



Software Engineering

Informatik II.

8. Software-Entwicklung – Qualitätsmanagement –

Dipl.-Inform. Hartmut Petters

Vorwort – was ich noch zu sagen hätte ...

Basis dieser Vorlesung sind vor allem die folgenden Ausarbeitungen

- Vorlesungsskript „Software Engineering“
von Prof. Dr. Martin Glinz Universität Zürich
<http://www.ifi.unizh.ch/groups/req/courses/ses/>
- Vorlesungsskript „Informatik II – Software Engineering“
von Frau Prof. Dr. Kühn FH Karlsruhe FB W
<http://www.home.fh-karlsruhe.de/~kuin0001/inhalt.htm>
- Das Buch „Software Engineering“ 6. Auflage/2001
von Prof. Ian Sommerville University of Lancaster (UK)
Addison Wesley ISBN 3-8273-7001-9

Konkret entnommene Beiträge sind i.d.R. mit einem Quellen-Verweis gekennzeichnet – sollte dieser fehlen bitte ich um Nachsicht.

Den „**roten Faden**“ durch die Vorlesung habe ich dem Skript der Vorlesung von Prof. Dr. Martin Glinz entnommen und um eigene Beiträge erweitert bzw. aus den beiden anderen Quellen ergänzt.

Für die Möglichkeit der Verwendung der wesentlichen Inhalte möchte ich mich an dieser Stelle bei den Autoren herzlich bedanken.

Grundlagen + Definitionen – 1te

- „... liefern wir Ihnen termingerecht 50.000 Programmzeilen Spaghetti-Code.“
... was bedeutet Qualität?
 - **Nicht** die **Menge** ist *wichtig*.
 - *Qualität ist* sehr *wichtig!*
 - **Was ist Qualität?** (im Bezug auf Software-Engineering)
- **Qualität (quality)**
der Grad, in dem ein Satz *inhärenter Merkmale Anforderungen* erfüllt (ISO 9000:2000)
- **Anforderungen (requirements)**
ein Erfordernis oder eine Erwartung, das oder die *festgelegt, üblicherweise vorausgesetzt* oder *verpflichtend* ist.
- **Inhärentes Merkmal (inherent characteristic)**
eine *kennzeichnende Eigenschaft* einer *Einheit* (Produkt, Dienstleistung, Prozess, System, Person, Organisation, etc.), welche diese aus *sich selbst heraus hat* und die ihr nicht explizit zugeordnet ist.

Grundlagen + Definitionen – 2te

- Qualität ist **Zielerfüllung**
- Ziele (Anforderungen) sind entweder *explizit* festgelegt worden oder *implizit* durch gemeinsame Vorstellungen der Beteiligten gegeben
- **Kein absolutes Maß** für die **Güte** einer Einheit
- Eine Auffassung von Qualität als reine *Zweckeignung* oder *Kundenzufriedenheit* greift zu kurz

Grundlagen + Definitionen – 3te

- **Qualitätsmanagement (quality management)**
aufeinander abgestimmte *Tätigkeiten* zum *Leiten* und *Lenken* einer Organisation bezüglich Qualität.
Leiten und Lenken bezüglich Qualität umfassen üblicherweise das Festlegen der *Qualitätspolitik* und der *Qualitätsziele*, die *Qualitätsplanung*, die *Qualitätslenkung*, die *Qualitätssicherung* und die *Qualitätsverbesserung* (ISO 9000 ff.)
- **Qualitäts-Managementsystem /QM-System (quality management system)**
Management-System zum *Leiten* und *Lenken* einer Organisation bezüglich der Qualität (ISO 9000 ff.)

Anmerkungen zur Terminologie

- Historische Bezeichnung für Qualitäts-Management
 - ↳ „Qualitätssicherung“ (quality assurance)mit zwei Bedeutungen (bis ca. 1995):
 - umfassend: *Management* von Qualität
 - eng: *Sicherstellung* + *Darlegung* von Qualität
- Terminologie heute
 - umfassender Begriff: **Qualitäts-Management**
 - Bedeutung von „Qualitätssicherung“ **beschränkt** auf *Maßnahmen zur Schaffung von Vertrauen*, das die Qualitätsanforderungen erfüllt werden
- Aber „Qualitätssicherung“ wird **häufig noch im alten, umfassenden Sinn** verwendet!

Warum Software-Qualitäts-Management – 1te

- Handwerk (historisch)
 - **Qualitätsbewußtsein**, tradierte Standards, **direkte Rückkopplung**
 - ↳ kein explizites Qualitäts-Management erforderlich
- Massenproduktion
 - **Arbeitsteilige** Produktion, **kaum Rückkopplung**, große Stückzahlen
 - ↳ *Explizite Maßnahmen* zur Vermeidung von Pfusch und Ausschuss *notwendig*
 - *Mittel*
 - Messung + Lenkung von Qualität mit **statistischen Verfahren**
 - *Verfahren*
 - ↳ Statistische Qualitätsanforderungen an die Produkte
 - ↳ Anforderungen an den Produktionsprozess
 - ↳ Messen und statistische Auswertung
 - ↳ Rückkopplung der Prüfergebnisse auf den Produktionsprozess
 - ↳ Kontrolle + Steuerung der Produktqualität

