

Fakultät Informatik

Professur Softwaretechnologie

SOFTWAREMANAGEMENT

20_QUALITÄTSSICHERUNG

Prof. Dr. Uwe Aßmann
Dr.-Ing. Birgit Demuth
Sommersemester 2017

Überblick Qualitätssicherung (QS)

- Einführung
 - Standards zur QS
 - Warum man Qualitätssicherung und -management braucht
 - Qualitätsbegriff
- Konstruktive QS im Prozess (Qualitätslenkung)
 - Technische Maßnahmen
 - Organisatorische Maßnahmen
- Analytische QS (Qualitätsprüfung)
 - Prozessgestützte analytische QS
 - Werkzeuggestützte analytische QS
 - Testende QS-Verfahren
- Aufgabenmanagement im Qualitätsmanagement (Issue Management)
- Kundenmanagement

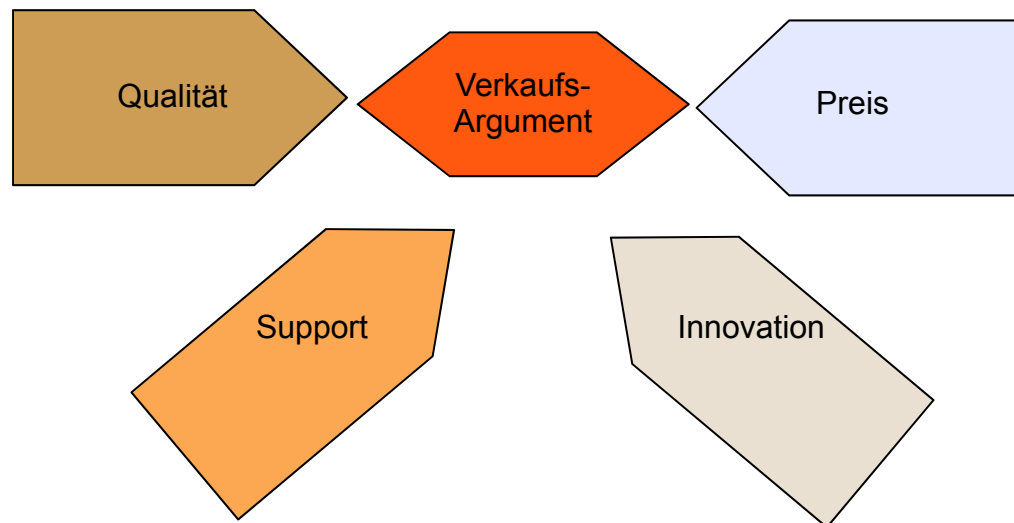
Einführung

Standards zur QS

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 Software-Test-Dokumentation
DIN 66270	Bewerten von Softwaredokumenten, Qualitätsmerkmale
ANSI/IEE 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

Warum man Qualitätssicherung und -management braucht

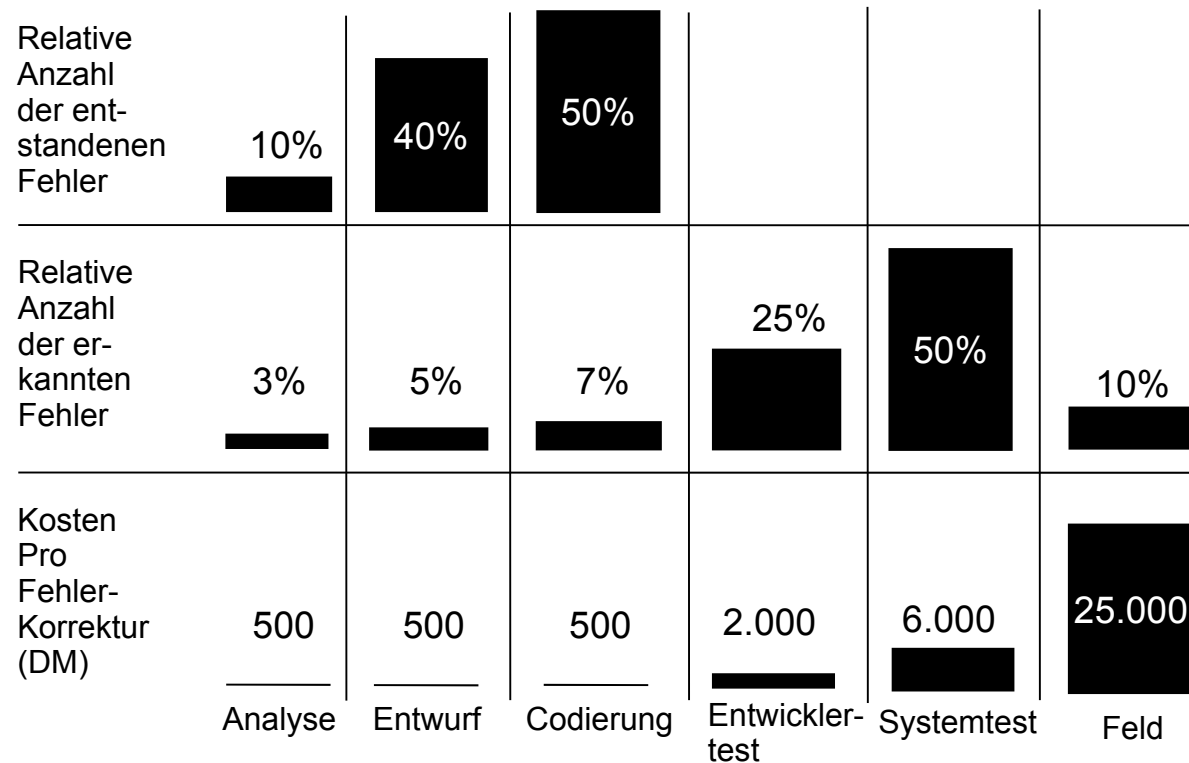
95% aller Dresdner Softwarefirmen verkaufen über die Qualität.
[KMU-Studie]



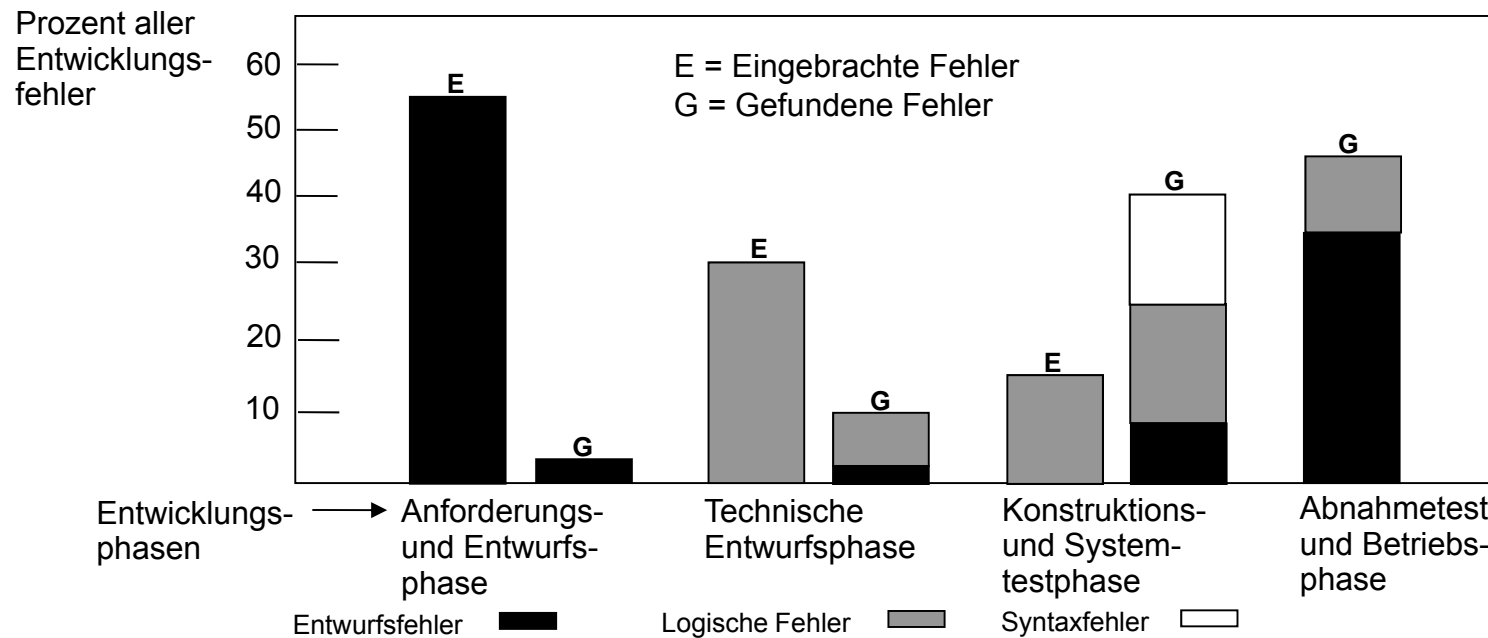
Der Gegner: Fehleranzahl und -kosten

(Empirische Daten)

