



Quelle: [ Jenny, S. 185 ]



Prof. Uwe Armann

# Regelkreis des QM

Das QM läuft in einem PDCA-Regelkreis, da es geplant, durchgeführt, überprüft und verbessert werden muss:

- ▶ **Qualitätsplanung (PLAN)**
  - Ist-Zustand ermitteln, Ziele und Rahmenbedingungen für das QM festlegen
  - Konzepte und Abläufe erarbeiten
- ▶ **Qualitätslenkung/-steuerung (DO)** - die in der Planphase gewonnenen Ergebnisse werden umgesetzt
- ▶ **Qualitätssicherung (CHECK)** - Auswerten qualitativer und quantitativer Qualitätsinformationen
  - Entgegennahme von Fehlern, die beim Kunden gefunden werden ("tickets")
  - Entgegennahme von neuen Anforderungen für Funktionen ("feature request")
  - Kosten-Nutzen-Betrachtungen (Wirtschaftlichkeitsbetrachtungen): Was kostet welche Fehlerkorrektur, welches neue Feature/Funktion?
  - Releaseplanung: Wann kommt eine neue Funktionalität ins Produkt?
  - Überprüfen von gemachten Annahmen
- ▶ **Qualitätsgewinn (ACT)** -
  - Maßnahmen umsetzen zur Steigerung der Produktqualität und Prozessoptimierung.
  - Erfolge und Ergebnisse werden kommuniziert.
  - Fehlerkorrektur
  - Einführung von neuen Funktionen

# Bestandteile des QM

- ▶ **Qualitätsplanung**
  - Festlegung aller Anforderungen und Ziele an das System und den Projektentwicklungsprozess
  - 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**
  - **Prozessgestützte Qualitätsprüfung:**
    - Analyse und Auswertung des Entwicklungsprozesses nach den häufigsten und gravierendsten Qualitätsmängeln
  - **Produktgestützte Qualitätsprüfung:**
    - **statische** Prüfungen (Prüfung der Entwurfsdokumente)
    - **dynamische** Prüfungen (Ausführung des Prüfobjekts, Testen)



# Festlegung von Qualitätszielen für Projekte

20

- ▶ Zunächst sollten Qualitätsziele festgelegt werden, separat für die Stakeholdergruppen: Nutzer, Entwickler, Management
- ▶ Qualitätszielbestimmung für das Projekt in Form von einfachen ordinalen Güteklassen. Beispiel:

Quelle: [Balzert]

Produktqualität	sehr gut	gut	normal	nicht relevant
Nutzerqualitäten	x			
Funktionalität	x			
Zuverlässigkeit		x		
Benutzbarkeit	x			
Sicherheit			x	
Effizienz				x
Entwicklerqualitäten		x		
Änderbarkeit				
Übertragbarkeit	x			
Return of Investment	x			
Managerqualitäten				
Markttattraktivität		x		

# Qualitätsziele und abgeleitete Maßnahmen für Zuverlässigkeit (Bsp.)

21

- ▶ a) Verbesserung der Fehlerverhütung
  - Ziele des Projektes festlegen (Req.-Katalog, PH)
  - Projektmanagement: Die Qualität eines Produkts entsteht aus Qualität der Phasenergebnisse ==> Summationseffekt
  - Qualifikation der Mitarbeiter
- ▶ b) Fehlerentdeckung/Beseitigung (mittels analysierender Verfahren)
  - Audits, Reviews, Code-Inspektionen, Walkthroughs
  - statische Programmanalyse
  - Verifikation
- ▶ c) Verbesserung der Systemstruktur
  - Metriken, Refactoring
- ▶ d) Verbesserung der Entwicklung (Entwicklungshilfen, Werkzeuge)
  - Integrationshilfen
  - Testfallbibliotheken
  - Fehlersuchhilfen (Debugging)

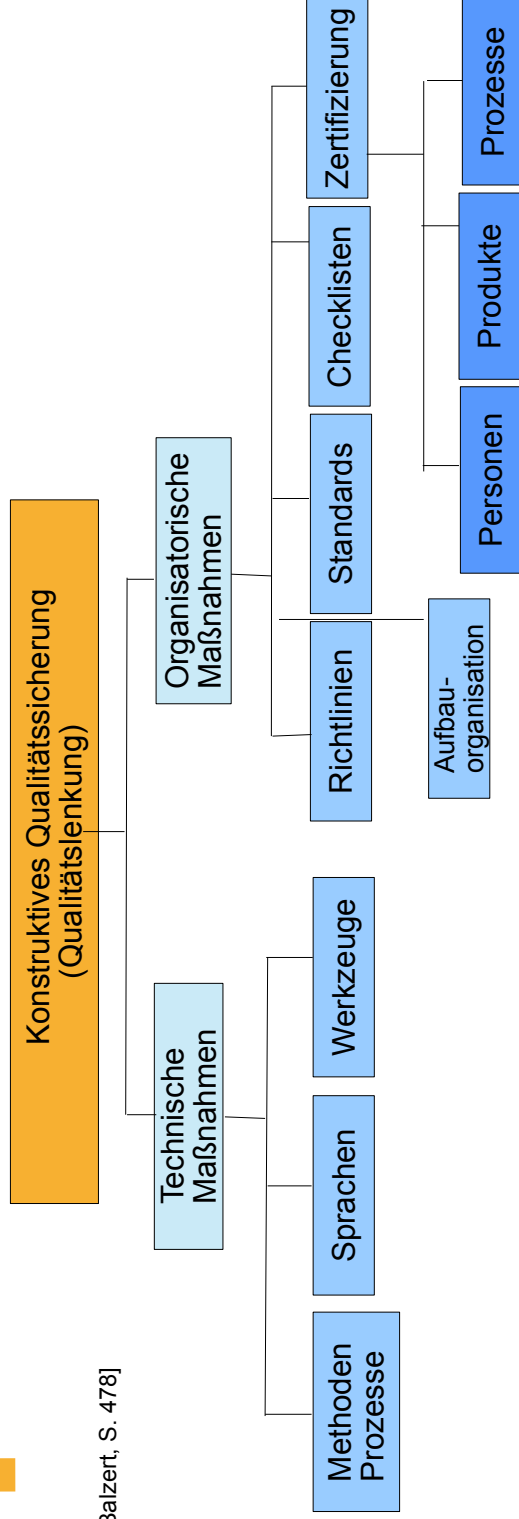
# 20.2 Konstruktive Qualitätssicherung im Prozess (Qualitätslenkung)



- Konstruktive QS verbessert die Konstruktionsprozess des Produkts durch Qualitätslenkung



# Maßnahmen zum konstruktiven QS



[Balzert, S. 478]



# Beispiele für technische Maßnahmen zur konstruktiven QS

26

## ▶ Methoden:

- Einsatz einer **Schätzmethode**, wie Delphi, 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**

## ▶ Sprachen

- Einsatz von **Modellgetriebener Entwicklung**, um Verfolgbarkeit von Anforderungen zum Code zu realisieren
- **Programmiersprache** mit strengem Typkonzept, um auch zur Laufzeit Typprüfungen vornehmen zu können.

## ▶ Werkzeuge

- Ticketingsysteme, um Fehler zu erfassen und zu korrigieren, um Anforderungen zu sammeln
- Metrikwerkzeuge, um Architektur und Code nach Güte zu beurteilen

# Beispiele für organisatorische Maßnahmen zur konstruktiven QS

27

## ▶ Richtlinien:

- Definition von Entwicklungsprozessen in einer **Prozesslandkarte (process map)**
- 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

- der Entwicklungsprozesse durch externe Organisationen, z.B. TÜV (ISO 9000, SPICE, PRINCE)
- der Mitarbeiter (ISTQB, ISAQB)
- der Produkte