Warum Software-Qualitäts-Management – 2te

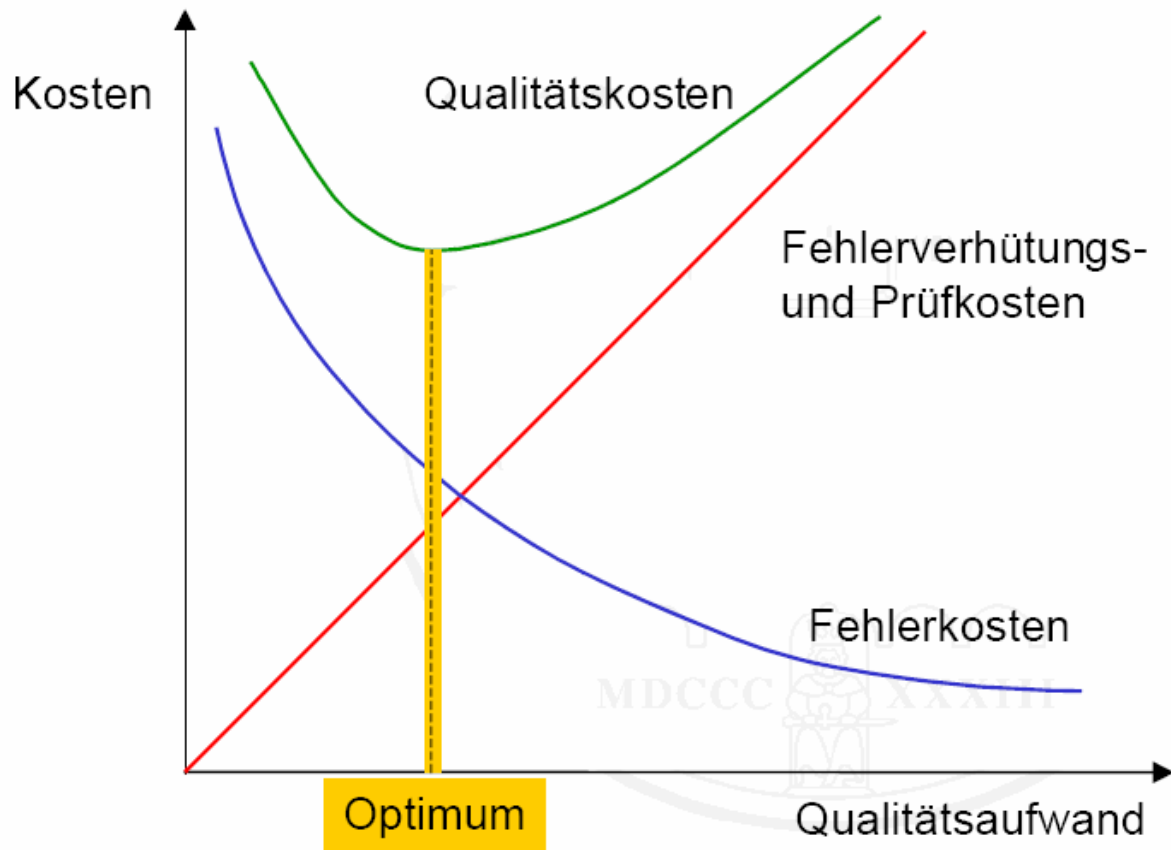
■ *Entwicklung von Produkten*

- Erst bei großen, komplexen Entwicklungsvorhaben notwendig
- Problem
 - keine großen Stückzahlen
 - Was soll/kann man messen?
 - Wie soll/kann man messen?
 - Statistische Verfahren?
 - ↳ vorwiegend Individualprüfung
Rückkopplung auf den Prozess schwierig!

■ *Software-Qualitäts-Management*

- *Besonderheiten*
 - *nur Entwicklung*, keine Produktion
 - keine tradierten Standards
 - *immateriell* – schwierig zu messen + zu prüfen
 - Rückkopplung wird nur ansatzweise beherrscht

Qualitätskosten



nach Frühauf, Ludewig,
Sandmayr (1988/2000)

Qualität ist wirtschaftlich

Was sagt die ISO 9000 ff.



■ Was ist ein **Qualitätssicherungssystem** (QS-System)?

- Ein QS-System stellt die *Aufbau-* und *Ablauforganisation* eines Unternehmens für den übergreifenden Funktionsbereich QS dar.
- Es gibt aufgrund der Vielfältigkeit der Unternehmen *kein "normiertes" QS-System*
- Die *ISO 9000/EN 29000 definiert Modelle*, an denen sich Vertragspartner bei der Ausgestaltung von QS-Systemen orientieren können.

Was regelt die ISO 9000 ff.

■ Die *ISO 9000* und ihre Folgenormen 9001/9002/9003

Die *ISO 9000* wird dann angewendet, wenn die *Verträge zwischen zwei Parteien* spezielle Leistungen hinsichtlich der *Spezifikation und Konstruktion von Erzeugnissen* verlangen, und die technischen Daten der Produkte nur in Form von Leistungsanforderungen vorliegen oder erst erarbeitet werden müssen

- ISO 9001 für die Phasen Spezifikation, Konstruktion, Produktion, Installation und Service
- ISO 9002 für die Phasen Produktion und Installation
- ISO 9003 für die Phasen Test und Endabnahme

ISO 9000 ff. – Relevanz

■ Die technische *Produktdokumentation*

Die technische Produktdokumentation ist die Gesamtheit, der während der Lebensphase eines Erzeugnisses erstellten technischen Dokumente.

■ Relevanz Software-Engineering

- ISO 9001 § 4.5.1 *Herausgabe* und Genehmigung von Dokumenten
- ISO 9001 § 4.5.2 *Änderung / Modifikation* von Dokumenten
- ISO 9001 § 4.8 Identifikation und *Rückverfolgbarkeit* von Dokumenten
Überprüfung von Dokumenten

ISO 9000 ff. – Inhalt (Beispiele) – 1te

■ Inhalt der ISO 9001 (1):

- *Verantwortlichkeit* der Unternehmensleitung
- *Umfang* eines QS-Systems
- regelmäßige *Qualitäts-Audits, Qualitätsaufzeichnung*
- *Identifikation und Rückverfolgbarkeit der Produktentwicklung (4.8)*
- *Kundenanforderungen* klar spezifiziert
- *Vertragsprüfung, Vertragsanalyse*
- *Beherrschung* des Designs
- Designdaten: Wie? Warum? Wo? Wann? Mit wem? ...
- Design Review Unterstützung, Verwendungsnachweise, ...
- *Design-Änderungen, FreigabeprozEDUREN*

ISO 9000 ff. – Inhalt (Beispiele) – 2te

■ Inhalt der ISO 9001 (2):

- *Überwachung und Überprüfung der Dokumente (4.5)*
 - *Erzeugung, Änderung: Warum? Wann? Wo? Wer? ...*
 - *Freigabe, Speicherung, Verteilung*
- Einkauf, Beschaffung, Zukaufteile
- Beherrschung des Verfahrens, Prüfung, Geräte, Test, Messung
- Behandlung/Kontrolle nichtkonformer Produkte
- Handling, Lagerung, Verpackung und Versand/Auslieferung
- Schulung, Ausbildung
- Service, technischer Kundendienst
- statistische Methoden

ISO 9000 ff. – Qualitätssicherung – 1te

■ **Qualitätssicherung** nach ISO 9001

- *Identifikation und Rückverfolgbarkeit der Produktentwicklung*
- *Überwachung und Überprüfung der Dokumente*
- *ISO 9001 Absatz 4.8*
 - Wo es zweckmäßig ist, muss der Lieferant Verfahren zur eindeutigen Zuordnung des Produktes zu den zugehörigen Zeichnungen, Spezifikationen oder anderen Dokumenten während aller Phasen der Produktion, Lieferung und Installation einführen und aufrechterhalten

ISO 9000 ff. – Qualitätssicherung – 2te

Genehmigung und Herausgabe von Dokumenten

■ ISO 9001, Absatz 4.5.1

Der Lieferant muss Verfahren zur Lenkung aller Dokumente und ihrer Inhalte einführen und aufrechterhalten, die sich auf die Forderung dieser internationalen Norm beziehen. Diese Dokumente müssen vor ihrer Herausgabe hinsichtlich ihrer Angemessenheit durch dazu befugtes Personal überprüft und genehmigt werden. Diese Lenkung muss sicherstellen, dass

- die zutreffenden Ausgaben der einschlägigen Dokumente an allen jenen Stellen verfügbar sind, wo Tätigkeiten ausgeführt werden, die für das wirksame Funktionieren des Qualitätssicherungssystems wesentlich sind;
- überholte Dokumente sofort an allen Stellen entfernt werden, an denen sie herausgegeben oder benutzt werden.

