

Software-Qualitätssicherung VU Block 1

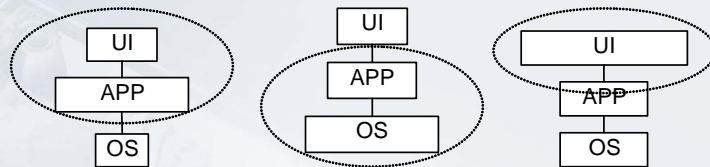
Stefan Biffl

Organisation
Einführung in die QS
Qualitätsplanung und -erhaltung
Reviews

Planung & Sicherung von Qualität

Definitionen, Leistungen, Prinzipien der QS/QM

Systemtypen im Überblick: Fokus Wert/Aufwand



A) Kommerzielles System

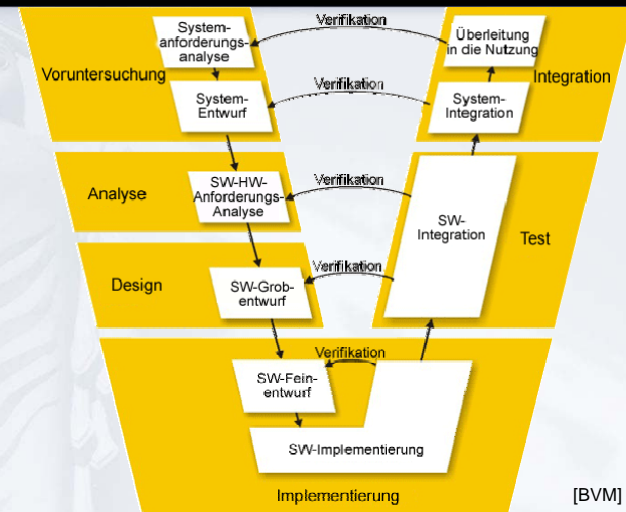
B) Echtzeit

C) Webapplikation

D) Computerspiele

UI ... User Interface (Benutzerschnittstelle)
APP ... Application (Anwendung)
OS ... Operating System (Betriebssystem)

Software-Herstellung: V-Modell

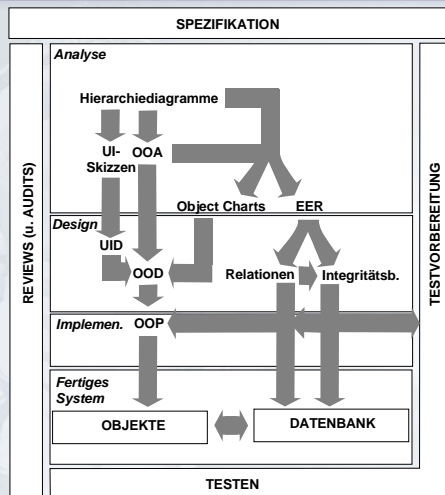


[BVM]

- **Qualität** ist die Eignung zur Erfüllung **vordefinierter Anforderungen**.
- Qualität ist die Gesamtheit von Eigenschaften und Merkmalen eines **Produkts** oder einer **Leistung**, die sich auf die Eignung zur Erfüllung vordefinierter Anforderungen beziehen.
- Die Erzeugung von guten Produkten ‚nach Gefühl‘ basiert auf fähigen Leuten und auf Zufall. Die Einhaltung von **Qualitätsstandards** kann die Bedeutung von beidem für den Gesamterfolg verringern.
- Qualitative hochwertige Software-Produkte sind
 - termingerecht und im Rahmen des Budgets erstellt.
 - für den **Anwender** verwendbar.
 - für den **Professionisten** verständlich und änderbar.
 - für den **Betreiber** effizient und administrierbar.

Die IEEE beschreibt einen allgemeinen Ansatz zur Definition von **Qualitätsfaktoren**:

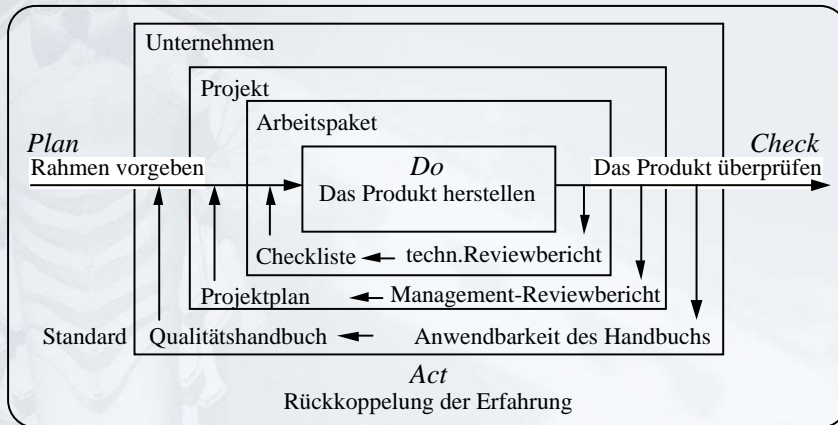
- Korrektheit
- Zuverlässigkeit
- Effizienz
- Integrität
- Verwendbarkeit
- Wiederverwendbarkeit
- Wartbarkeit
- Flexibilität
- Testbarkeit
- Portabilität
- Verknüpfbarkeit