# Bsp: Checkliste für Qualität von Anforderungsspezifikationen

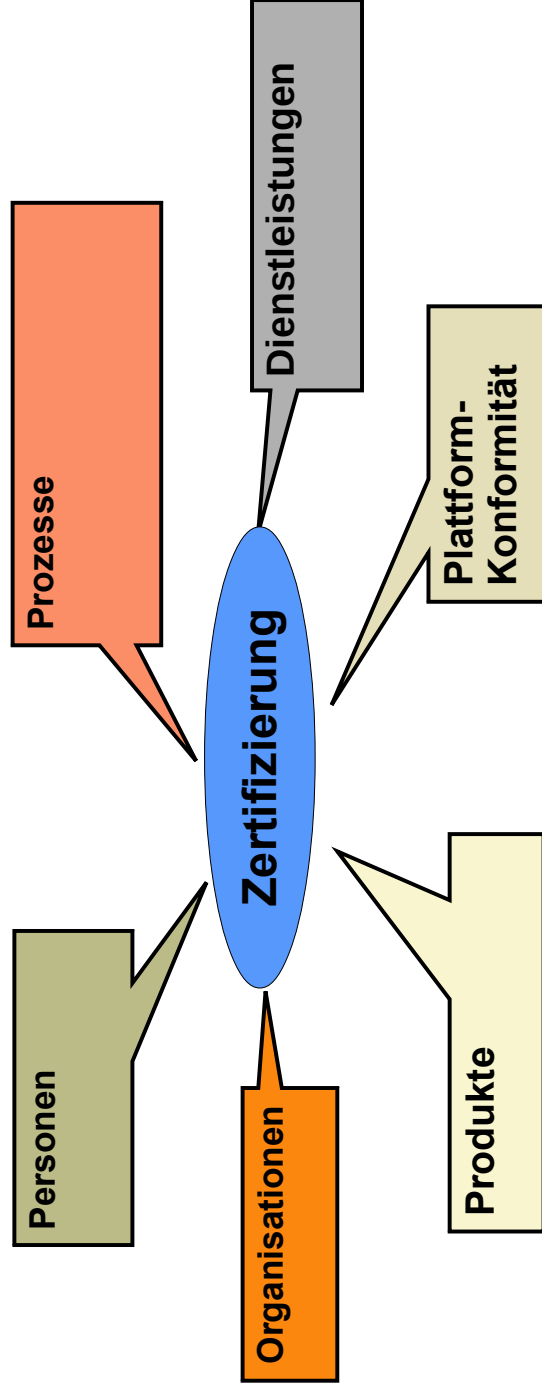
28

- ▶ 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 (Messbarkeitsskala, 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?

## 20.2.2 Zertifizierung

29

Als **Zertifizierung** (von lat. „certe“ = bestimmt, gewiss, sicher und „facere“ = machen, schaffen, verfertigen) bezeichnet man ein Verfahren, mit dessen Hilfe die Einhaltung bestimmter Anforderungen nachgewiesen wird.  
[Wikipedia]



# Was kann alles zertifiziert werden?

30

- ▶ Zertifizierung von Prozessen
  - Zertifizierung eines Erstellungsprozesses oder Managementsystems (zum Beispiel nach ISO 9001, ISO 14001).
  - Bis Ende 2009 über 1 Mio. Zertifikate ISO 9001; über 200k Zertifikate ISO 14001
- ▶ Zertifizierung von Organisationen
  - Zertifizierung der Informationssicherheit nach BS 7799 oder ISO/IEC 27001.
  - Zertifizierung der IT-Umgebung nach IT-Grundschutz mit Grundschutz-Auditoren und das BSI
- ▶ Zertifizierung von Produkten
  - Für Zertifizierungsstellen, die Zertifizierungssysteme für Produkte oder Dienstleistungen betreiben, besteht die EN 45011 bzw. der ISO/IEC Guide 65.
  - Zertifizierung von Softwareprodukten in Hinblick auf Funktionalität und Qualität.
    - TCSEC (USA), ITSEC (EU), Common Criteria (CC).
    - In Deutschland erfolgt die Zertifizierung durch das BSI.
- ▶ Zertifizierung von Personen
  - Entwickler
  - Tester
  - Softwarearchitekten

# Was kann alles zertifiziert werden? (ctd.)

31

- ▶ Zertifizierung von Dienstleistungen
  - ITIL ist eine Richtlinie für IT-Dienstleistungen. Es definiert drei Zertifizierungsniveaus: Foundation, Practitioner und Service Manager.
  - ITIL Foundation
  - ITIL Service Manager
  - ITIL Practitioner Configuration Management, Incident Management, Problem Management, Release Management, Change Management, Service Level Management, Financial Management, Capacity Management, Availability Management

# Zertifizierung in Software-Ökosystemen

32

Ein **Softwareökosystem** besteht aus einer Welt von Zulieferern, die auf einer Software-Plattform eines Herstellers oder Konsortiums Produkte durch den Bau von **Komponenten (plugins, add-ons)** zur Plattform erzeugen (*gemischte Wertschöpfung*)

- ▶ Beispiele:
  - Microsoft-Ökosystem
  - Java-Ökosystem
  - AutoSAR: Steuerungssoftware im Auto
  - GenIVI.com: in-vehicle infotainment im Auto
- ▶ Zertifizierung von Plugins und Produkten für Plattformkonformität:
  - Im Bereich Linux und freie Software ist ein wichtiges zertifizierendes Institut das kanadische LPI.
- ▶ [http://de.wikipedia.org/wiki/Liste\\_der\\_IT-Zertifikate](http://de.wikipedia.org/wiki/Liste_der_IT-Zertifikate)

# Personenzertifizierung

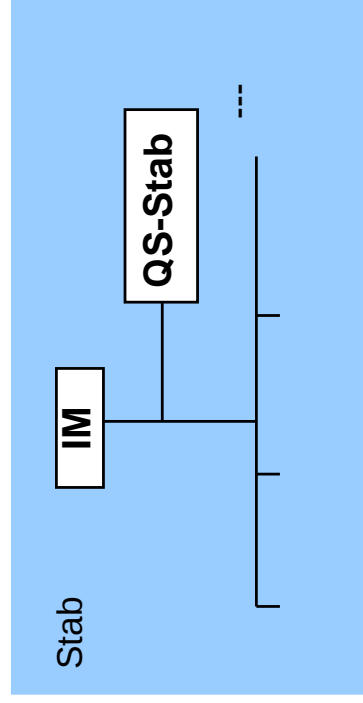
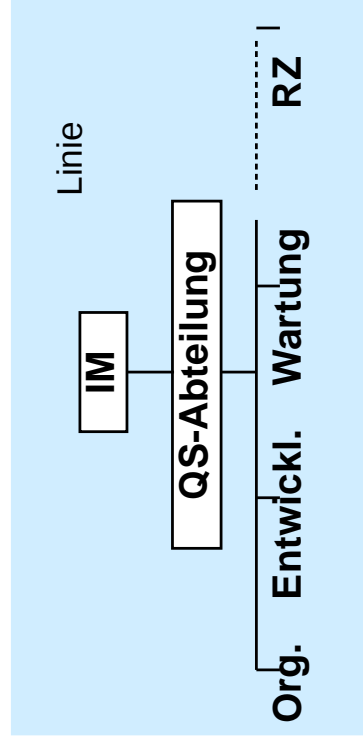
33

- ▶ Nachweis von Ausbildungsstandards oder besonders ausgearbeiteten Fachnormen bei Personenzertifizierungen.
- ▶ Zertifizierung der Mitarbeiter zur Dokumentation von Fähigkeiten, Qualifikation und Kompetenz. Siehe dazu Liste der IT-Zertifikate.
  - PMP (Project Management Professional) durch das PMI (Project Management Institute)
  - IPMA-Zertifikate Level D-A für Projektmanager
  - ISQF Tester-Zertifizierungen: Foundation level – Advanced level
  - iSAQB-Zertifizierungen
    - Vom International Software Architecture Qualifications Board angebotene Zertifizierung für Softwarearchitekten in verschiedenen Abstufungen und Spezialisierungsrichtungen.
    - Foundation Level: Certified Professional for Software Architecture (CPSA/CPSA-F)
    - Advanced Level (in Vorbereitung beim iSAQB CPSA-A)
    - Expert Level (in Vorbereitung beim iSAQB CPSA-E)
- ▶ Norm für Zertifizierungsstellen: EN ISO/IEC 17024 („Konformitätsbewertung – Allgemeine Anforderungen an Stellen, die Personen zertifizieren“)