ISO 9000 ff. – Qualitätssicherung – 3te

Änderung / Modifikation von Dokumenten

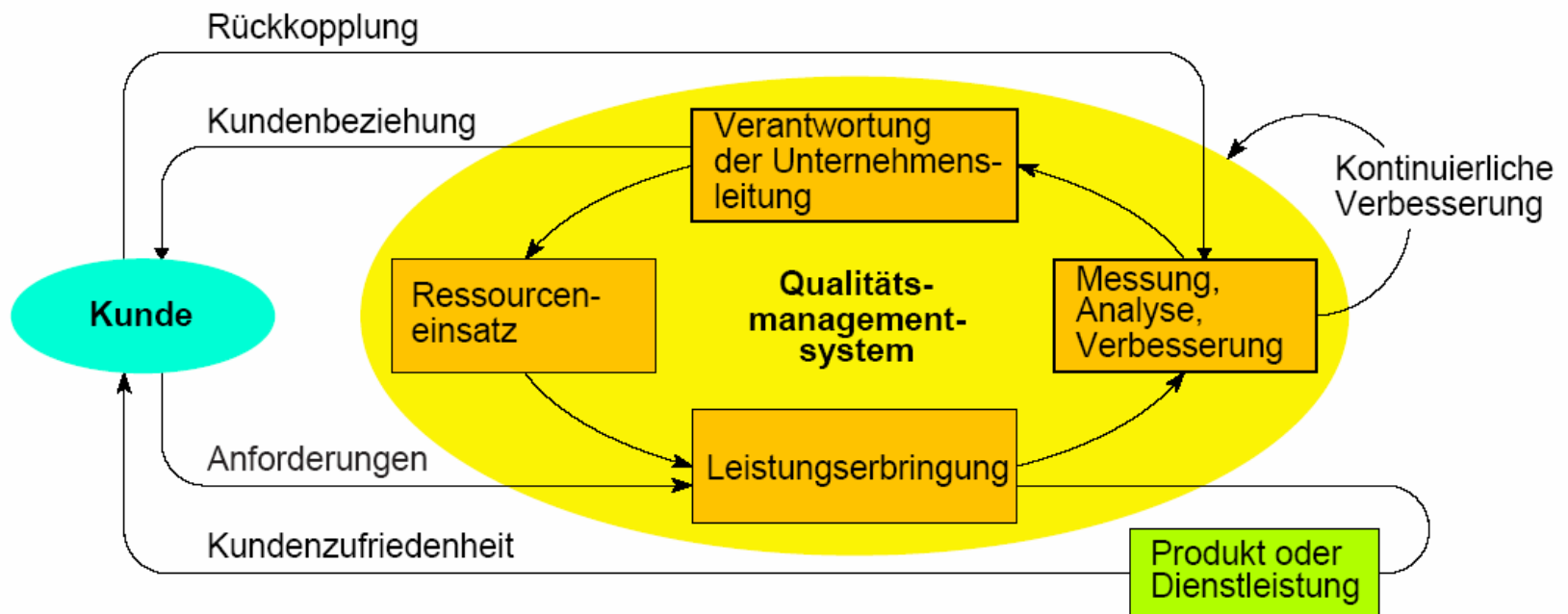
■ ISO 9001, Absatz 4.5.2

Änderungen von Dokumenten müssen durch dieselben Funktionen / Stellen überprüft und genehmigt werden, welche die Überprüfung und Genehmigung der Erst-Ausgaben ausgeführt haben, sofern nicht ausdrücklich anders festgelegt.

Die benannten Stellen müssen Zugang zu allen einschlägigen Informationen haben, auf die ihre Überprüfung und Genehmigung zu gründen ist.

Das Qualitäts-Management-System - Prinzipien

- Orientiert an
 - *Eigenverantwortung* aller Beteiligten
 - *Kundenzufriedenheit*
- *Prozessorientiert, systemischer* Ansatz zur Realisierung