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



Fehlerbeseitigungskosten



55% aller Fehler entstehen in der Anforderungs- und Entwurfsphase

Quelle: [BalzertSM]

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

Total Quality Management (TQM) (Ständige Qualitätsverbesserung)

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

- Jeder **unzufriedene Kunde** besitzt keine Kundenbindung und wird kein Stammkunde werden
- **Beschwerde-(Issue-)Management** heißt, aus einem unzufriedenen Kunden einen zufriedenen Kunden zu machen und Kundenbindung zu erzeugen

Nur ein zufriedener Kunde bleibt auch ein Kunde

Ziel: Kundenzufriedenheit in CoTiQQ (cost, time quality, quantity)
Abgeleitet von PDCA

Qualitätsbegriff

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]

Qualität

- des Produkts (Produktqualität)
- des Entwicklungsprozesses (Prozessqualität)
- der Beteiligten (Personenqualität)

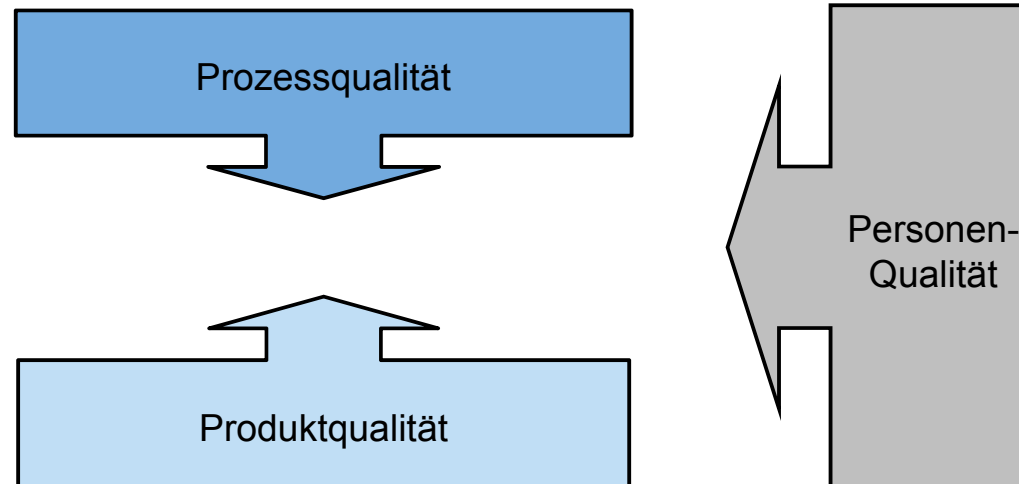
Qualitätsmerkmale:

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



Faktoren zur Erringung von Qualität

- Maßnahmen zur Prozessqualität werden zuerst eingeführt, da am einfachsten
- Danach Maßnahmen zur Produktqualität
- Maßnahmen zur Personenqualität sind anfangs und mittelfristig sehr wichtig; kurzfristig lässt sich nichts ändern



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

Merkmal

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

Teilmerkmale

Richtigkeit (Korrektheit)
Angemessenheit
Interoperabilität
Ordnungsmäßigkeit (governance)
Sicherheit
Reife
Wiederherstellbarkeit

Verständlichkeit
Erlernbarkeit
Bedienbarkeit
Zeitverhalten
Verbrauchsverhalten
Analysierbarkeit
Modifizierbarkeit
Prüfbarkeit
Stabilität
Anpassbarkeit
Installierbarkeit
Austauschbarkeit,
Konformität (gegenüber Normen)

Merkmalsbeschreibung

Fähigkeit des Systems die
geforderten Anforderungen
zu erfüllen

Einhaltung eines Leistungsniveau
unter festgelegten Bedingungen
über einen definierten Zeitraum

Aufwand zur Benutzung der
Software durch unterschiedliche
Benutzergruppen

Benötigte Zeit und Verbrauch von
Betriebsmitteln für Aufgabe

Maß für Möglichkeit der
Modifizierung von Software auf
Basis interner und externer
Einflüsse

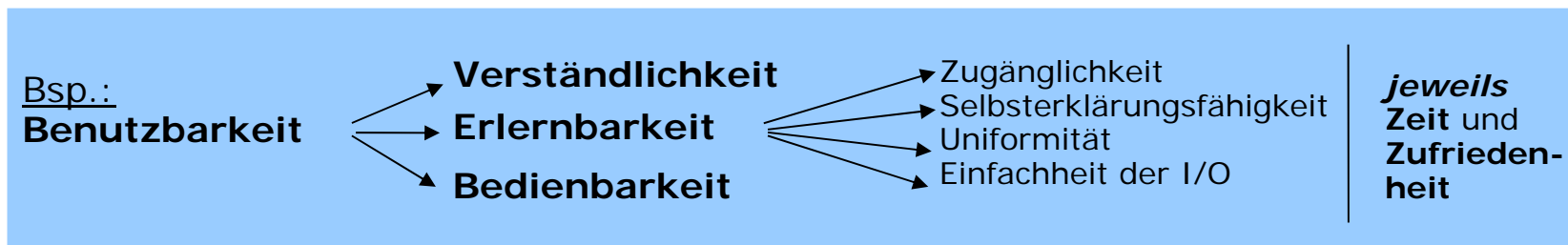
Maß für Offenheit und
Portabilität von Software zur
Lauffähigkeit auf anderen
Soft- und Hardwaresystemen