## 20.2.3 Aufbauorganisation der QS

34

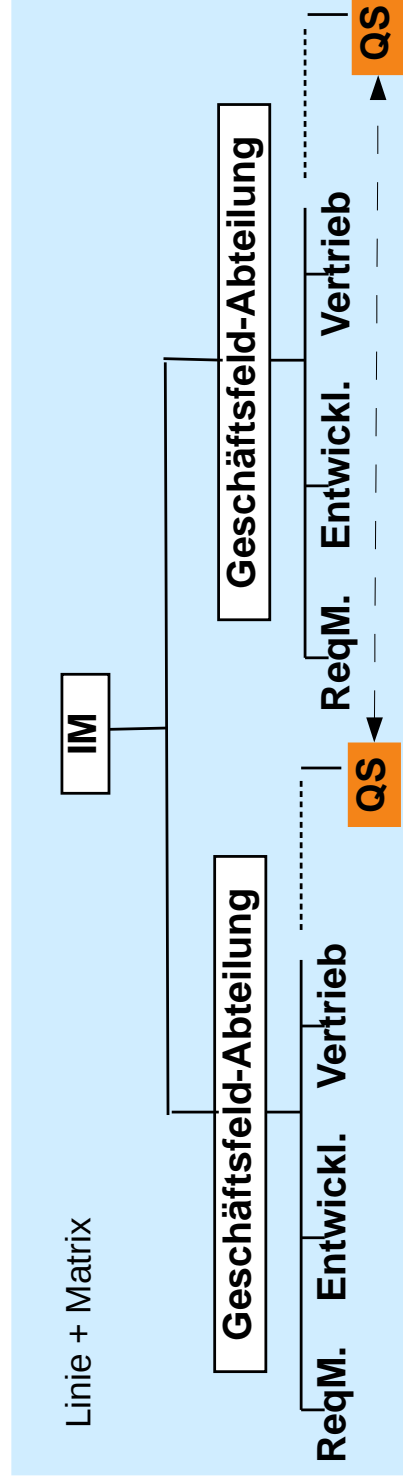
- ▶ **a) QS durch externe Unternehmen (Berater, Spezialisten, Auditing)**
  - in kleineren Unternehmen Akzeptanzproblem der QS-Mitarbeiter („unproduktiv“)
  - Einsatz externer Subunternehmer (Test- und QS-Dienstleister)
  - Wichtig, wenn man intern nicht weiterkommt
- ▶ **b) QS durch eigenständige Abt. im Org./DV-Bereich**
  - QS unterhalb des Information Management (IM), entweder als Linienstelle (Abteilung)
- ▶ **c) Projektmanagement-Büro oder Stabsstelle**



## Forts.: Aufbauorganisation der QS

35

- ▶ **d) QS durch Kompetenzteams**
  - als kleine Teams oder auch Rollen in einem Produkt- oder Geschäftsfeldabteilung
  - Die QS-Teams aller Abteilungen bildet "Projekt" in einer Matrix



# 20.3 Analytische Qualitätssicherung des Produkts (Qualitätsprüfung)

36

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

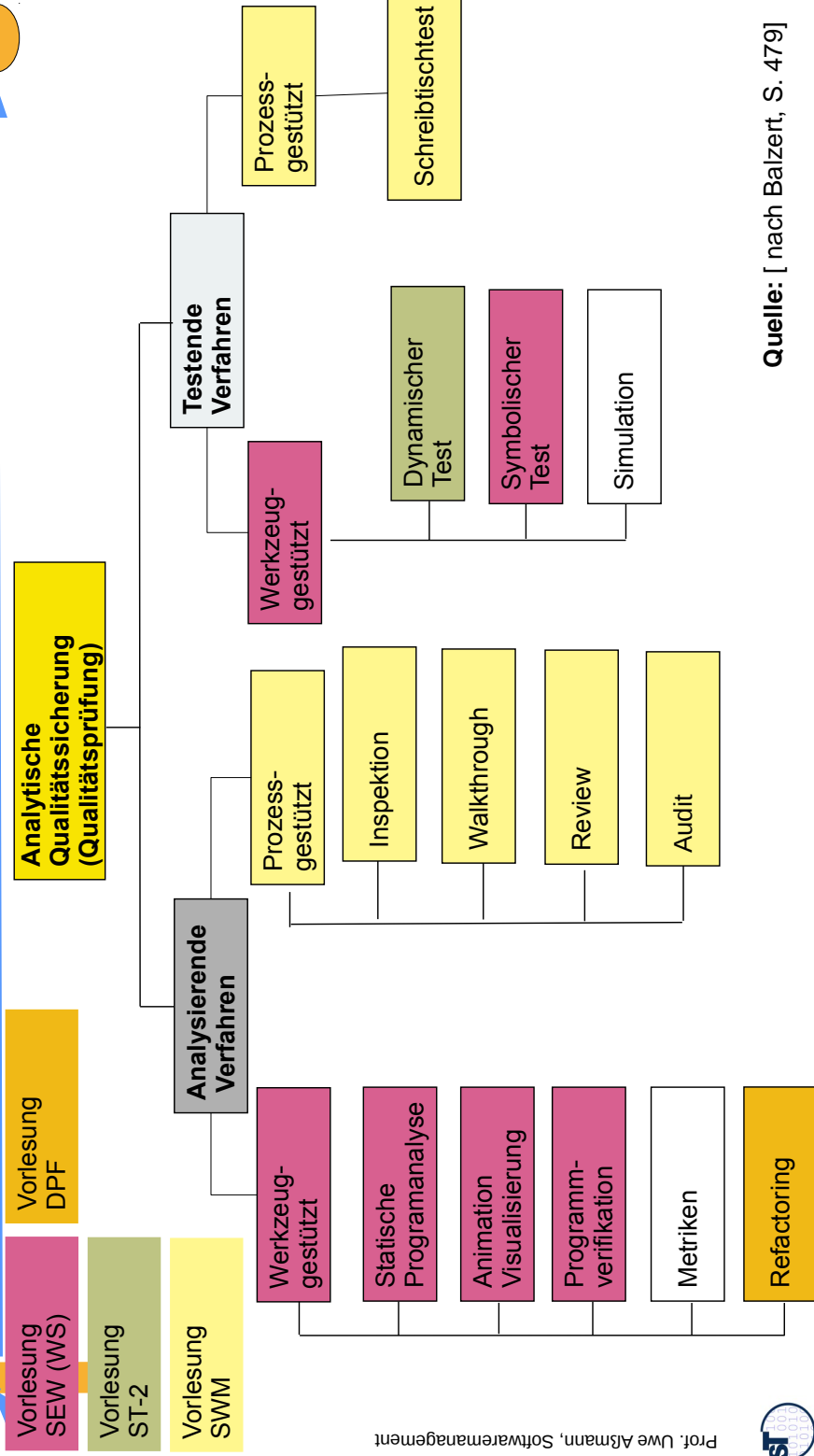
- Analyse
- Test



Softwaremanagement, © Prof. Uwe Alßmann, Technische Universität Dresden, Fakultät Informatik

# Analytisches Qualitätssicherung (Qualitätsprüfung)

37





## 20.3.1. Analytische QS-Verfahren

- 20.4.1.1 Prozessgestützte QS-Prüfung

38



Softwaremanagement, © Prof. Uwe Alßmann, Technische Universität Dresden, Fakultät Informatik

## Reviews

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.)

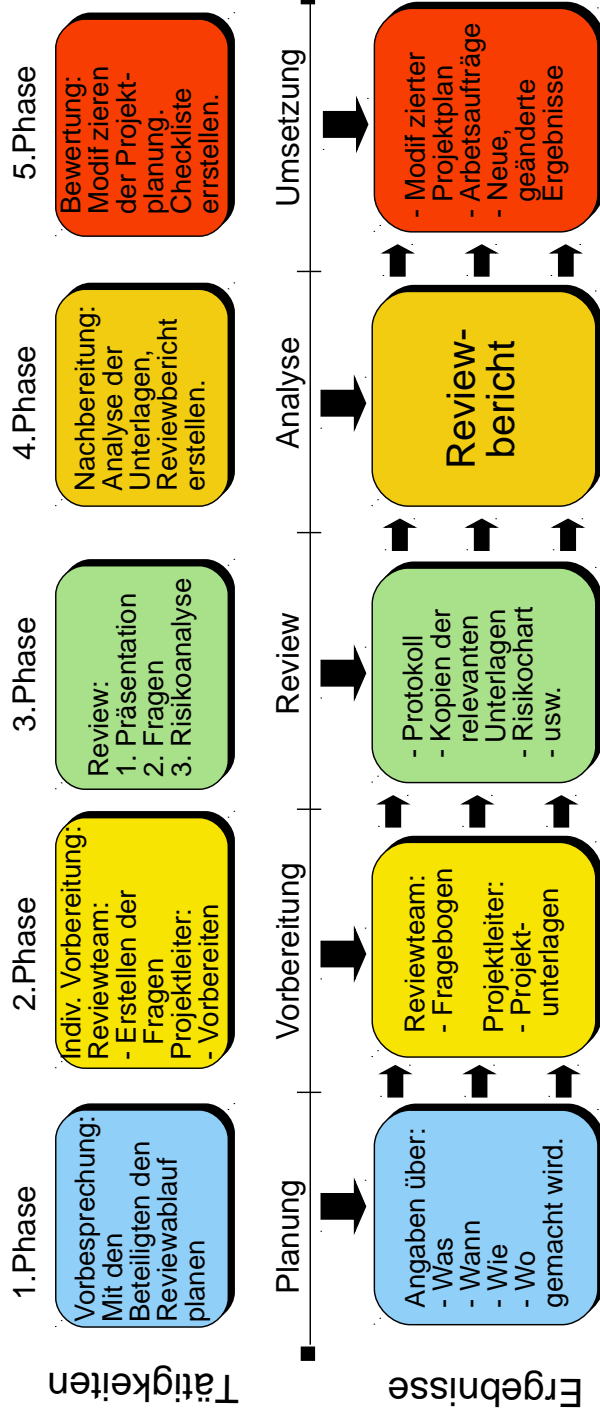
- ▶ Reviews fokussieren sich auf Produktqualität