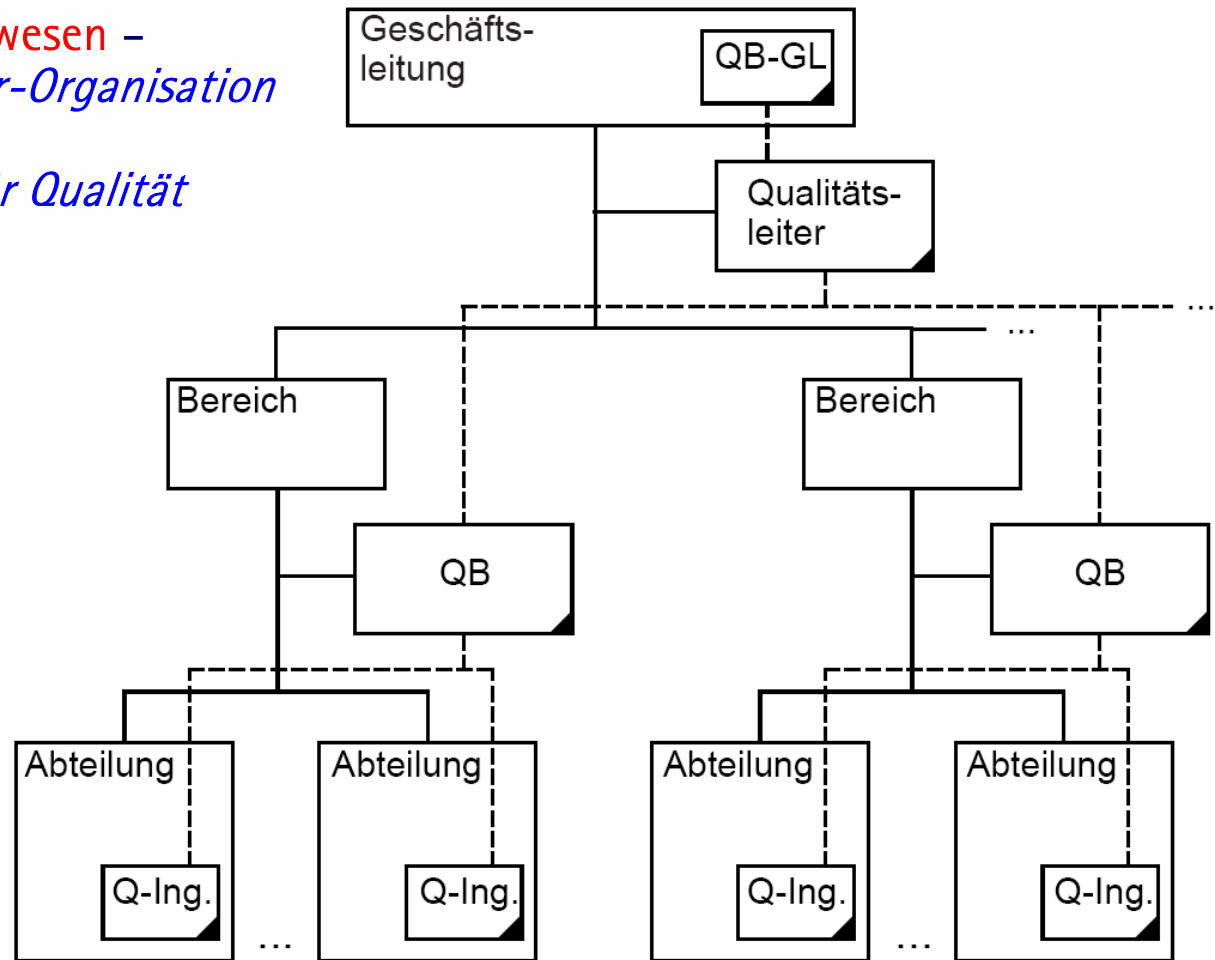


Sechs Grundsätze des Qualitäts-Managements

1. Qualität muss *erzeugt* werden, sie kann *nicht erprüft* werden
2. Qualität bezieht sich immer auf *Produkte* und auf *Prozesse*
3. *Qualitätsverantwortung* ist untrennbar verbunden mit Sach-, Termin- und Kostenverantwortung
4. Das Qualitätswesen erbringt *Dienstleistungen* und ist verantwortlich für die *Ermittlung (Messung) der Qualität*
5. Das Qualitätswesen muss einen unabhängigen *Berichtserstattungspfad* haben, der bis zur Geschäftsleitung geht
6. Die Entwickler müssen über die Qualität ihrer Arbeit *orientiert* werden

Die Aufbauorganisation

Das **Qualitätswesen** –
eine *Sekundär-Organisation*
mit den
Fachleuten für Qualität



Q-Ing.: Qualitätsingenieur QB: Qualitätsbeauftragter

Die Ablauforganisation

- Das Qualitäts-Management-System regelt alle qualitätsrelevanten
 - *Kompetenzen*
 - *Verantwortlichkeiten*
 - *Beziehungen*
- Qualitätsaufgaben *in* die *Unternehmensprozesse integrieren*
- Möglichst *wenig* Qualitätsaufgaben *separat geregelt*

Qualitäts-Management-Verfahren – 1te



Qualitätsplanung

- Definition der Qualitätsziele
 - ↳ Das wollen wir erreichen!

Qualitätslenkung

- Konstruktive Maßnahmen
 - ↳ So müssen wir arbeiten!

Qualitätsprüfung

- Analytische Maßnahmen
 - ↳ Haben wir die Ziele erreicht?
 - ↳ Haben wir richtig gearbeitet?

Qualitäts-Management-Verfahren – 2te

■ Qualitätsplanung

- *Im Allgemeinen*
Aufbau und Dokumentation des QM-Systems, allgemeine Qualitätsziele
- *Im Speziellen*
Festlegung der Qualitätsziele für individuelle Projekte

■ Qualitätslenkung

- *Im Allgemeinen*
 - Methoden, Sprachen, Werkzeuge
 - Ausbildung
 - Vereinheitlichung der Arbeitsweise
- *Im Speziellen*
Maßnahmen der Projektführung zur Erreichung der geplanten Qualität

■ Qualitätsverfolgung

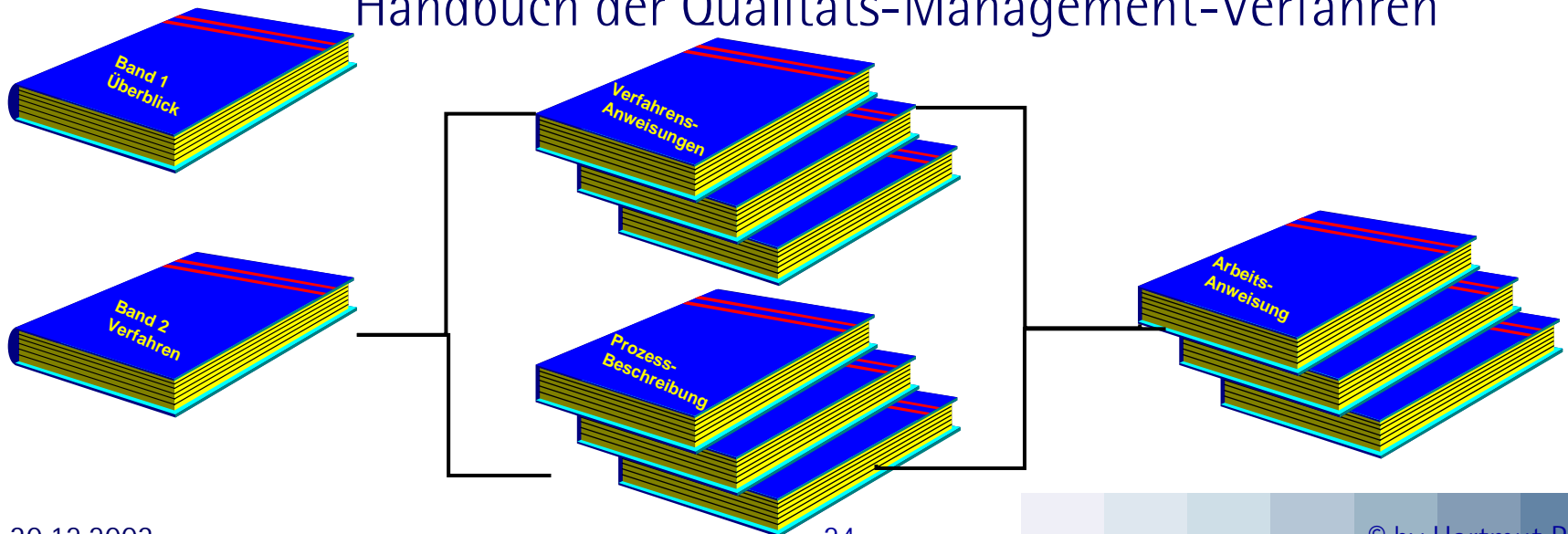
- Folgt später

Dokumentation Qualitäts-Managements – 1te

■ Qualitäts-Handbuch

beschreibt das Qualitäts-Management-System

- *Band 1* (eventuell Offenlegung für Kunden)
Qualitäts-Management-*Organisation, Überblick* über die Qualitäts-Management-Maßnahmen
- *Band 2* (nur intern / vertraulich)
Handbuch der Qualitäts-Management-Verfahren