Qualitätsmerkmale ISO/IEC 25000

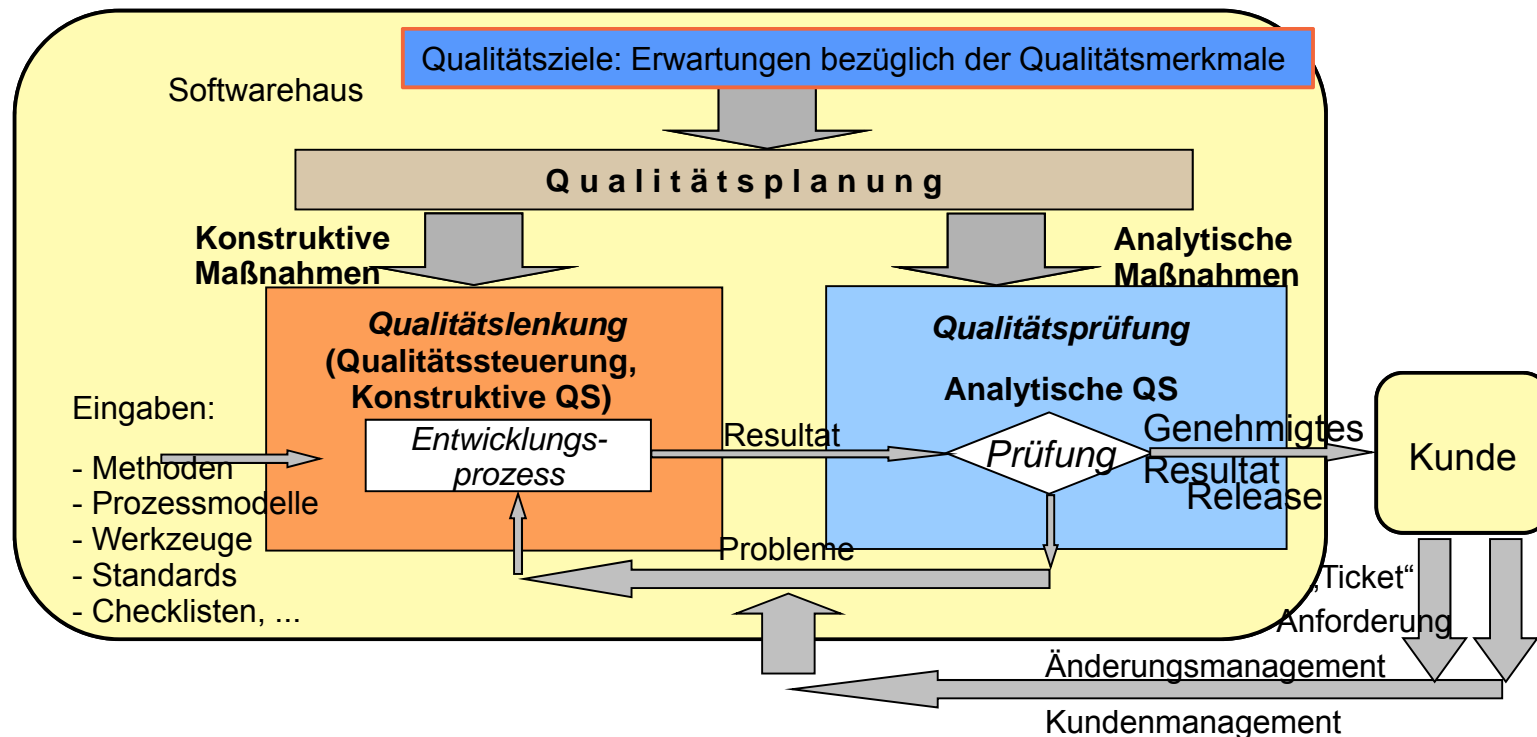
Merkmale

Teilmerkmale

Elementar- merkmale



Qualitätsmanagement (QM)-System (im Application Lifecycle Management ALM)



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

Qualitätsprüfung (-sicherung) (CHECK)

- Auswerten qualitativer und quantitativer Qualitätsinformationen, z.B. Tickets, Feature request, Kosten-Nutzen-Betrachtungen (Wirtschaftlichkeitsbetrachtungen), Releaseplanung

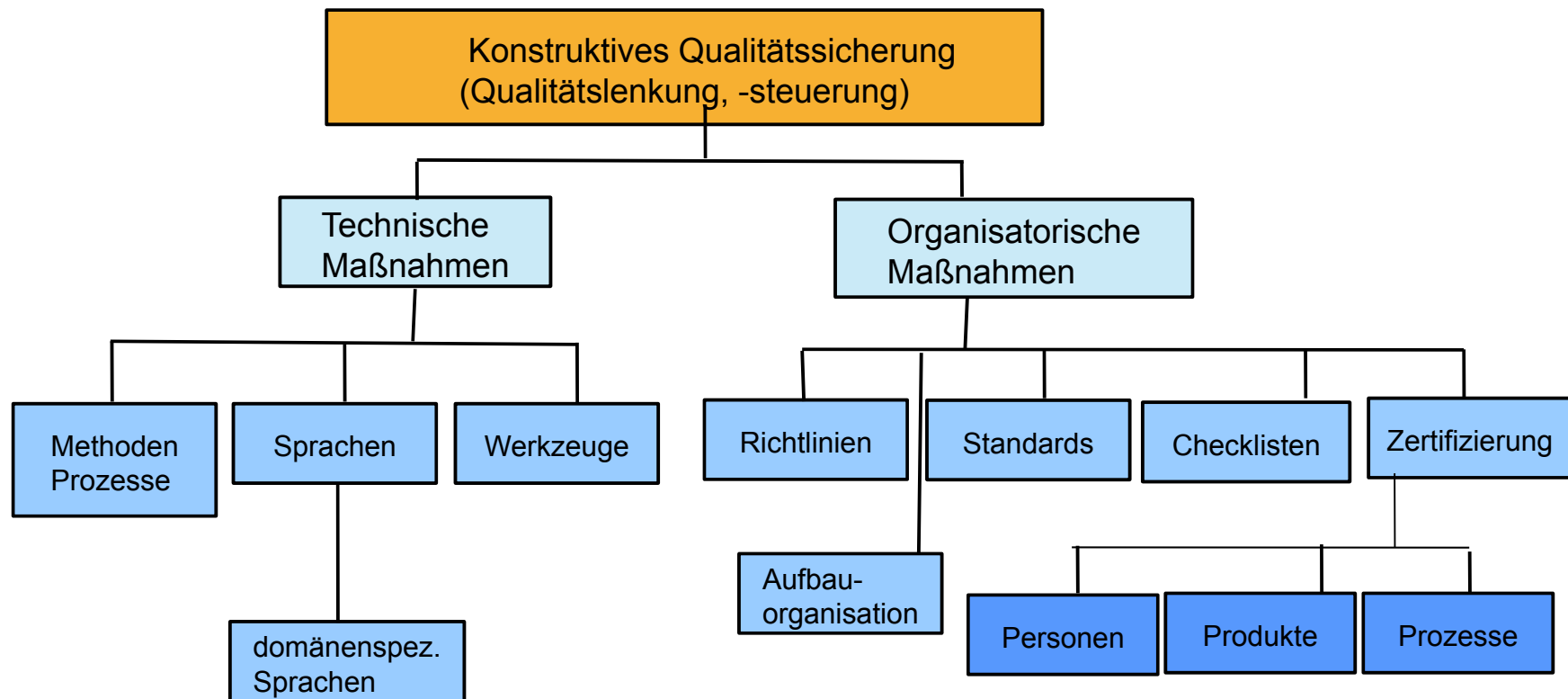
Qualitätsverbesserung (ACT)

- Maßnahmen umsetzen zur Steigerung der Produktqualität und Prozessoptimierung, Erfolge und Ergebnisse werden kommuniziert, Fehlerkorrektur, Einführung von neuen Funktionen

Aufbauorganisation der QS

- QS durch externe Unternehmen (Berater, Auditoren)
- QS durch eigenständige Abt. im Org./DV-Bereich
- Projektmanagement-Büro oder Stabsstelle
- QS durch Kompetenzteams

Konstruktive QS im Prozess (Qualitätslenkung)



Beispiele für technische Maßnahmen zur konstruktiven QS (1)

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
- Einsatz von **domänenspezifischen Sprachen**, um Fehler zu vermeiden und domänenspezifische Constraints durchzusetzen
- **Programmiersprache** mit strengem Typkonzept, um auch zur Laufzeit Typprüfungen vornehmen zu können.

Beispiele für technische Maßnahmen zur konstruktiven QS (2)

Werkzeuge

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

Beispiele für organisatorische Maßnahmen zur konstruktiven QS

Richtlinien

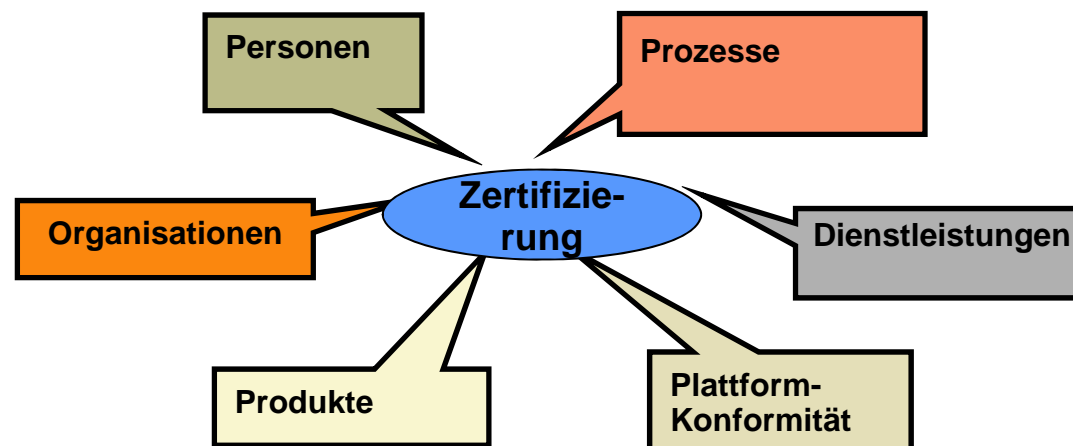
- Definition von Entwicklungsprozessen in einer **Prozesslandkarte (process map)**
- Projektbegleitende **Dokumentationsfortschreibung** möglichst nach einem Standard und werkzeuggestützt
- Dokumentenmuster für **Pflichtenheft**, dass eine sichere Erfassung aller Anforderungen gewährleistet.
- Software-**Konfigurationsmanagement** 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
- der Dienstleistungen