39



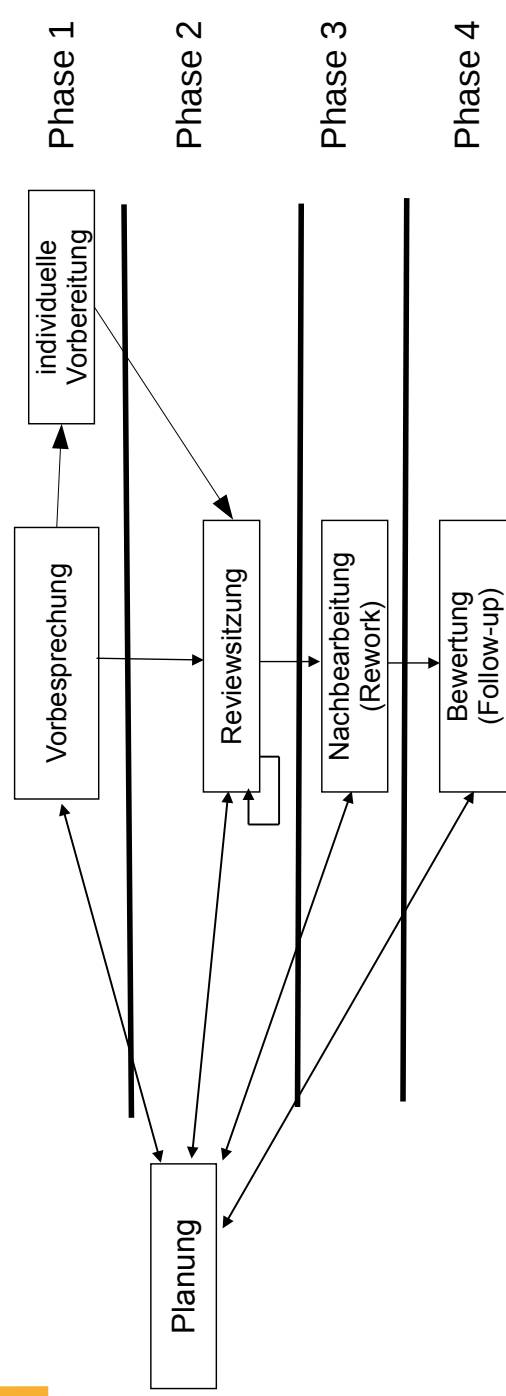
# Projektreview-Ablauf

Der Reviewablauf besteht aus fünf Phasen:



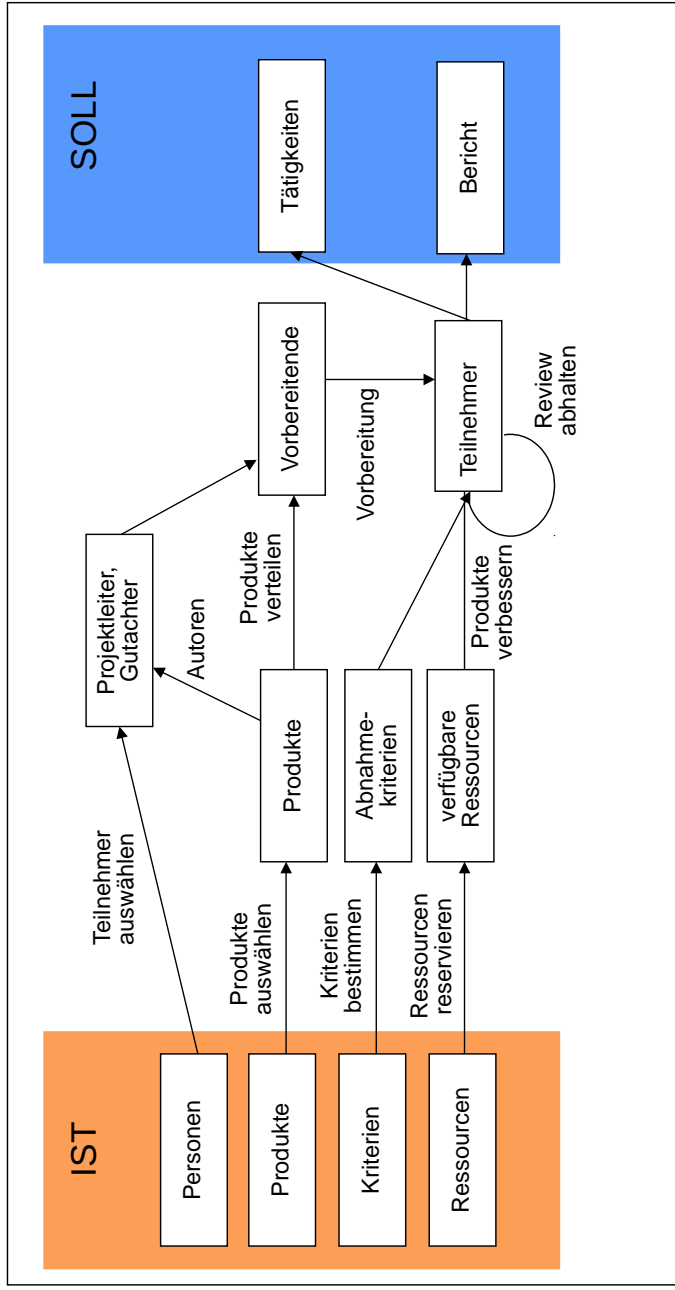
# Wdh.: Ablauf eines Reviews als

# Aktivitätsdiagramm (Vorgangsknotennetz)



# Wdh.: Abhängigkeitsdiagramm(-graph) eines Review als Vorgangspfeil-Netz über Daten

42



Quelle: [ Zuser, W. S. 122 ]

## Arten von Reviews (1)

43

- ▶ **Inspektionen:** (white-box review) Die Dokumentation des Prüfgegenstandes wird von den Inspektoren Zeile für Zeile gelesen und geprüft
  - Teilnehmer (Moderator, Autor, Gutachter, Protokollführer)
  - Inspektionen sind in jeder Phase möglich (Bsp. Code-, Design-, Pflichtenheft-Inspektion)
  - Vorbereitung nötig (Einladung, Richtlinien, Rollen, Prüflinge)
  - Prüfung anhand von Checklisten
  - Inspektoren tragen Fehler vor
  - Inspektionsprotokoll durch Protokollführer
  - ggf. Freigabe durch Moderator
- ▶ **Walkthroughs:** (black-box 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)

44

## ▶ Round-Robin-Review:

- Die Gutachter sollen in der Vorbereitung nach Argumenten suchen, warum die Qualität des Prüflings hoch 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

45

## 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) - Formale Reviews von Außen

46

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)

47

- ▶ 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 fordern von extern ausgewählte Untersuchungs- und/ oder Prüfdokumente an (z.B. Projektplan, Vorgehensmodell, zugrunde liegende Vorgaben, Metriken u.a.).
  - Durchführung: erfolgt in der Firma durch Interviews mit Prozessverantwortlichem und Dokumentensichtung
  - Abschluss: zum Abschluss des Audits erfolgt ein vorläufiges Feedback an alle Beteiligten
  - Ergebnisse werden in einem ausführlichen Audit-Bericht dokumentiert
- ▶ Audits müssen oft durchgeführt werden, um Berichtspflichten nach außen zu erfüllen
  - SOX Sarbanes-Oxley-Act
  - Basel I-III Kriterien
  - Prüfung auf Gemeinnützigkeit