Dokumentation Qualitäts-Managements – 2te

■ Software-Qualitätsplan

Vorgaben betreffend Qualität für die Entwicklung von Produkten

- In der Regel existiert ein allgemeiner Rahmenplan
- Rahmenplan wird bei Bedarf projektspezifisch ergänzt
- Manchmal auch vom Kunden vorgegeben

■ Software-Entwicklungs-Handbuch

detaillierte Entwicklungs-Richtlinien und Ausführungsbestimmungen

- Software-spezifisch
- Entlastet das Qualitäts-Handbuch von Software-Details

Qualitätssicherung

Die bloße Existenz eines Qualitäts-Management-Systems genügt nicht – ist nicht hinreichend!

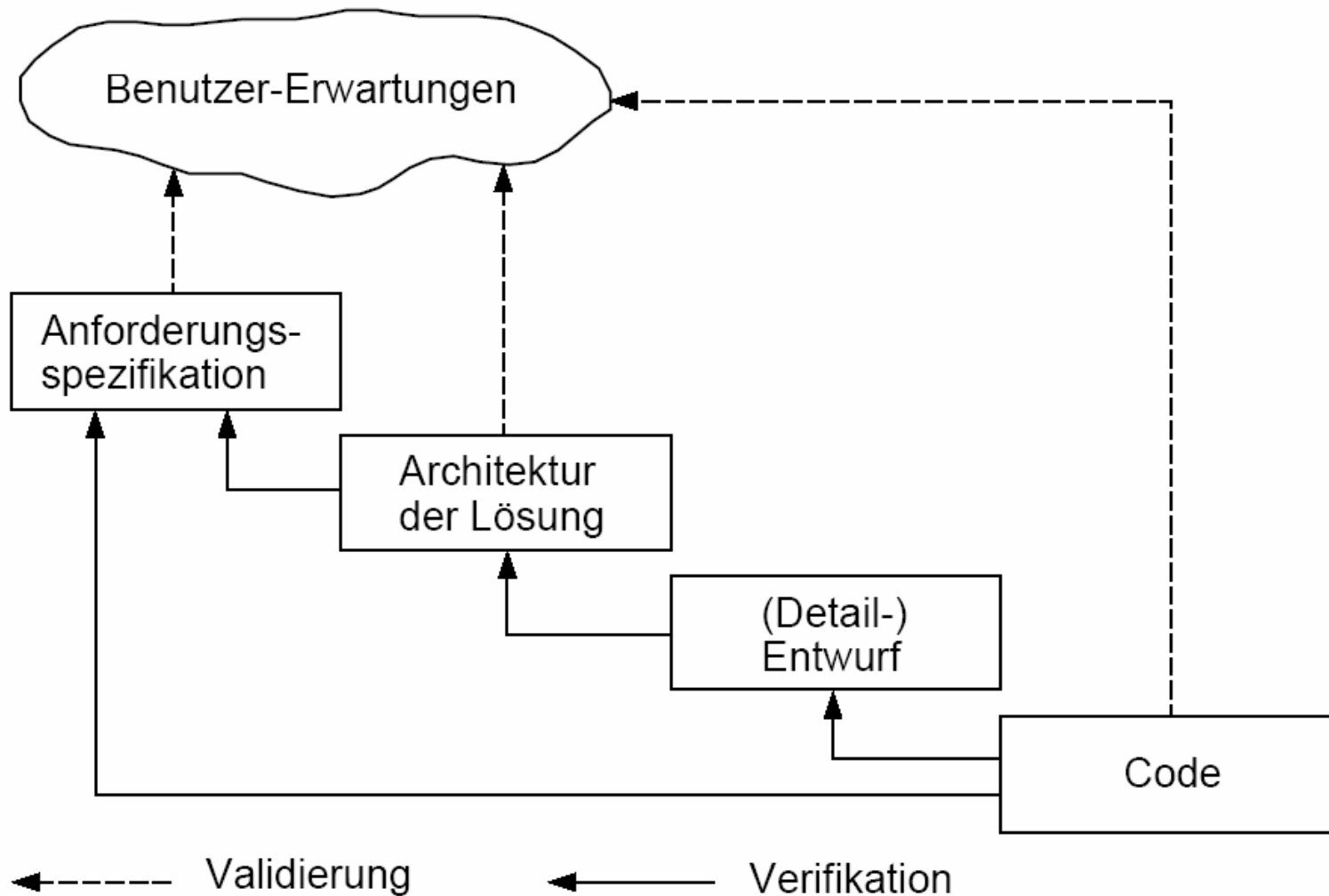
- **Qualitätssicherung (quality assurance)**
Darlegung des Qualitäts-Managements, d.h. alle Tätigkeiten zur *Schaffung von Vertrauen*, dass die Qualitätsanforderungen erfüllt werden.

- Überprüfung durch **Audits**
 - *Internes* Audit
durch das Qualitätswesen – die Qualitätsbeauftragten
 - *Externes* Audit
Überprüfung durch eine externe, auf Qualitäts-Management spezialisierte Organisation
 - Auf Veranlassung des Kunden
 - Zur *ISO-9000 Zertifizierung*

Qualitätsprüfung

- Entsprechen die Arbeitsergebnisse den Vorgaben und Erwartungen?
- Prüfung am Schluss kann zu spät (weil zu teuer) sein!
 - ↳ Qualitätsprüfung ist eine **permanente**, *die Entwicklung begleitende Aufgabe!*
- Zwei Arten der Überprüfung
 - *Wurden die Erwartungen erfüllt?*
Wurde das Richtige getan?
 - ↳ **Validierung**
 - *Wurden die Vorgaben eingehalten?*
Wurde richtig gearbeitet?
 - ↳ **Verifikation**

Validierung + Verifikation – 1te



Validierung + Verifikation – 2te

■ Validierung (validation)

Der Prozess der Beurteilung eines Systems oder einer Komponente während oder am Ende des Entwicklungsprozesses, mit dem Ziel, festzustellen, ob die spezifizierten *Anforderungen erfüllt* sind.