Zertifizierung

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]



Zertifizierung von Software-Produkten als Wettbewerbsvorteil

- **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:** Mutter aller Akkreditierungsstellen **DAKKS**
 - Deutschen Akkreditierungsstelle GmbH (DAKKS) <http://www.dakks.de/>
[http://www.dakks.de/doc_ze-geraete]
- **Europa: ECITC** (European Committee for Information Technology Certification)

Zertifizierung von Software-Produkten

Aufgabe der DAKKS:

- Neuzulassung von Zertifizierungsstellen für Produkt-, Personen- und Prozessakkreditierung
- Koordinierung der nationalen IT&T-Prüf- und Zertifizierungsaktivitäten
- Begutachtung und Akkreditierung von Prüflaboratorien und Zertifizierungsstellen
- Bestellung von Gutachtern

Ablauf von Prüfung und Zertifizierung:

- Antrag auf Konformitätsprüfung eines Softwarepaketes an ein Prüflaboratorium
 - 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

Zertifizierung in Software-Ökosystemen

Ein **Software-Ökosystem** 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

Zertifizierung von Personen (1)

- Prozess-Zertifizierung, z.B. nach ISO 9000, wird im Kap. "Prozessverbesserung" behandelt
- CERT-IT <http://www.cert-it.com/>
- Zertifizierungsstelle für die deutsche IT- und Bildungsbranche, anerkannt von der Deutschen Akkreditierungsstelle (DAKKS)
- Gegründet von BITKOM, ZVEI, IG Metall, ver.di, Gesellschaft für Informatik, Fraunhofer Gesellschaft.
- <http://www.cert-it.com/personen-zertifizierung/certified-tester/literatur/>

Zertifizierung von Personen (2)

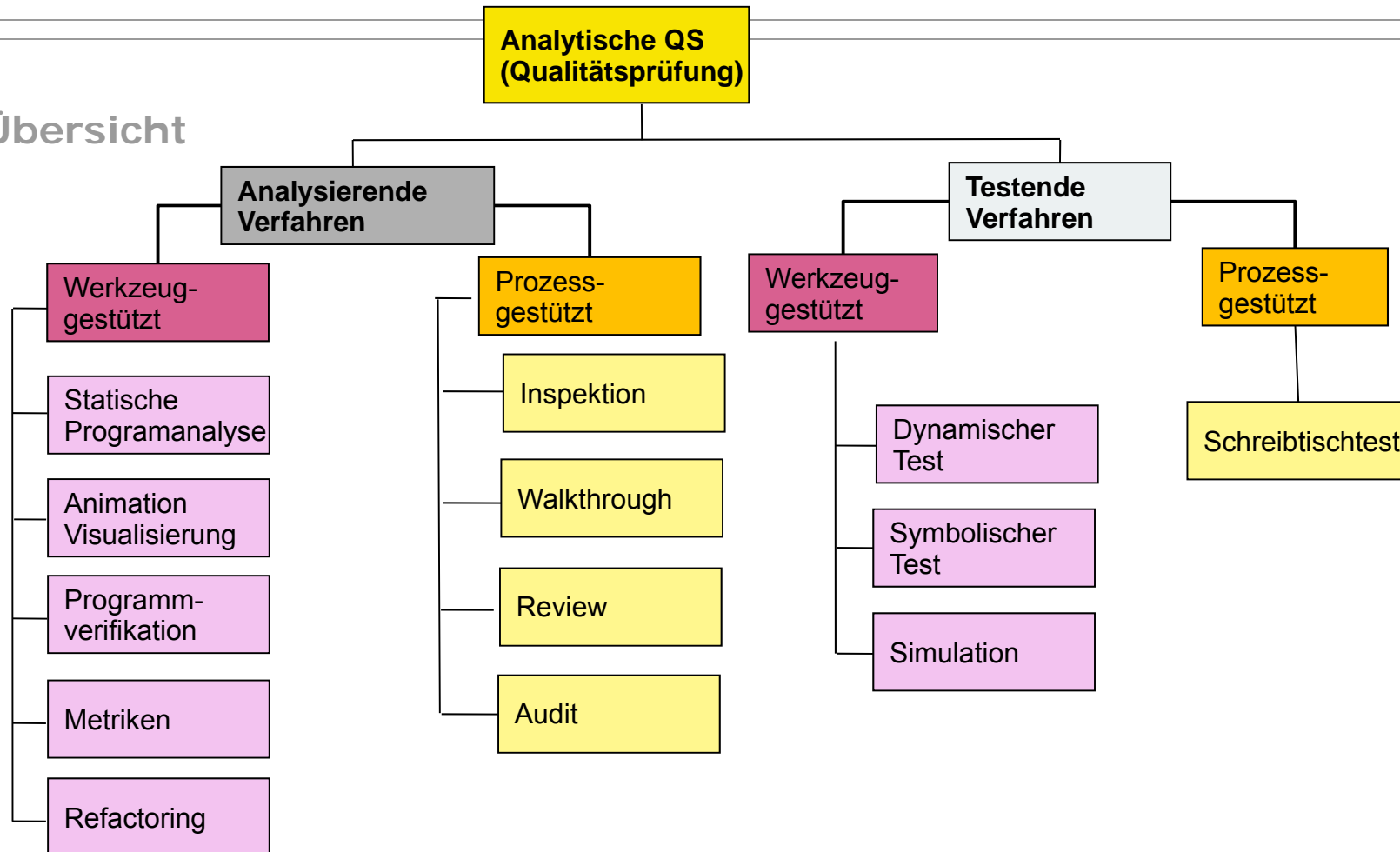
- Nachweis von Ausbildungsstandards oder besonders ausgearbeiteten Fachnormen bei Personenzertifizierungen.
- Zertifizierung der Mitarbeiter zur Dokumentation von Fähigkeiten, Qualifikation und Kompetenz. Siehe dazu Liste der IT-Zertifikate.
 - PMP (Project Management Professional) durch das PMI (Project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

Analytische QS (Qualitätsprüfung)