## 20.3.1. Analytische QS-Verfahren

49

- 20.3.1.2 Werkzeuggestützte Qualitätsprüfung eines Produktes



Softwareengineering, © Prof. Uwe Alßmann, Technische Universität Dresden, Fakultät Informatik

## Statische Programmanalyse

50

- ▶ mit der Hilfe von Werkzeugen ==> Vorlesung "Software-Werkzeuge"
- ▶ **Lexikalische Analyse** ermittelt lexikalische Informationen
  - z. B. Länge und Häufigkeit von Programmelementen, unerreichbarer Code, falsche bzw. nicht referenzierte Sprungmarken
- ▶ **Syntaktische Analyse** und **Metriken** ermitteln syntaktische Informationen
  - z. B. Komplexitätsgrade, Aufrufgraphen, Strukturbäume, Architekturprinzipien, Endlosschleifen, Aufrufe nicht existierender Prozeduren, unerlaubte Verschachtelung von Schleifen und Verzweigungen
  - **Layout-Prüfung** und **Verbesserung**: Pretty-printing
- ▶ **Statische semantische Analyse** ermittelt **semantische Informationen**
  - **Abstrakte Interpretation** interpretiert das Programm statisch mit abstrakten Werten, die Fehlerwerte entdecken lassen
  - Typprüfungen (wie Typkonflikte, falsche Parameterübergaben)
  - Steuerflussanomalien wie Sicherheitsprüfungen (z.B. Buffer overflow analysis, driver protocol analysis)
  - Datenflussunverträglichkeiten wie deklarierte aber nicht verwendete Variable, nicht initialisierte Variable, falsche Verwendung globaler und lokaler Variablen



## Statische Programmanalyse (Forts.)

### ▶ Statische Vertragsprüfung mit Werkzeugen

- Theorembeweiser, gute Übersetzer für Programmiersprachen mit Verträgen wie Eiffel
  - Prüfung von Qualitätsverträgen z.B. mit Quality (www.qualityne.org)
- ### ▶ Model checking prüft die Gültigkeit von Prädikaten in einem Zustandssystem

51

## 20.3.2. Testende QS-Verfahren

- (Wdh aus Softwaretechnologie-II, zum Selbststudium)

52

# SW-Testmethoden

54

## ▶ 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?
  - (Kontrollflussorientierter Test - „Durchspielen“ aller Fälle; ==> aufwendig)
  - vergessene Funktionen werden nicht gefunden !!

Quelle: [nach Zehnder, C.,A.. Informatik-Projektentwicklung; Teubner Verlag 1991]

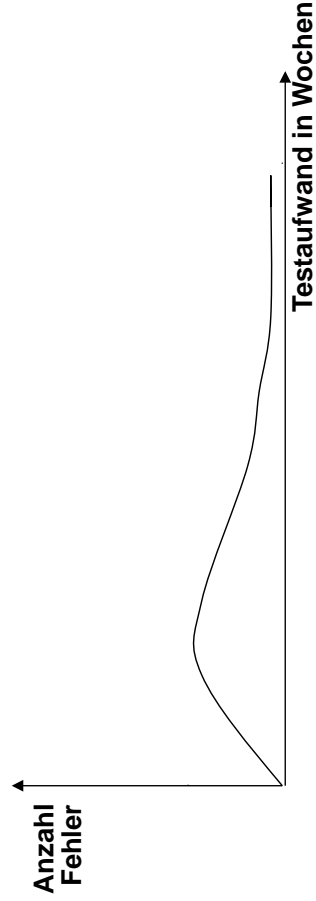
# SW-Testmethoden – Datenbezogener Test

55

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

Siehe Vorlesung Softwaretechnologie-II

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





# SW-Testmethoden (Black Box-Test)

56

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

## Methoden:

- **Äquivalenzklassenbildung**
  - Einteilung der E/A-Daten in Äquivalenzklassen (gültige und ungültige)
- **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, ...,

# Überdeckungs-Testmethoden (White Box-Test)

57

▶ Ziel: Entdeckung von Fehlern durch ablauforientierte Testfälle

- interne Struktur / Quelltext muss bekannt sein
- Es wird eine Teilmenge aller möglichen Pfade durch Testfälle abgedeckt (Pfadabdeckung)

▶ Methoden:

- Pfadabdeckung (wenigstens eine Mindestzahl von Pfaden prüfen)
- Anweisungsabdeckung (entsprechend Spezifikation, alle oder Auswahl)
- Bedingungsabdeckung: Abdeckung aller If- und Case-Bedingungen
- Zweig-/Bedingungsabdeckung: Abdeckung aller einzelnen alternativen Pfade
- Abdeckung aller Kombinationen von verschachtelten Mehrfachbedingungen
- n-Schleifenabdeckung: Abdeckung der ersten n Schleifendurchläufe von Schleifen
  - (nur n=1 oder n=2 praktikabel)

# QS: Anforderungsdefinition und Abnahmekriterien

58

- ▶ Abnahmekriterien bereits während der Anforderungsdefinition aus den Qualitätszielen und den Anforderungen ableiten (im 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 mit Testfällen
  - Äquivalenzklassenbildung
  - Grenzwertanalyse
  - Intuitive Testfallermittlung (Ergänzung der o. g. aus Erfahrung)
- ▶ ablaforientierte (White-Box-) Abnahmekriterien: welches Überdeckungskriterium?

## 20.4 Zertifizierung

59

- Prozess-Zertifizierung, z.B. nach ISO 9000, wird im Kap. "Prozessverbesserung" behandelt

# Zertifizierung von Software-Produkten als Wettbewerbsvorteil

60

- ▶ **Grundidee:** Prüfung von **Software** auf bestimmte Eigenschaften (DIN, ISO) durch unabhängige Stellen, danach **Zertifizierung**
  - Wichtig für sicherheitskritische Software in eingebetteten Systemen
    - Auto (Drive-by-Wire)
    - Flugzeug (Fly-by-Wire: Airbus, Boeing)
    - cyber-physikalische Systeme (Life-by-Wire)
  - Überprüfung mittels Checklisten, Verifikation und Validation
- ▶ **Deutschland:** Akkreditierungsstellen **DEKIZ, DATECH, DAKKS**
  - DEKIZ: Deutsche Koordinierungsstelle für IT-Normenkonformitätsprüfung und Zertifizierung“. (seit 1992 = Akkreditierungsverbund mit der BAPT (Post und Telekom)
    - DEKIZ im Jahr 2000 mit DATEch fusioniert ==> <http://www.datech.de>
  - DATEch (Deutsche Akkreditierungsstelle Technik)
    - Im Jahr 2009 fusioniert mit
  - DAKKS: Deutschen Akkreditierungsstelle GmbH (DakKS). <http://www.dakks.de/>
- ▶ **Europa:** **ECITC** („European Committee for Information Technology Certification“)

# Zertifizierung von Software

61

- ▶ **Aufgabe der DEKIZ:**
  - ▶ Koordinierung der nationalen IT&T-Prüf- und Zertifizierungsaktivitäten
  - ▶ Begutachtung und Akkreditierung von Prüfaboratorien und Zertifizierungsstellen
  - ▶ Bestellung von Gutachtern
- ▶ **Ablauf von Prüfung und Zertifizierung:**
  - ▶ Antrag auf Konformitätsprüfung eines Softwarepaketes an ein Prüfaboratorium
    - auf Wunsch wird Prüfmethode, Ort und Zeit bekanntgegeben wegen eigener Vorprüfung und Verbesserung
    - Prüfbericht ist Eigentum des Antragstellers
    - wenn überhaupt, dann Veröffentlichung in vollständiger Form
  - ▶ Erteilung des Zertifikates mit Angabe der Prüfbedingungen

## 20.4.1 Zertifizierungen von Testern

62



Softwareengineering, © Prof. Uwe Alßmann, Technische Universität Dresden, Fakultät Informatik

## ISQF Levels

63

- ▶ Der ISQF ist eine internationale Vereinigung zum Ausbilden und Zertifizieren von Testern.
- ▶ Foundation Level (CTFL)
  - In etwa das, was Sie hier im Master lernen (Vorlesungen Softwaretechnologie-II, Softwarewerkzeuge)
- ▶ Advanced Level – Test Manager
- ▶ Advanced Level – Test Analyst (vormals Functional Tester)
- ▶ Advanced Level – Technical Test Analyst (vormals Technical Tester)
- ▶ Advanced Level (CTAL) – Full Advanced Level (nach Bestehen der o. a. Teilprüfungen Advanced Level)
- ▶ Expert Level – in Vorbereitung