■ Verifikation (verification)

1. Der Prozess der Beurteilung eines Systems oder einer Komponente mit dem Ziel, festzustellen, ob die *Resultate* einer gegebenen Entwicklungsphase den Vorgaben für diese Phase *entsprechen*.
2. Der *formale Beweis der Korrektheit* eines Programms

Prüfverfahren

- **Review**
Eine formell organisierte Überprüfung eines Arbeitsergebnisses durch eine Gruppe von Gutachtern
- **Test**
Die Überprüfung eines Programms, ob dieses bei gegebenen Eingaben die erwarteten Resultate liefert
- **Prototyp**
Eine Vorab-Realisierung eines kritischen Systemteils

Seltener:

- **Simulation**
Modellierung eines bestimmten Aspekts eines Systems zur Überprüfung seines Verhaltens

In Spezialfällen:

- **Formale Verifikation**
Mathematischer Korrektheitsbeweis
- **Model Checking**
Überprüfung der Gültigkeit wichtiger Eigenschaften durch systematisches, automatisiertes Erproben

Reviews

■ Review

Eine *formell organisierte* Zusammenkunft von *Personen* zur inhaltlichen oder formellen *Überprüfung* eines Produktteils (Dokument, Programmstück, etc.) nach vorgegebenen Prüfkriterien und Prüflisten.

■ Ziele eines Reviews

- Aufzeigen der *Schwachstellen* und *Mängel* des Prüflings
- Betätigen der *Stärken* des Prüflings
- Beurteilung der **Qualität**

■ Abgrenzung

keine Reviews im hier betrachteten Sinn sind

- *informelle Prüfungen*, z.B. Durchlesen durch Kollegen
- *Management-Reviews* zur Überprüfung von Kosten + Terminen

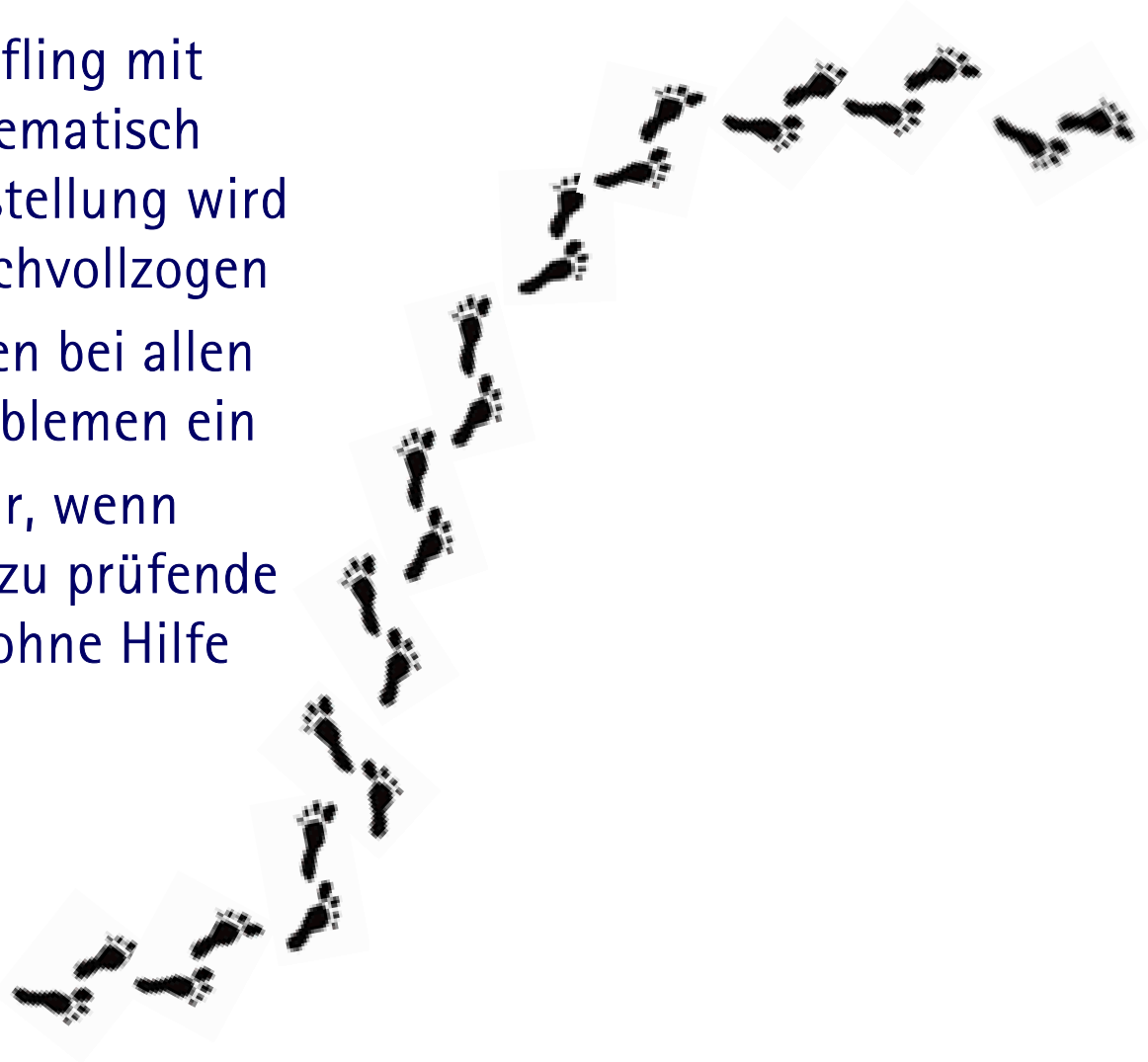
Review-Formen – 1te: Inspektion



- Prüfling wird von Gutachtern *nach vorgegebenen Prüfkriterien systematisch* Zeile für Zeile inspiziert
- Setzt voraus, dass die Gutachter das zu prüfende Material ohne Hilfe lesen und verstehen können.
- *Wirksamste* Review-Technik
- *Wo immer möglich*
 - ↳ Reviews als *Inspektion* durchführen!
- Nachstehende Ausführungen gelten primär für Inspektionen

Review-Formen – 2te : Walkthrough

- Autor geht Prüfling mit Gutachter systematisch durch, die Darstellung wird gemeinsam nachvollzogen
- Gutachter haken bei allen Mängeln + Problemen ein
- Auch einsetzbar, wenn Gutachter das zu prüfende Produkt nicht ohne Hilfe lesen können



Review-Material

■ *Prüfunterlagen*

- Der zu prüfende Gegenstand in lesbarer Form
 - ↳ Jedes von Menschen lesbare Artefakt ist „review“-fähig

■ *Referenzunterlagen*

- Sachlich-inhaltliche Vorgaben
- Vorgeschriebene Standards
- Prüflisten (Checklisten)

Warum Reviews?