- Qualitätsplanung**
- Reviews**
- Testvorbereitung**
- Testdurchführung**
- Fehlerbehebung**

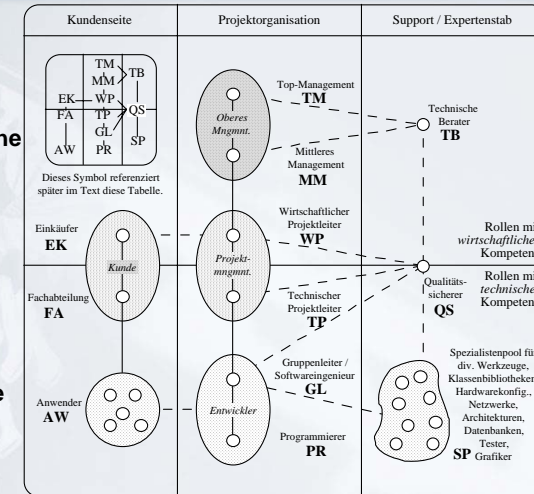
- **Qualitätssicherung** (QS) besteht in der Durchführung von **Verifikation** und **Validierung** in jeder Phase der Software-Herstellung.
- Die Organisation der QS sollte **unabhängig** von einem bestimmten Projekt sein, und projektübergreifende Leistungen erbringen.
- Qualität ist keine Eigenschaft, die später hinzugefügt werden kann. Qualität muß **während der Entwicklung** gesichert werden.
- **Qualitätsmanagement** ist die Menge aller Aktivitäten, Vorgehensweisen, Techniken und Hilfsmittel, die sicherstellen, daß ein Software-Produkt **vordefinierte Standards** erreicht oder übertrifft.

Das Plan-Do-Check-Act Konzept



Wirtschaftliche Aspekte

Technische Aspekte

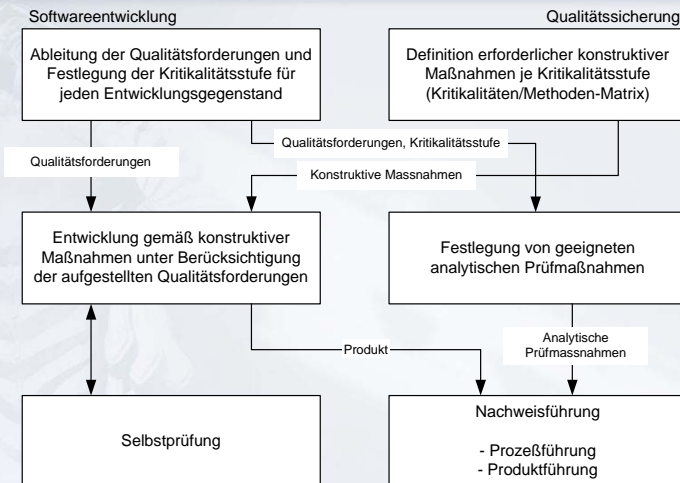


Software-Qualität wird ermöglicht, erreicht und ergänzt durch:

- Organisatorische Maßnahmen
- Konstruktive Maßnahmen
- Analytische Maßnahmen

Die Aufgaben der Qualitätssicherung lassen sich in drei Haupttätigkeiten einteilen:

- Planungsaktivitäten
- Prüfungsaktivitäten
- Lenkungsaktivitäten



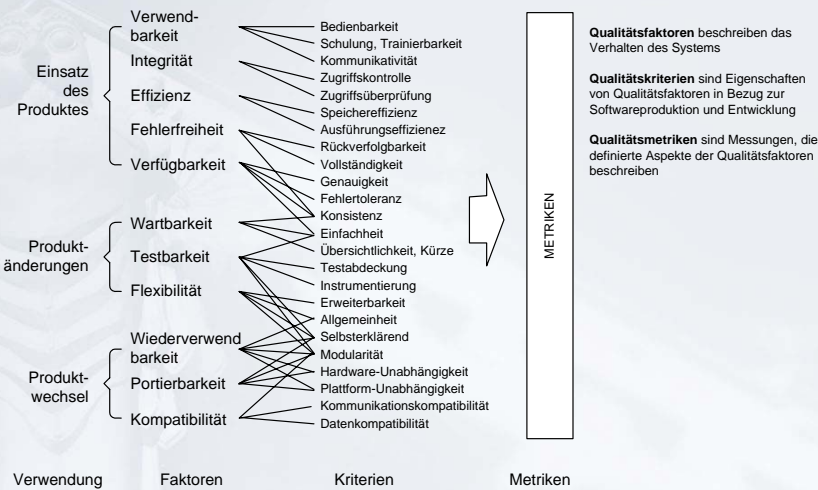
- **Qualitätsplanung** als Teil der Projektplanung.
- **Herstellen lokaler Standards** auf Projekt- und Organisationsebene.
- **Review** zentraler Projektdokumente.
- Organisieren von Reviews: Ausbildung, Planung, Durchführung, **Verbesserungsvorschläge**.
- Unterstützung bei **Personalauswahl** und Software-Zukauf.
- Vorbereitung und Auswertung von **Produkttests**.