## 20.4.2 Zertifizierungen von Softwarearchitekten



Softwareengineering, © Prof. Uwe Aßmann, Technische Universität Dresden, Fakultät Informatik

## iSAQB Levels



- ▶ Der **International Software Architecture Qualification Board (iSAQB)** ist eine internationale Vereinigung zum Ausbilden und Zertifizieren von Softwarearchitekten
  - <http://www.isaqb.org/>
- ▶ **Foundation Level (Certified Professional for Software Architecture, CP-SA-F)**
  - <http://www.isaqb.org/downloads/pdf/isaqb-Lehrplan-foundation.pdf>
  - In etwa das, was Sie hier im Master lernen (Vorlesungen Design-Patterns and Frameworks, Component-Based Software Engineering, Softwaretechnologie-II)
- ▶ **Advanced Level (CP-SA-F)** [Quelle <http://www.isaqb.org/>]
  - Methodische Kompetenz: Systematisches Vorgehen bei Architekturaufgaben, unabhängig von Technologien
  - Technische Kompetenz: Kenntnis und Anwendung von Technologien zur Lösung von Entwurfsaufgaben
  - Kommunikative Kompetenz: Fähigkeiten zur produktiven Zusammenarbeit mit unterschiedlichen Stakeholdern, Kommunikation, Präsentation, Argumentation, Moderation
- ▶ **Expert Level (CP-SA-E):** in Vorbereitung



# 20.5 Aufgabenmanagement im Qualitätsmanagement (Issue Management)



Softwaremanagement, © Prof. Uwe Aßmann, Technische Universität Dresden, Fakultät Informatik

## Aufgabenmanagement



- ▶ **Aufgabenmanagement** beinhaltet die Erfassung, Registratur, Speicherung und Verfolgung von
  - Stand der Bearbeitung und Arbeitspaketen und Aktivitäten eines Projektplans
  - Qualitätssicherungsaufgaben
  - Änderungsmanagement (sieht der Kunde):
    - Problem- und Fehlermeldungen (“tickets”) in der korrekativen Wartung
    - Änderungsanforderungen (“change requests”) in der Pflege
    - Anforderungen für neue Funktionalität (“feature requests”)
  - beinhaltet also das kontinuierliche Requirements-Management
- ▶ und nutzt dazu
  - verteilte Eingaben über das Web
  - die Speicherung aller “tickets” und “requests” in einer zentralen Repository/Datenbank

Das Aufgabenmanagement ist heute von zentraler Bedeutung für die Steuerung eines Softwarehauses, denn es hilft, die Qualität der Produkte und Dienstleistungen zu verbessern und ein Stammkundengeschäft aufzubauen



# Die Bedeutung des Aufgabenmanagements

68

- ▶ Das Aufgabenmanagement dient als zentrales Element der Qualitätsmanagements und steuert alle dazu notwendigen Vorgänge:
  - **Bewertung der Änderung:** Nach der Notwendigkeit der Auswirkungen, wobei jede Änderung einer Version zu einer neuen Version führt
  - **Planung und Entscheid des Änderungsvorgehens:** Vorgehen nach einer definierten Änderungsprozedur mit Durchlauf geforderter Zustände(V-Modell).
  - **Einleitung und Überwachung der Änderungsdurchführung:** Darstellung des Änderungsgeschehens in einem Repository. Daraus sollen Änderungsstatistiken auf Abruf generierbar sein.
  - **Abschluß und Auswertung der Änderung:**
    - Alle Änderungen sollen nachvollziehbar und rekonstruierbar sein.
    - Es ist eine Historie zu führen, die alle Änderungsdaten einschließlich eines ausführlichen Kommentars enthält.
  - **Release-Planung und Management:** Aufgabenmanagement erlaubt Entscheidungen, wann neue Funktionalität ins Produkt einzieht

# Aufgabenmanagement und Strategisches Management

69

- ▶ Das Aufgabenmanagement dient ebenfalls als strategisches Instrument:
  - **Analysen auf dem Repository:** Statistiken über Häufigkeit von Tickets
  - **Schlussfolgerungen über die Effizienz** des eigenen Entwickelns und der eigenen Qualitätssicherung
  - **Ständige Verbesserung des QM**
- ▶ Als Vorgehensbaustein „Problem- und Änderungsmanagement“ im V-Modell XT realisiert.

# Aufgabenmanagement-Werkzeuge

70

- ▶ **Issue Management:** betont, dass jedes Ticket für das Softwarehaus eine Aufgabe darstellt (“issue”)
- ▶ **Ticket Management:** betont, dass der Kunde “tickets” schreibt

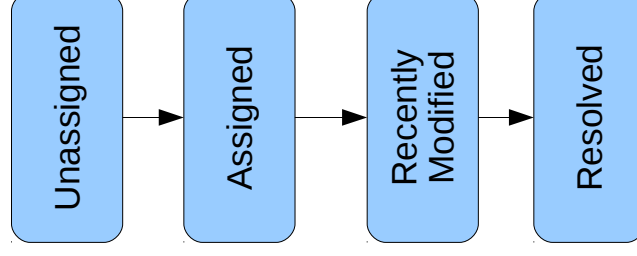
Werkzeug	Status	Webadresse
Bugzilla	Mozilla (OSS)	<a href="http://www.bugzilla.org">www.bugzilla.org</a>
Mantis	OSS	<a href="http://www.mantisbt.org/">http://www.mantisbt.org/</a>
JIRA	Atlassian	<a href="http://atlassian.com/software/jira">http://atlassian.com/software/jira</a>
codeBeamer	IntLand Software	<a href="http://intland.com/products/codebeamer/overview/">http://intland.com/products/codebeamer/overview/</a>
RedMine	OSS	<a href="http://en.wikipedia.org/wiki/Redmine">http://en.wikipedia.org/wiki/Redmine</a>
Team Foundation Server	Microsoft	<a href="http://en.wikipedia.org/wiki/Team_Foundation_Server">http://en.wikipedia.org/wiki/Team_Foundation_Server</a>

[http://en.wikipedia.org/wiki/Comparison\\_of\\_issue\\_tracking\\_systems](http://en.wikipedia.org/wiki/Comparison_of_issue_tracking_systems)

# Aufgabenmanagement (Fehler und Änderungen) mit Mantis

71

- ▶ Mantis ist ein webbasiertes Aufgabenmanagement-System (issue tracking system)
  - Zustandsmaschine für Fehler und Änderungswünsche
  - Generierung von emails über Statusänderungen
  - Visualisierung von Zuständen und Bearbeitern





# Überblick über Aufgaben [www.mantisbt.org]

72

View Issues - MantisBT DEMO site

Project: Demo [Switch] [Jump]

Issue # [Apply Filter] [Advanced Filters] [Create Permalink]

P.	ID	USER	#	Category	Severity	Status	Updated
	0005069		3	GUI	minor	resolved (MacGyver)	2009-01-28
	0005095			Other	minor	assigned (mow)	2009-01-28
	0005091			Other	trivial	confirmed (ossegawat)	2009-01-28
	0004702			Other	minor	resolved (000willberry)	2009-01-28
	0005093			GUI	minor	assigned (enzyme)	2009-01-28
	0005082			Other	minor	assigned (121212)	2009-01-28
	0005094			GUI	minor	new	2009-01-28
	0005092			GUI	minor	new	2009-01-28
	0005090			GUI	minor	assigned (Händersen)	2009-01-28
	0005089		3	Website	major	resolved (Russell)	2009-01-28
	0005076		1	GUI	feature	assigned (drksaboteur)	2009-01-28
	0005088			GUI	tweak	assigned (abarbosa)	2009-01-27
	0005087			GUI	trivial	assigned (ramyap)	2009-01-27
	0005086		2	GUI	trivial	assigned (Element)	2009-01-27
	0005083		1	Other	major	acknowledged (saloc)	2009-01-27
	0005085			GUI	major	assigned (celso)	2009-01-27
	0005081			GUI	minor	resolved (deepak64)	2009-01-27
	0005080			GUI	feature	assigned (mmiat)	2009-01-27
	0005079			GUI	minor	assigned (kedar)	2009-01-26
	0005078		4	GUI	text	assigned (jodj)	2009-01-26
	0005077			GUI	minor	new	2009-01-26
	0005065		3	GUI	tweak	new	2009-01-26
	0005074		2	Other	minor	assigned (maksadmin)	2009-01-26
	0005075			Website	major	assigned (patrioalomes)	2009-01-26
	0005071		2	Website	major	assigned (deepak64)	2009-01-26