■ Fehler finden

- Inspektionen *finden* die meisten *Fehler*
- Reviews sind *billiger* als testen
 - Weniger Vorbereitungs- und Durchführungsaufwand
 - Findet Fehlerursachen, nicht nur Symptome
 - Nicht jede Software-Einheit ist testbar, aber jede ist review-fähig

■ Besser werden

- *Richtiges bestätigen* / beibehalten / *verstärken*
- Arbeitsweise *vereinheitlichen*
- Ergebnisse *auswerten*
 - ↳ Prozesse + Qualität *lenken*
- Aus Schwächen + Stärken *lernen*
 - Wissenstransfer von den guten zu den schlechten Software-Entwicklern
- Reviews sind *wirtschaftlich!*

Durchführung eines Reviews (Inspektion)

■ Planung

- ↪ Termin ein*planen*
- ↪ Aufwand *budgetieren*
- ↪ Teilnehmer *einladen*
- ↪ *Material verteilen*

■ Vorbereitung

- ↪ Teilnehmer *bereiten sich individuell vor*
- ↪ *Inspizieren* den Prüfling
- ↪ *Notieren die Befunde / Ergebnisse*

■ Sitzung

- ↪ *Moderiertes Zusammentragen* und *Bewerten* der Befunde
- ↪ *Erstellen* des **Review-Berichts**

■ Überarbeiten + Nachkontrolle (nach dem Review-Termin)

- ↪ *Entscheidung* über Änderungen
- ↪ *Durchführung* der Änderungen
- ↪ *Nachkontrolle* durch Projektverantwortlichen oder neues Review

Rollen der Beteiligten (Inspektion)

- **Moderator**
 - *organisiert*
 - *leitet*
- **Gutachter**
 - *prüft*
 - *nennt Befunde*
 - *bewertet*
 - verhält sich *positiv* + *kooperativ*
- **Schreiber**
 - *protokolliert* für alle Beteiligten *sichtbar*
- **Autor**
 - *hört zu*
 - verhält sich *passiv*

Die Qualitätsverantwortung bei Reviews

- Qualität des *Reviews*
 - *Review-Teilnehmer*
 - Qualität des *Produkts*
 - *Hersteller* und deren *Management*
- ↪ **Qualitätsverantwortung**
+ *Sach-/ Kosten-/ Terminverantwortung*
gehören immer zusammen!!!

Der Review-Bericht

- *Formblätter* vorgeben / verwenden
- *Liste der Fehler-Befunde entstehen öffentlich* während der Sitzung (Video-Projektor, Folien + Taglichtprojektor, Board mit Kopierfunktion, ...)
- Bericht am Ende des Reviews *sofort fertig stellen*
- Bericht zur Durchsicht *sofort verteilen*
- Nach *Durchsicht* von *allen Beteiligten* *unterzeichnen* lassen

Review-Bericht – Deckblatt – 1te

Review-Bericht	Review-Nr.:	Seite _____ von _____
	Datum:	Zeit von: _____ bis: _____
Zu prüfendes Produkt		
Nummer / Version	Titel	
Referenzunterlagen		
Nummer / Version	Titel	
Bewertung	<input type="checkbox"/> akzeptiert (kein neues Review erforderlich) <input type="checkbox"/> wie es ist <input type="checkbox"/> nicht akzeptiert (neues Review erforderlich) <input type="checkbox"/> kleine Änderungen <input type="checkbox"/> Review nicht beendet <input type="checkbox"/> große Änderungen <input type="checkbox"/> Überarbeitung	
Review-Team		

Review-Bericht – Deckblatt – 2te

<input type="checkbox"/> Review erforderlich)				<input type="checkbox"/> Überarbeitung	
<input type="checkbox"/> Review nicht beendet					
Review-Team					
Name		Funktion		Vorbereitungszeit	Unterschrift
Freigabe (nach Nachkontrolle)					
Name		Datum		Unterschrift	

Review-Bericht – Befundliste

Review Nr.:		Review-Bericht Seite ____ von ____	
Liste der Befunde		<input type="checkbox"/>	im zu prüfenden Produkt
		<input type="checkbox"/>	in den Referenzunterlagen
Nr.	Ort	Befund	Ge- wicht

Review-Regeln – 1te

- Material *rechtzeitig* vor der Sitzung *verteilen*
- Alle Teilnehmer *rechtzeitig einladen*
- Alle Gutachter kommen **vorbereitet**
(unverzichtbar bei Inspektionen!)
- *3 bis 7* Beteiligte
- Dauer des Reviews *maximal 2 Stunden*
- Nicht mehr Material verteilen, als in der
Besprechung bewältigt werden kann
- Probleme nur *erkennen* – nicht lösen!
- *„Dritte Stunde“* nach der Review-Sitzung zur
Diskussion von Problemlösungen

Review-Regeln – 2te

- *Positives + Negatives* nennen
- *Keine Stilfragen* diskutieren
- *Produkt bewerten* – nicht den Ersteller
- Review-Bericht *niemals zur schematischen Bewertung von Mitarbeitern* verwenden
- Anhand von *Standards + Prüflisten* bewerten
- *Fehler in Referenz-Unterlagen* ebenfalls erheben / feststellen und in separater Befundliste notieren

Testen

■ Test

Ausführung eines Programms oder eines Software-Systems zwecks *Überprüfung*, ob dieses bei *gegebenen Eingaben* die *erwarteten Resultate* liefert

■ Zwei mögliche Testziele

- Möglichst viele *Fehler finden*
 - Myers (1979):
Testen ist der Prozess, ein Programm mit der Absicht auszuführen, Fehler zu finden
- Aus fehlerfreier *Ausführung mit definierten Testdaten* statistisch auf fehlerfreie *Ausführung mit realen Daten* im Einsatz *schließen*

Testen - Grundlagen

- Jeder Test ist eine **Stichprobe**
- *Korrektheit / Fehlerfreiheit* kann durch Testen **nicht bewiesen** werden
 - Beispiele
 - Addition von zwei 32-Bit-Zahlen: 2^{64} mögliche Testfälle
 - Bearbeitung einer Zeichenkette mit 32 Zeichen: 2^{256} mögliche Testfälle
- *Erwartete Ergebnisse* müssen im Voraus **bekannt** sein
 - Testen gegen die Spezifikation
 - Testen gegen vorhandene Ergebnisse (Regressionstest)
- *Unvorbereitete* und *undokumentierte* Tests sind *Zeitverschwendung*
- Testen findet *Fehlersymptome*, keine *Fehlerursachen*
- Nach dem Test
 - Fehlerursachen suchen, finden und beheben (**Debugging**)

Aufwand für vollständigen Test

- Kleine Aufwandsrechnung:

$$2^{64} \approx 1,8 * 10^{19}$$

- Annahme 1:

- Manueller Test mit 1 Testfall/Sek.