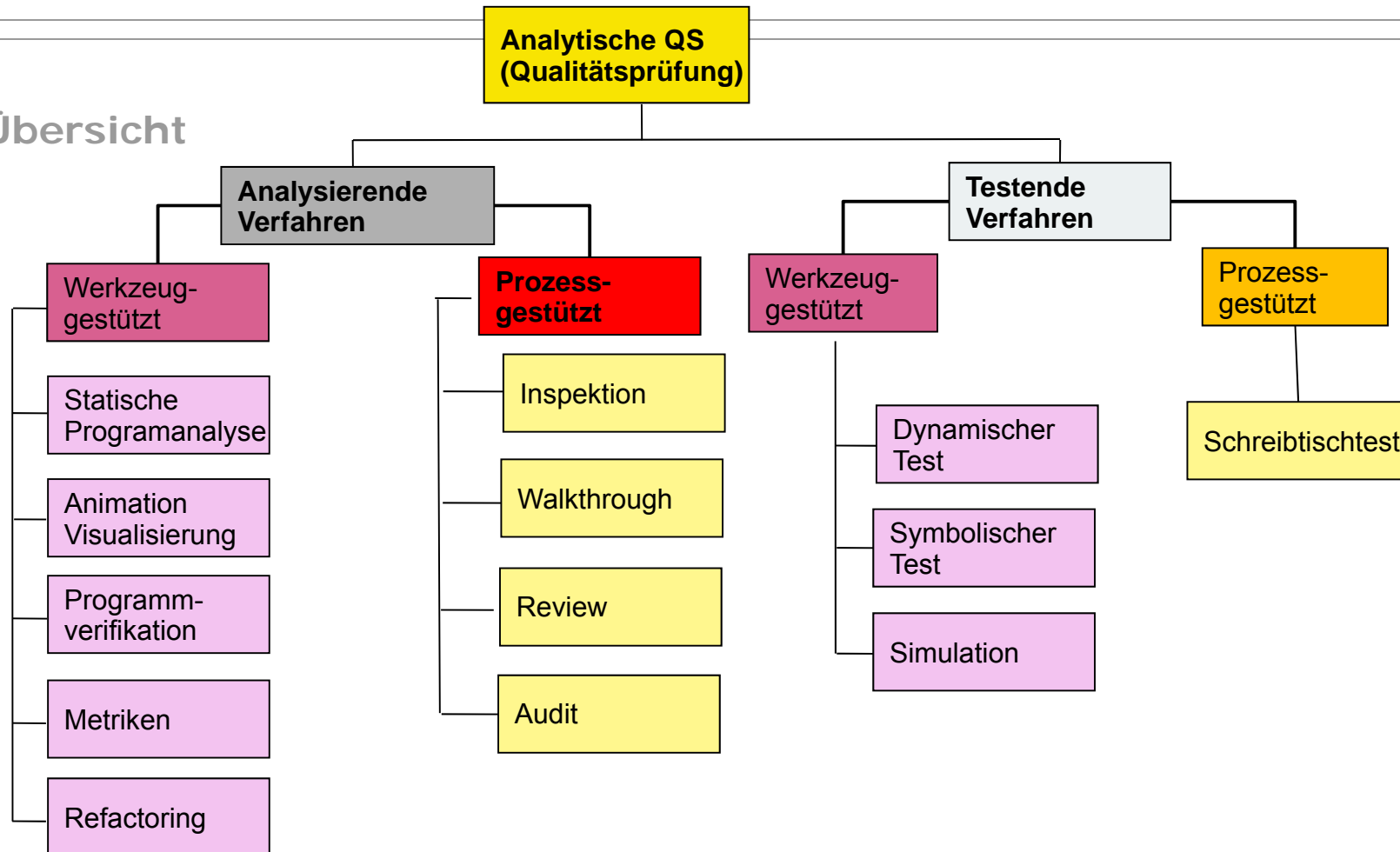
Qualitätsprüfung analysiert die Qualität von Produkt und Prozess und versucht, Verbesserungen vorzuschlagen

- Analyse
- Test

Übersicht



Übersicht



Review

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.)

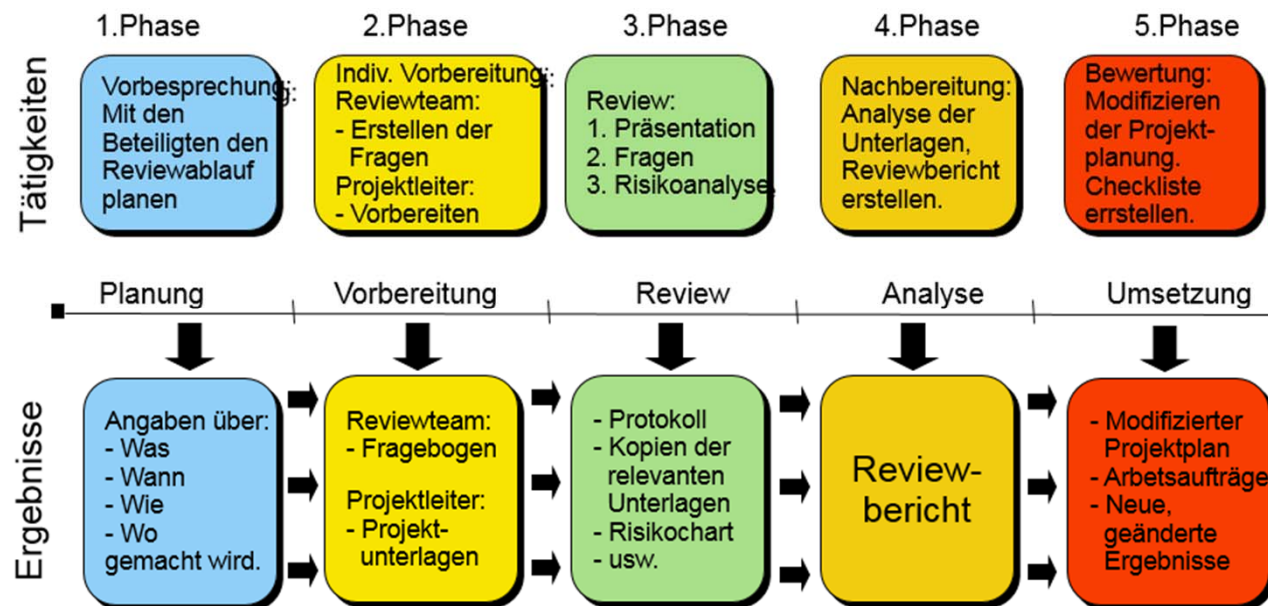
Reviews fokussieren sich auf Produktqualität

- Endprodukt (PBS)
- Zwischenprodukte (Modelle, Dokumentation, Testsuiten)

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

Projektreview-Ablauf

Der Reviewablauf besteht aus fünf Phasen:



Arten von Reviews (1)

- **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

Arten von Reviews (2)

Tools

http://wiki.eclipse.org/Reviews/R4E/User_Guide

<http://www.agilereview.org/wp-content/uploads/2012/04/quickstart.pdf>

➤ **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“)

Audits (1) – Formale Reviews von Außen

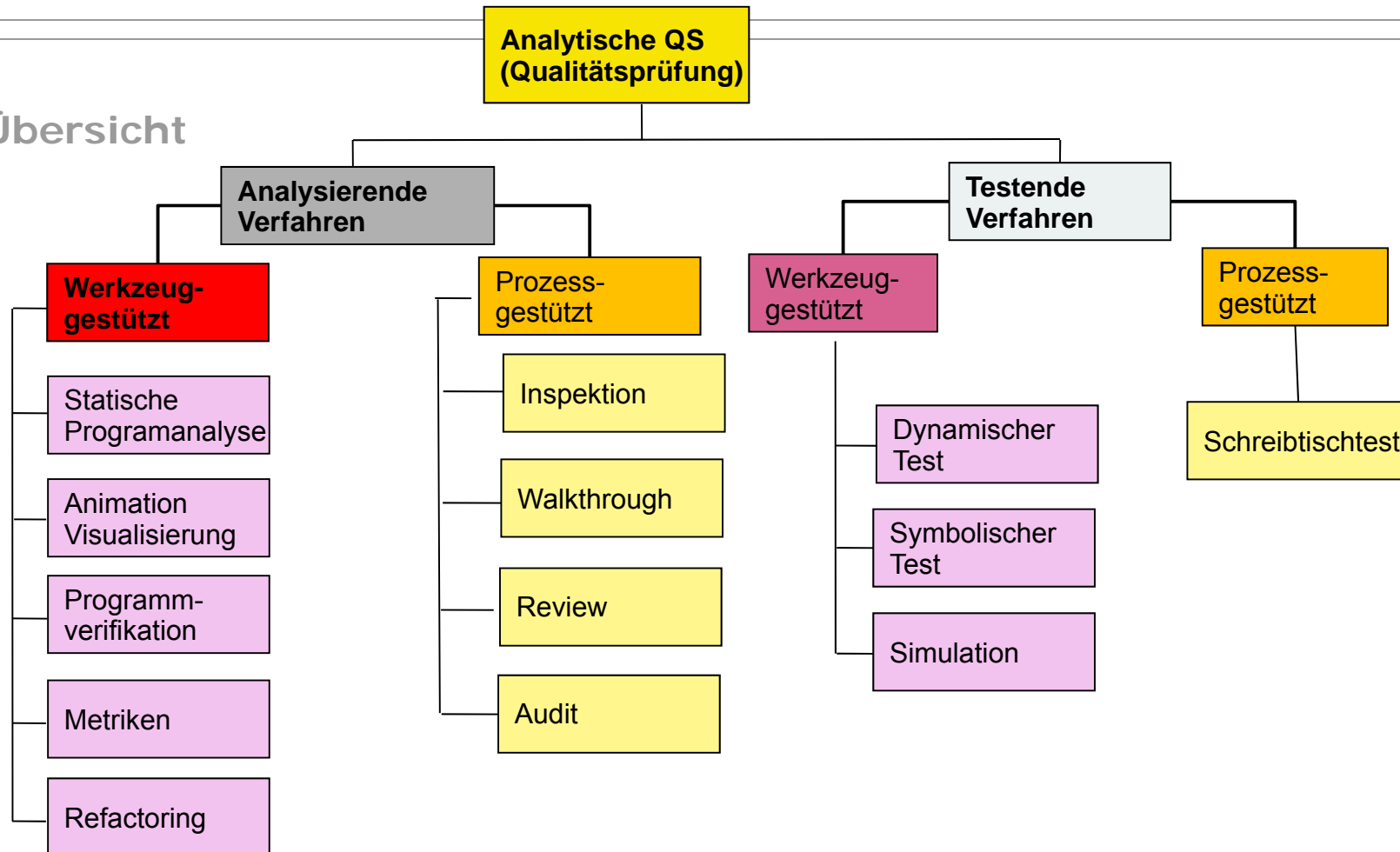
Ein **Audit** ist eine *systematische* und *unabhängige* Untersuchung, bei der sowohl die Übereinstimmung mit Spezifikationen, Standards, vertraglichen Vereinbarungen oder anderer Kriterien (Angemessenheit, Einhaltung vorgegebener Vorgehensweisen und Anweisungen), als auch deren Wirksamkeit und Sinnhaftigkeit 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 entsprechend den Anforderungen vollständig, dokumentiert und wirksam sind.
- Audit des Finanzmanagements
- Audit des Entwicklungs- und Managementprozesses, z. B.: Produktivität des Projektteams , Einhaltung vorgegebener Standards