# Überblick über Zustände der Aufgaben

73

View Issues - MantisBT DEMO site

Project: Demo [Switch] [Jump]

Issue # [Apply Filter] [Advanced Filters] [Create Permalink]

P.	ID	USER	#	Category	Severity	Status	Updated
	0005094			GUI	text	Test fehler attache	2009-01-28
	0005092			GUI	text	Test fehler attache	2009-01-28
	0005091			Website	major	Needs attention	2009-01-28
	0005076			GUI	feature	Problems loading JavaScript on Main.html	2009-01-28
	0005088			GUI	tweak	I can't pob...	2009-01-27
	0005087			GUI	trivial	sadasdasda	2009-01-27
	0005086			GUI	trivial	sadasdasda	2009-01-27
	0005083			Other	major	Detaillbereich wird nicht angezeigt	2009-01-27
	0005085			GUI	major	teste com erro e associacao com arquivo de repositorio	2009-01-27
	0005081			GUI	minor	resolved (deepak64)	2009-01-27
	0005080			GUI	feature	???	2009-01-27
	0005079			GUI	minor	test	2009-01-26
	0005078			GUI	text	test	2009-01-26
	0005077			GUI	minor	cant abc	2009-01-26
	0005065			GUI	tweak	hhhh	2009-01-26
	0005074			Other	minor	VZ funktioniert nicht	2009-01-26
	0005075			Website	major	Erro no site	2009-01-26
	0005071			Website	major	Website Issue	2009-01-26



## 20.5.2 Änderungsmanagement

74

- Änderungsmanagement ist der Teil vom Aufgabenmanagement, den der Kunde sieht
- Also ein spezielles Aufgabenmanagement in der Anforderungsmanagement, Wartung, Pflege und Releasemanagement

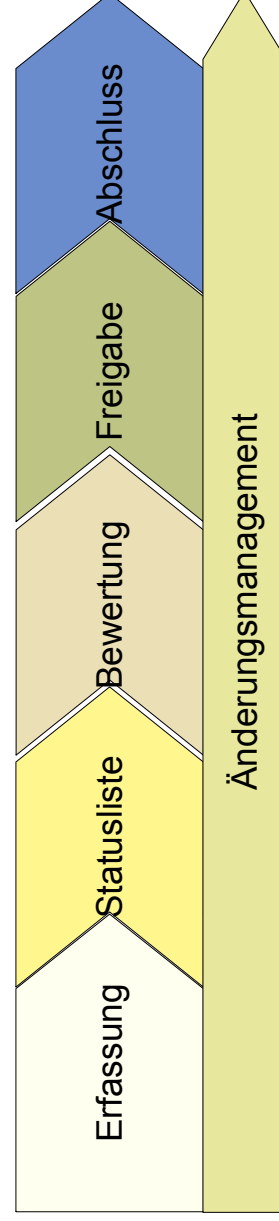


Softwaremanagement, © Prof. Uwe Alßmann

## Aufgaben der Aktivitäten des Problem- und Änderungsmanagements laut V-Modell XT

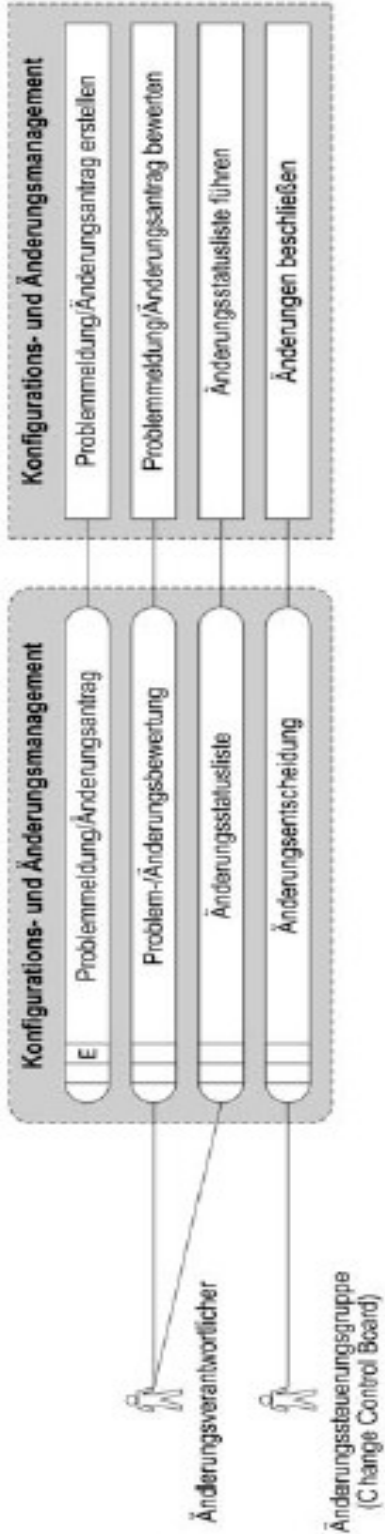
75

- ▶ **Zustandserfassung** von Problemmeldungen/Änderungsanträgen (korrektive Wartung, Pflege)
- ▶ Dokumentieren und Verwaltung aller Problemmeldungen und Änderungsanträge über eine **Statusliste**
- ▶ Änderungen **bewerten** (Ursachen, Auswirkungen, ...)
- ▶ Entscheidung, **Freigabe** und Veranlassung der Bearbeitung
- ▶ **Abschluss** der Änderung, Information der Betroffenen
- ▶ Erfassung von Problemmeldungen, Fehlermeldungen, Verbesserungsvorschlägen und Änderungswünschen



# Vorgehensbaustein Problem- und Änderungsmanagement

76



Notwendige Artefakte (Produkte, Belege) sind:

- Problemmeldung und Änderungsantrag
- Problem- und Änderungsbewertung
- Änderungsentscheidung, -mitteilung
- Änderungsstatusliste

Sie werden in den zugehörigen Aktivitäten des V-Modells XT bearbeitet.

# Änderungsmanagement

77

- ▶ ist nötig für Firmen, die sukzessive neue Versionen ihrer Produkte erzeugen
  - im Produktgeschäft tätig sind
  - im Produktlinien-Geschäft tätig sind
- ▶ weniger nötig für eine Anwendungslandschaft in einer Firma

## Aktivität

# Problemmeldung/Änderungsantrag erstellen

79

- ▶ Jede Rolle kann aus den verschiedensten Gründen eine Problemmeldung/**Änderungsantrag** auslösen, der grundsätzlich folgende Informationen enthält:
  - Beschreibung des Problems bzw. der gewünschten Änderung
  - Identifikation Antragsteller, Projekt, betroffenen Konfiguration
  - Begründung des Antrages bzgl. Nutzen bzw. Schaden bei Nichtdurchführung
  - Lösungsvorschlag aus Sicht des Antragstellers
  - Nummer Änderungsantrag/Problemmeldung
  - Vergabe einer Registriernummer pro Problemmeldung/Änderungsantrag
- ▶ Gründe für Änderungen können sein:
  - neue Entwicklungserfordernisse
  - Probleme im Controlling: Zeitprobleme, Kosteneinhaltung
  - Compliance: Änderungen gesetzlicher Vorschriften
    - Sarbanes-Oxley Act, Basel-II
  - Verbesserung von Marktchancen
    - Erscheinen neuer Plattformen
  - Änderungen in den Anforderungen: Nutzerwünsche

## Aktivität Problemmeldung/Änderungsantrag bewerten

80

- ▶ Problemmeldung/Änderungsantrag analysieren wie dringend Lösung des Problems bzw. der beantragten Änderung ist
- ▶ Lösungsvorschläge erarbeiten mit vollständiger bzw. auch erst nur teilweiser Lösung. Folgende Informationen sollte er enthalten:
  - Teile des Projektes, die von der Änderung betroffen sind
  - Phase des Entwicklungsprozesses, in der Änderung anfällt
  - Lösungsbeschreibung und -vorgehen
  - erforderliche Aufwendungen
  - Auswirkungen der Änderung auf das Projekt
- ▶ Empfehlung aussprechen:
  - auf Basis der erarbeiteten alternativen Lösungsvorschläge
  - alle Lösungsvorschläge sind anhand ihrer Auswirkungen auf das Projekt zu bewerten
  - aus dieser Basis ist eine Entscheidung zu fällen und zu begründen
- ▶ Alle Bewertungsaktivitäten werden im Produkt Problem-/Änderungsbewertung niedergelegt