Vollständiger Test dauert dann ca. $1,8 * 10^{19}$ Sek. \approx 58,5 Mrd. Jahre

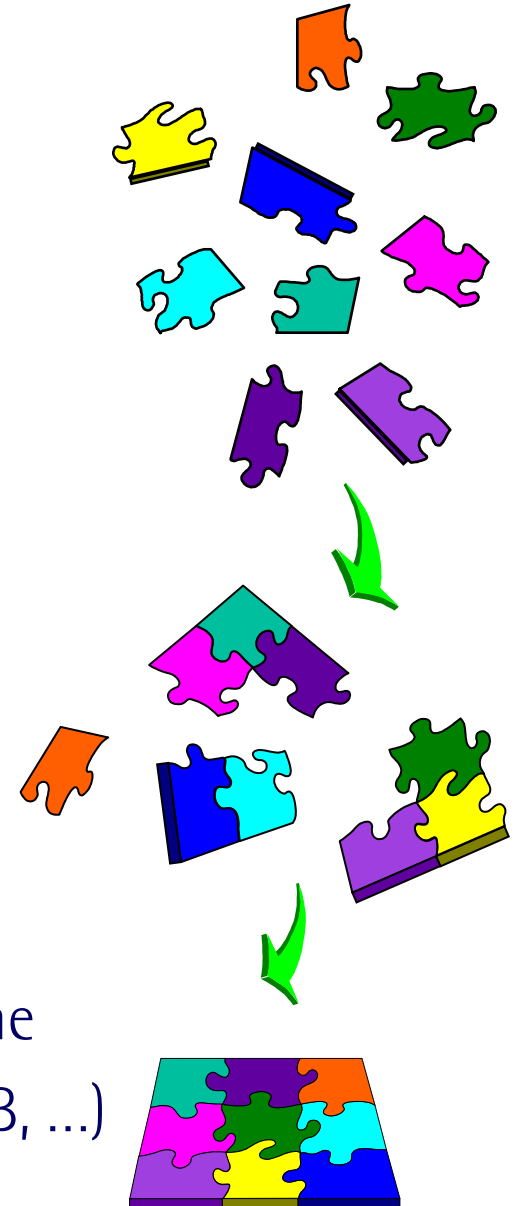
- Annahme 2:

- Automatisierter Test auf 1.000 Rechner parallel,
1 Testfall/ μ s \rightsquigarrow 10^9 Testfällen/Sek.

Vollständiger Test dauert dann ca. $1,8 * 10^{10}$ Sek. \approx 58,5 Jahre

Testgegenstände – wie testen

- Einzelkomponenten (Modultest)
 - Jede Funktion
 - Jede Routine
- Baugruppen (Integrationstest)
 - Jedes Teilprogramm
 - Jede geschlossene Funktionseinheit
 - Jedes Objekt
- Systeme (Systemtest, Abnahmetest)
 - Alle zusammenhängenden Teilsysteme
 - Komplettsystem + Zusatzsysteme (DB, ...)



Testablauf – 1te

■ *Planung*

- Teststrategie
Was? Wann? Wie? Wie lange?
- Einbettung des Testens in die Entwicklungsplanung

■ *Vorbereitung*

- Auswahl der Testfälle
- Bereitstellung der Testumgebung
- Erstellung der Testvorschriften

■ *Durchführung*

- Testumgebung einrichten
- Testfälle nach Testvorschrift ausführen
- Ergebnisse aufzeichnen
- Prüfkandidaten während des Tests nicht verändern

Testablauf – 2te

■ *Auswertung*

- Testbefunde zusammentragen + dokumentieren

■ *Fehlerbehebung* (ist ein Bestandteil des Tests!)

- Gefundene Fehler (-Symptome) analysieren
- Fehlerursachen bestimmen (Debugging)
- Fehler beheben
- Test der Korrekturen

Testverfahren

- **Funktionsorientierter Test** (Black-Box-Test)
 - *Auswahl* der Testfälle aufgrund der *Spezifikation*
 - Programmstruktur braucht *nicht bekannt* zu sein
 - Geforderte *Eigenschaften* des Prüflings möglichst vollständig *durch Testfälle abdecken*
 - *Fehlerträchtige Situationen* gezielt *provozieren*
- **Strukturorientierter Test** (White-Box-Test, Glass-Box-Test)
 - Auswahl der Testfälle aufgrund der *Programmstruktur*
 - Spezifikation muss bekannt sein (für erwartete Ergebnisse)
 - *Programmstruktur* möglichst vollständig *durch Testfälle abdecken*

Zusammenfassung - Qualitätsmanagement

- Qualität ist nicht per Definition gegeben
 - Qualität muss erzeugt – nicht erprüft werden
 - Betrifft Produkt + Prozesse
 - Qualitätsverantwortung ist fest mit Sach-, Termin- und Kostenverantwortung verbunden
 - Für Qualität sind „alle“ verantwortlich – Eigenverantwortung eines jeden
 - Rahmen ist definiert durch ISO 9000 ff.
- Qualitätsmanagement-System (QM-System)
 - Empfehlung über Rahmen des QM-Systems + Vorgehen
 - Nie Standard – immer unternehmensspezifisch
 - Eigene Parallel-Organisation zur Ablauforganisation
 - Planung – Lenkung – Prüfung
 - Permanente Aufgabe zur Sicherstellung der Qualität

Literatur – Software Engineering

- Skript Informatik II Prof. Dr. Kühn / Fb W FH Karlsruhe
<http://www.home.fh-karlsruhe.de/~kuin0001/inhalt.htm>
- Skript Software Engineering Prof. Dr. Martin Glinz Universität Zürich
http://www.ifi.unizh.ch/groups/req/courses/se_I/
- Skript Software Engineering II Bernd Kahlbrandt FH Hamburg
<http://www.informatik.fh-hamburg.de/~khh/st4se2/>
- Software Engineering
Ian Sommerville (ISBN3-82737-001-9)
- Software Engineering
- Grundkurs für Praktiker –
Roger S. Pressman (ISBN 3-89028-163-X)
- Software Entwurf
- Methoden und Werkzeuge –
A. Schulz (ISBN 3-486-21608-2)
- Software Engineering und Prototyping
Thorsten Spitta (ISBN 3-540-17542-3)
- CASE
Helmut Balzert (ISBN 3-411-03224-3)
- Software-Qualitätssicherung
Ernest Wallmüller (ISBN 3-446-15846-4)

Software Engineering

Informatik II.

9. Software-Entwicklung – Dokumentation –