Audits (2) – Formale Reviews von Außen

- 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, z.B. für Basel I-III Kriterien, Prüfung auf Gemeinnützigkeit

Übersicht



Metrik-Werkzeuge

Ein Analysewerkzeug nutzt eine **Softwaremetrik** zur statischen Auswertung von Komplexitätsattributen von Code und Modellen.

Beispiele:

- Coupling Between Object Classes (CBO): number of neighbored, i.e., coupled classes
- Depth of inheritance hierarchy (DIT)
- Breadth of inheritance hierarchy (BIT) or Number of Children (NOC)
- Number of Methods (NOM)
- Lines Of Code (LOC) in a class
- McCabe metric (cyclomatic complexity)
- Object Points

Statische Programmanalyse (1)

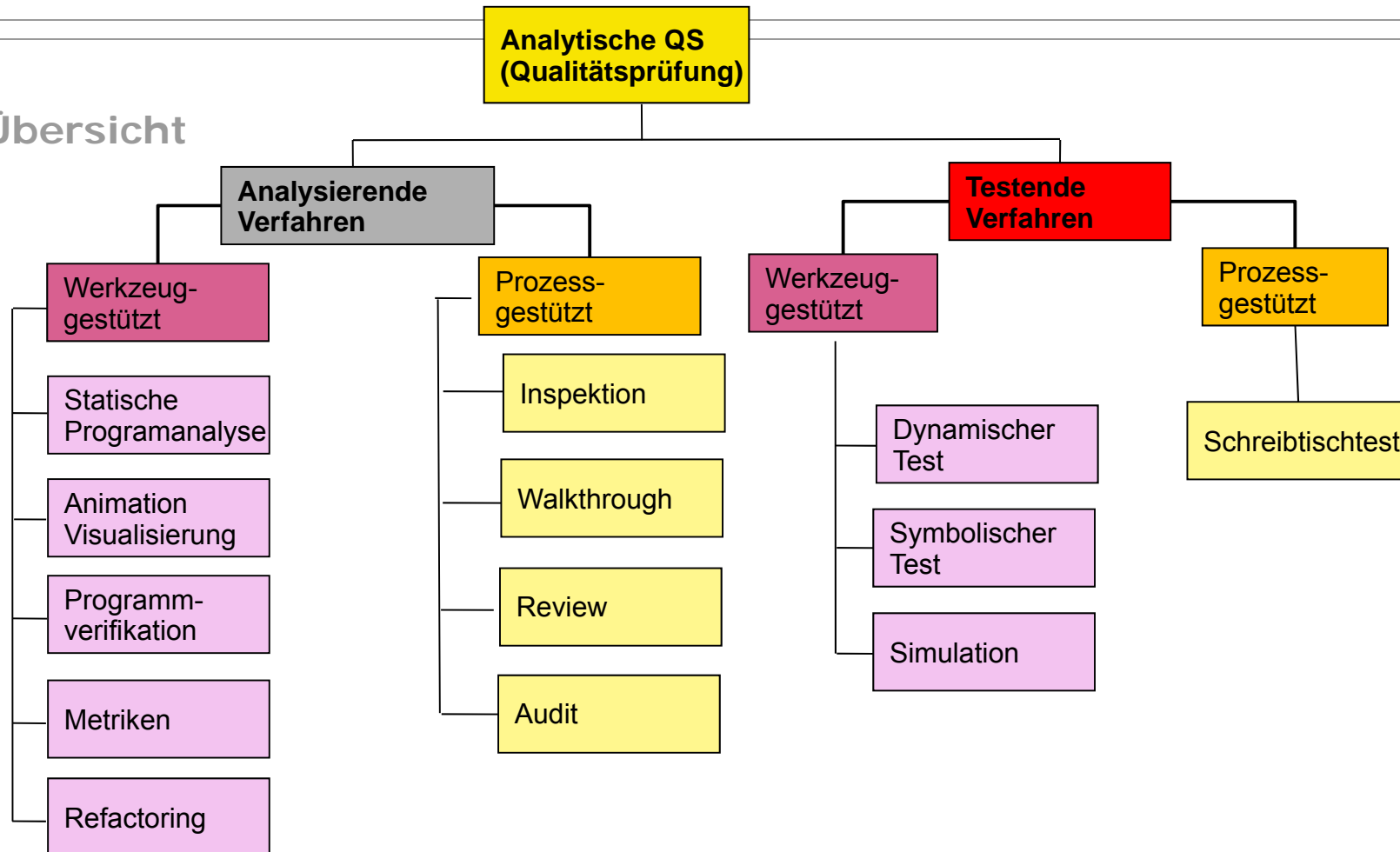
Werkzeuggestützt!

- **Lexikalische Analyse** ermittelt lexikalische Informationen (u.a. dead code)
- **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** durch Analyse des Datenflusses zwischen Variablen
 - **Abstrakte Interpretation** interpretiert das Programm statisch mit abstrakten Werten, die Fehlerwerte entdecken lassen
 - Typprüfungen (wie Typkonflikte, falsche Parameterübergaben)
 - Datenflussunverträglichkeiten wie deklarierte aber nicht verwendete Variablen, nicht initialisierte Variable, falsche Verwendung globaler und lokaler Variablen

Statische Programmanalyse (2)

- **Statische Vertragsprüfung** mit Werkzeugen
 - Theorembeweiser, gute Übersetzer für Programmiersprachen mit Verträgen wie Eiffel
 - Prüfung von Qualitätsverträgen z.B. mit Qualitune (www.qualitune.org)
- **Model checking** prüft die Gültigkeit von Prädikaten in einem Zustandssystem

Übersicht



Testende QS-Verfahren

- Siehe Vorlesungen ST II, Requirements Engineering und Testen, Ringvorlesung „Softwareentwicklung in der industriellen Praxis“