## Aktivität

# Änderungen beschließen

81

- ▶ Vorbereitung des Entscheidungsmeetings durch Sammeln aller Anträge und Bewertungen, **Erstellen der Agenda** für das Meeting
- ▶ Einladungen an beteiligte Rollen oder Stakeholder **verschicken**
- ▶ Anträge vorstellen und **präsentieren** mit:
  - entstehenden Kosten
  - Verfügbarkeit von Mitteln und Personal
  - zeitliche Projektverzögerung
  - technische Eignung der vorgeschlagenen
- ▶ Änderungsentscheidung **beschließen** und Dringlichkeit der Umsetzung festlegen
  - Festlegung der Kategorie (Fehler [in Spezifikation, Entwurf, Codierung, im Verfahren], Problem, Modifikation, Erweiterung, Verbesserung, usw.)
  - gewünschter Fertigstellungszeitpunkt
- ▶ Auswirkungen der Änderung ermitteln
- ▶ Änderungsentscheidung – im Änderungsbescheid - **protokollieren**
- ▶ Änderungsentscheidung **verteilen** bzw. kommunizieren
- ▶ Alle beschlossenen Änderungen werden im Produkt Änderungsentscheidung/-mitteilung niedergelegt

## Aktivität

# Änderungsstatusliste führen

82

- ▶ Änderungsstatusliste dient dem Ziel, alle wichtigen Informationen zum Projekt hinsichtlich Änderungsanforderungen und –auswirkungen zu aktualisieren und dokumentieren
- ▶ Ablauf und Dokumentation ist für jede Änderungsanforderung gleich:
  - **Änderungsanforderungen registrieren** mit Prüfung der benötigten Daten auf Vollständigkeit
  - **Änderungsanforderungen prüfen** auf Realisierbarkeit und Festlegung der erforderlichen Mittel, Termine und Verantwortlichkeiten
  - **Änderungsstatusliste aktualisieren** nach bereits bestehenden Änderungsanforderungen bzw. durch Hinzufügen neuer Anforderungen
- ▶ Änderungsstatus sind z. B.: ‚beantragt‘, ‚beabsichtigt‘, ‚abgelehnt‘, ‚genehmigt‘, ‚zurückgestellt‘, ‚beauftragt‘, ‚erledigt‘
- ▶ Bemerkungen bei Beziehungen zu bereits gestellten Änderungsanträgen
- ▶ Referenzen auf die Änderungsbewertung oder die Änderungsentscheidung sind in der Änderungsstatusliste ebenfalls festzuhalten
- ▶ Wird oft in einer Datenbank geführt
  - z.B. MANTIS-System

## 20.6 Kundenmanagement



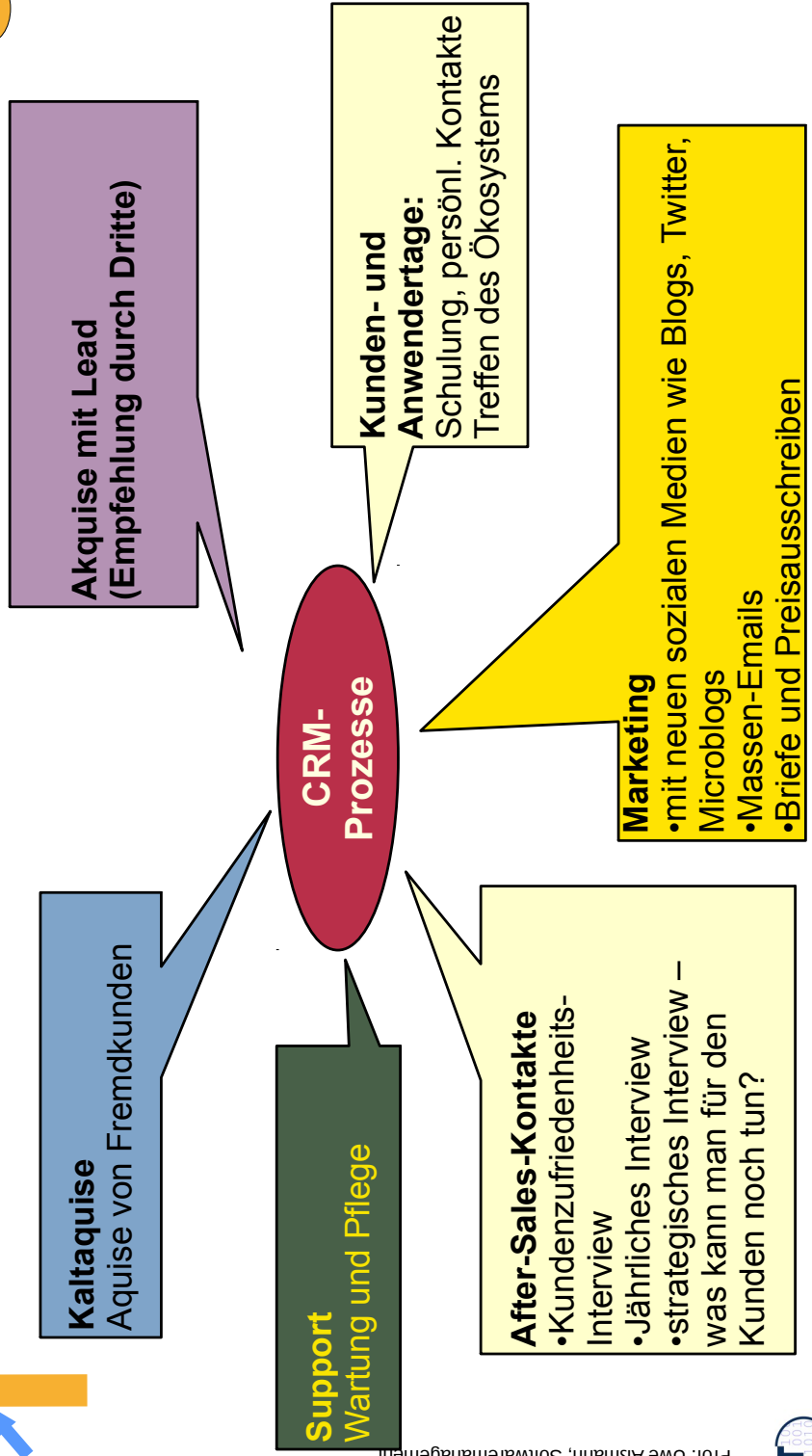
## Kundenmanagement



Das **Kundenmanagement (Customer relationship management, CRM)** verwaltet die Kontaktdaten und Geschichte der Interaktionen mit Kunden (Dokumente, Verträge, emails, Telefonnotizen, Kontaktnotizen). Ziel ist es, kundenzentriert alle Daten auf einen Blick, an einem Platz zu sammeln und für die ganze Firma zugreifbar zu halten.

- ▶ Besonders wichtig bei
  - **Stammkunden**, denn man darf keine Fehler mit ihnen machen
  - einem **Massenmarkt** mit Tausenden von Kunden
- ▶ Ein gutes CRM ist eine der wichtigsten Prozesse einer Softwarefirma, weil das Feedback des Kunden die wichtigste Voraussetzung zum Halten von Stammkunden ist

# CRM-Prozesse



# CRM-Systeme

- ▶ Ein CRM-System ist ein Informationssystem, das folgende Aufgaben erfüllt:
  - Kontakt-Datenbank mit Geschichte der Kontakte und Interaktionen
  - Unterstützung aller CRM-Prozesse (s.o.)
  - Schnittstellen zu anderen Systemen, wie dem Dokumentenmanagementsystem, dem ERP-System, dem Aufgabenmanagement
- ▶ Die Auswahl von passenden CRM-Systemen wird als schwierig eingeschätzt, ist aber unerlässlich für ein gutes Kundenmanagement und den Erfolg der Firma

Firma

Beispiele			
vtiger	OSS	<a href="https://www.vtiger.com">https://www.vtiger.com</a>	
Dynamics	Microsoft	<a href="http://www.microsoft.com/dynamics/">http://www.microsoft.com/dynamics/</a>	
salesforce	Salesforce	<a href="http://www.salesforce.com/">http://www.salesforce.com/</a>	Cloud-basiert
SugarCRM	dual licensing	<a href="http://www.sugarcrm.com/">http://www.sugarcrm.com/</a>	

- ▶ [http://en.wikipedia.org/wiki/CRM\\_software](http://en.wikipedia.org/wiki/CRM_software)
- ▶ [http://en.wikipedia.org/wiki/Comparison\\_of\\_CRM\\_systems](http://en.wikipedia.org/wiki/Comparison_of_CRM_systems)



The End



87