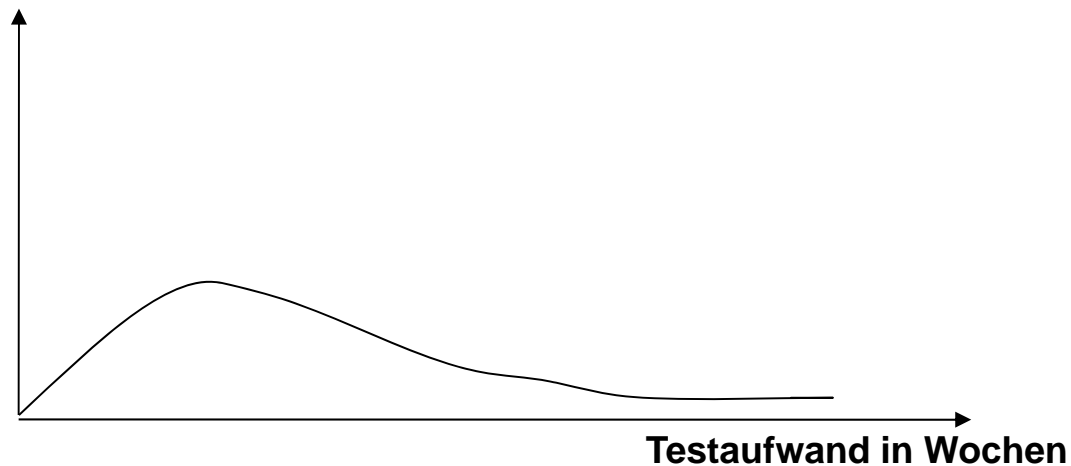
- Dynamische Prüfungen
 - **Datenbezogenes Testen** mit Testdaten: Datenstrukturen, Referenz- oder Betriebsdaten (bei großen Programmen lassen sich kaum alle Datenkombinationen erproben)
 - **Funktionsbezogenes Testen:** abschnittsweiser Vergleich des Codes incl. E/A-Verhalten mit der Spezifikation
 - **Ablaufbezogenes Testen:** werden alle Schleifen, Verzweigungen durchlaufen?

Die sieben Grundsätze des Softwaretestens nach ISTQB

- (1) Tests zeigen nur die **Anwesenheit** (nicht die Abwesenheit) von Fehlern.
- (2) **Vollständiges Testen** ist nicht möglich.
- (3) **Wespennestprinzip**: wo einmal Fehler gefunden sind, da ist die Wahrscheinlichkeit groß, dass noch andere dort existieren.
- (4) **Frühzeitig** mit der Testvorbereitung beginnen (sobald Lastenheft fertiggestellt oder ggf. noch früher).
- (5) **Wiederholung** eines Tests mit den gleichen Testdaten ist **unwirksam**.
- (6) **Abhängigkeit vom Umfeld** (sicherheitskritische Systeme verlangen andere Prüfung als kommerzielle Systeme).
- (7) Trugschluss: **keine Fehler** gefunden bedeutet **nicht automatisch**, dass ein **brauchbares System** vorliegt (ggf. wurden Anforderungen definiert, die nicht gewünschte Zielstellung erfüllen).

Datenbezogener Test

- **Regressionstest:** Vergleich zweier Versionen des gleichen Programms
- **Test-Endekriterien**
 - aus Erfahrung: in 5% aller Module stecken 95% der Fehler → Ende, wenn 5% fehlerhafte Moduln gefunden
 - nach Fehlerrate:



Black Box-Test

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

Ermittlung der Testfälle für SW-Module, -Komponenten, Pakete, Subsysteme

Methoden:

- **Äquivalenzklassenbildung**
 - Einteilung der E/A-Daten in Äquivalenzklassen (gültige und ungültige)
- **Grenzwertanalyse**
 - Testfälle an den Grenzen der Wertebereiche
- **Intuitive Testfallermittlung (unsystematisch)**
 - zusätzliche Testfälle durch Intuition (Liste möglicher Fehler aus Erfahrung, Standardfehler)
- **Funktionsabdeckung**
 - Testfälle für Normal- und Ausnahmeverhalten

White Box-Test (Überdeckungs-Testmethoden)

Ziel: Entdeckung von Fehlern durch ablaforientierte Testfälle

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

Methoden:

- **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

QS: Anforderungsdefinition und Abnahmekriterien

- Abnahmekriterien bereits während der Anforderungsdefinition aus den Qualitätszielen und den Anforderungen ableiten (im Pflichtenheft)
 - Aufdeckung von Lücken, Überschneidungen, Widersprüchen
 - Grundlage für den Nachweis des Erfüllungsgrades
 - Spezifikation eines oder mehrerer Abnahmekriterien zu einer Anforderung

- Funktionsabdeckung mit Testfällen
 - Äquivalenzklassenbildung
 - Grenzwertanalyse
 - Intuitive Testfallermittlung
- Ablauforientierte (White Box) Abnahmekriterien: welches Überdeckungskriterium?

Aufgabenmanagement im Qualitätsmanagement (Issue Management)

Aufgabenmanagement beinhaltet die Erfassung, Registratur, Speicherung und Verfolgung von

- Stand der Bearbeitung und Arbeitspaketen und Aktivitäten eines Projektplans
- Qualitätssicherungsaufgaben
- Änderungsmanagement (sieht der Kunde): Tickets, Change Requests, Feature Requests, Requirements-Management

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

Aufgabenmanagement und Strategisches Management

- Das Aufgabenmanagement dient auch 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

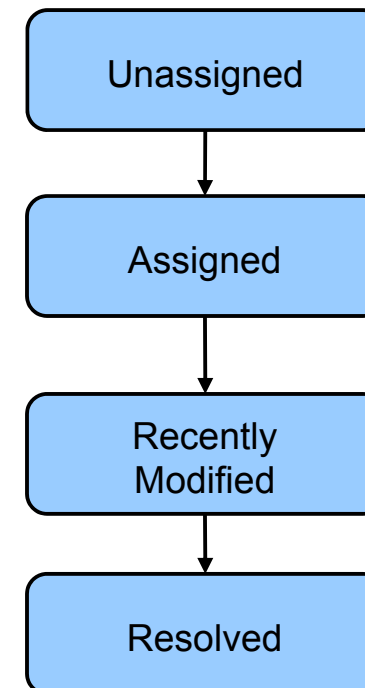
- **Issue Management:** betont, dass jedes Ticket für das Softwarehaus eine Aufgabe darstellt (“Issue”)
- **Ticket Management:** betont, dass der Kunde “Tickets” schreibt
- http://en.wikipedia.org/wiki/Comparison_of_issue_tracking_systems

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

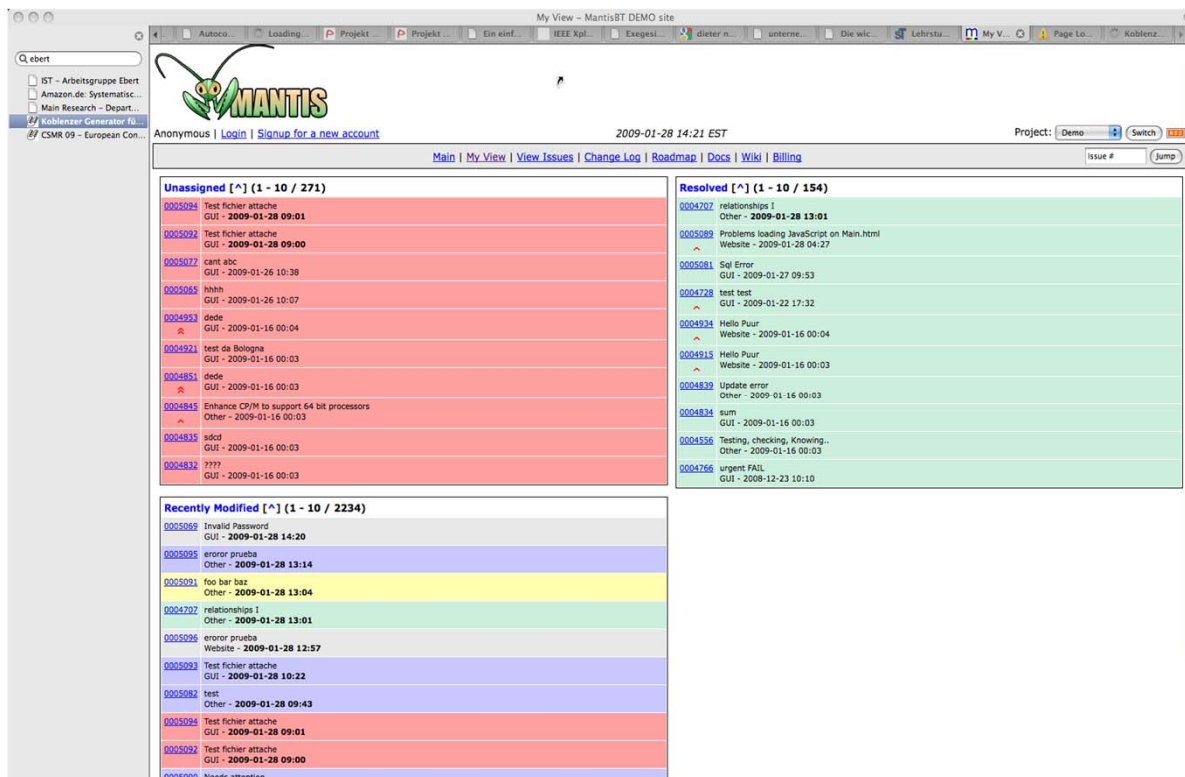
Aufgabenmanagement (Fehler und Änderungen) mit Mantis

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 Zustände der Aufgaben (Mantis)



The screenshot displays the MantisBT web interface for a project named 'Demo'. The page is titled 'My View - MantisBT DEMO site' and shows the date '2009-01-28 14:21 EST'. The interface is divided into three main sections for task status:

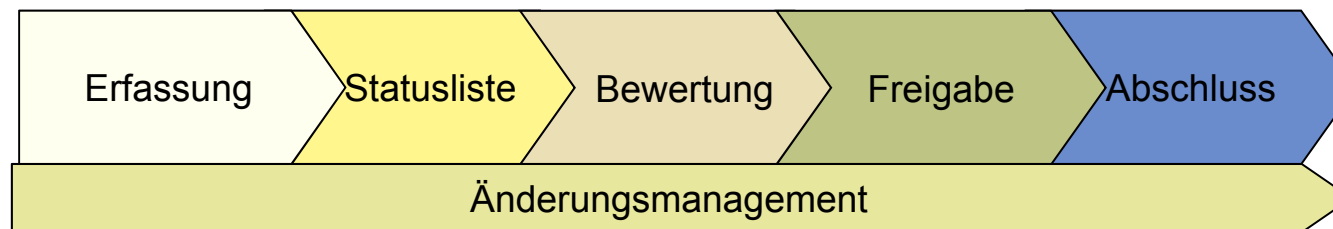
- Unassigned [^] (1 - 10 / 271)**: A list of 10 tasks with red backgrounds, including 'Test fichier attache', 'cant abc', 'hhhh', 'dede', 'test da Bologna', 'dede', 'Enhance CPM to support 64 bit processors', 'sido', and '????'.
- Resolved [^] (1 - 10 / 154)**: A list of 10 tasks with green backgrounds, including 'relationships I', 'Problems loading JavaScript on Main.html', 'Sql Error', 'test test', 'Hello Paar', 'Hello Paar', 'Update error', 'sum', 'Testing, checking, Knowing...', and 'urgent FAIL'.
- Recently Modified [^] (1 - 10 / 2234)**: A list of 10 tasks with various background colors (grey, blue, yellow, green, blue, blue, red, red, blue), including 'Invalid Password', 'error prueba', 'foo bar baz', 'relationships I', 'error prueba', 'Test fichier attache', 'test', 'Test Fichier attache', 'Test fichier attache', and 'Needs attention'.

Änderungsmanagement

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

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

- **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

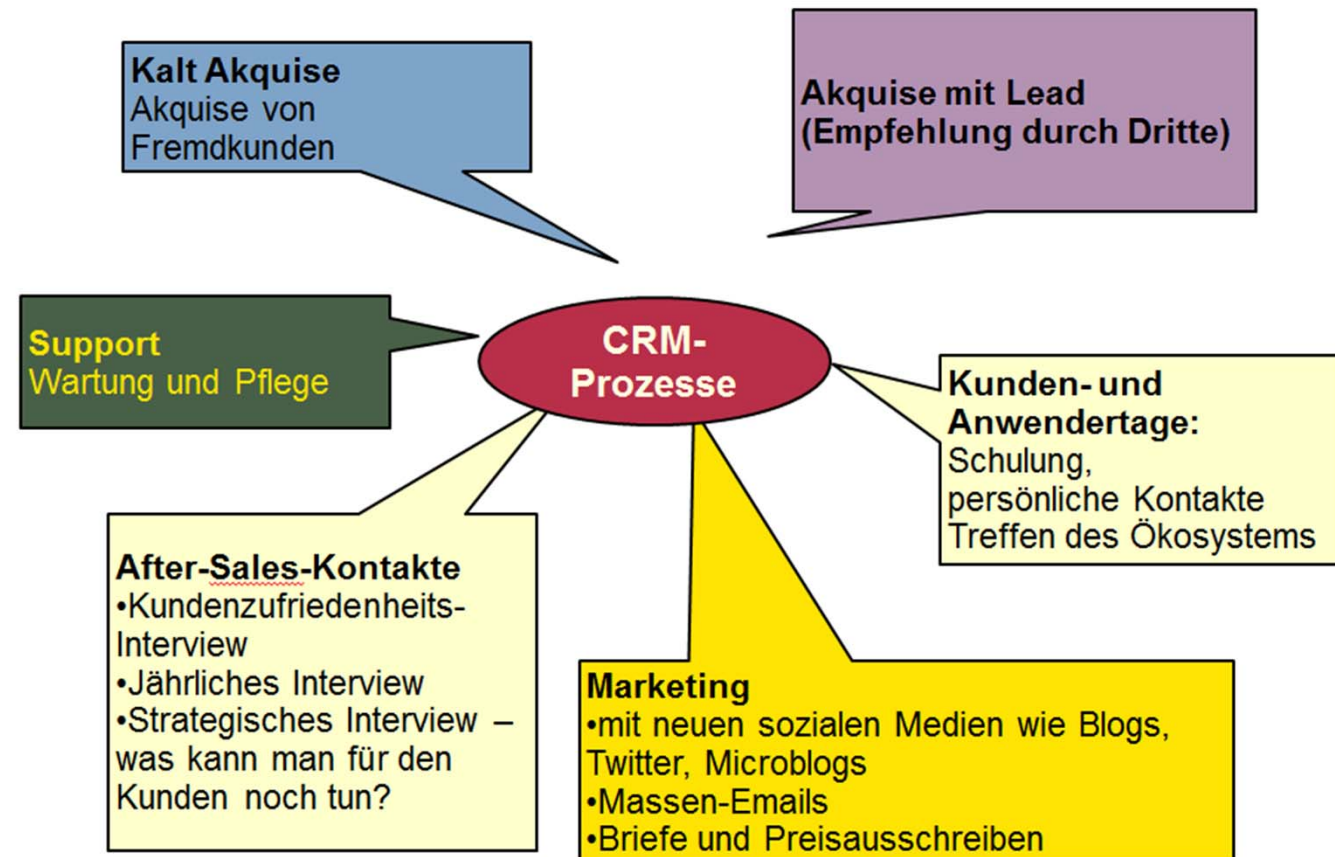


Kundenmanagement

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 den Erfolg der Firma
 - https://en.wikipedia.org/wiki/Customer_relationship_management
 - https://en.wikipedia.org/wiki/Comparison_of_CRM_systems

Beispiele			
vtiger	OSS	https://www.vtiger.com	
salesforce	Salesforce	http://www.salesforce.com/	Cloud-basiert
SugarCRM	dual licensing	http://www.sugarcrm.com/	

Literatur

- Wallmüller, E.: Software-Qualitätssicherung in der Praxis; Hanser Verlag 1990 sowie 2. Auflage erschienen 2001
- Trauboth; H.: SW-Qualitätssicherung; Oldenbourg Verlag 1996
- Balzert, H.: Lehrbuch der Softwaretechnik – Softwaremanagement. Spektrum Verlag 2008
- 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

Ende