- Konkrete **operationalisierbare** Qualitätsmerkmale
- Produkt- und **projektabhängige** Qualitätsplanung
- **Rückkopplung** der Ergebnisse der Qualitätsprüfungen
- **Mehraugenkontrolle** bei Qualitätsprüfungen
- **Frühzeitige Entdeckung** und Behebung von Fehlern und Mängeln
- **Integrierte Qualitätssicherung**
- **Unabhängigkeit** bei Qualitätsprüfungen
- **Bewertungen** der eingesetzten Qualitätsmaßnahmen

- Qualität
- Qualitätssicherung/Qualitätsmanagement
- Einbindung in SE-Projekte
- Rollen
- Leistungen

Erzeugung und Erhaltung von Qualität

Produkte, Prozesse, Personen
Qualitätsplanung: Faktoren, Maßnahmen
Prozeßmodelle



Umfeld. Bank: Kontodisposition, Wertpapierkonten, Risikoanalyse für Wertpapierkonten.

Ziel. Fernbewirtschaftung der Konten durch mündige Kunden (schneller, billiger, flexibler).

Technologie. Modem- und Internet-Anbindung mit Client/Server-Software.

Aufgaben.

1. Begriffe klären, **Qualitäten und Prioritäten** für Kunden, Entwickler und Betreiber.
2. Wichtigste **Qualitätskriterien** messbar bzw. testbar machen.
3. **QS-Maßnahmen** für Qualitäten, die für verschiedene Zielgruppen unterschiedlich wichtig sind (z.B. Kunden und Entwickler).

- Mehrwert oder Hygiene
- Sicht der beteiligten Rollen
 - Betreiber
 - Kunde/Interessent
 - Entwickler
- Feedback zu Qualitätserreichung in Phase
 - Entwicklung
 - Test
 - Betrieb
 - Wartung
- Feedback an Entwickler: direkt oder indirekt

- Mehrwert oder Hygienefaktor
 - Kunde zahlt gerne für Mehrwert.
 - Kunde kauft nicht bei Mängeln, die Hygienefaktoren betreffen, egal wie gut das Produkt sonst ist.
- Wie wichtig für welche Personengruppe
 - Geldgeber
 - Betreiber
 - Kunden
 - Entwickler
- Feedback an Entwickler
 - In welcher Phase (Entwicklung, Test, Betrieb, Wartung)
 - Direktes oder indirektes Feedback
- Achtung im QM auf ...
 - Hygienefaktoren (sind bei Entwicklern eher ungeliebt und werden daher oft ausgeblendet)
 - Faktoren, die für die Entwickler nicht besonders wichtig sind bzw. die erst in späten Phasen auftauchen oder nur indirekte Auswirkungen auf die Entwickler selbst haben.

- *Wirksam: vorbeugend oder ‚im Nachhinein‘*
- *Rolle führt Maßnahme durch*
 - *Betreiber*
 - *Kunde/Interessent*
 - *Entwickler*
- *Wirkung*
 - *sofort*
 - *Entwicklung*
 - *Test*
 - *Betrieb*
 - *Wartung*
- *Feedback an Entwickler: direkt oder indirekt*

vorbeugend	im nachhinein	Betreiber	Kunde/interessent	Entwickler	sofort	Entwicklung	Test	Betrieb	Wartung	direkt/indirekt
------------	---------------	-----------	-------------------	------------	--------	-------------	------	---------	---------	-----------------

- Vorbeugend oder im Nachhinein
- Betrifft welche Personengruppe(n)
 - Geldgeber
 - Betreiber
 - Kunden
 - Entwickler
- Feedback an Entwickler
 - In welcher Phase (sofort, Entwicklung, Test, Betrieb, Wartung)
 - Direktes oder indirektes Feedback
- Maßnahmen sind besonders zu unterstützen, die ...
 - ... Hygienefaktoren betreffen.
 - ... vorbeugend wirken sollen.
 - ... Bedürfnisse anderer Personengruppen an die Entwickler kommunizieren.
- Maßnahmen sind besonders wirksam, die ...
 - ... das Richtige einfach machen und das Falsche erschweren.
 - ... möglichst rasches und direktes Feedback an die Entwickler liefern.

- Entities
 - Definition
 - Attributes of Entities
 - Measurement of Attributes
- Measures of Attributes
- Measurement Scale Types
- Meaningfulness of Statements
- Measurement Concepts and Properties

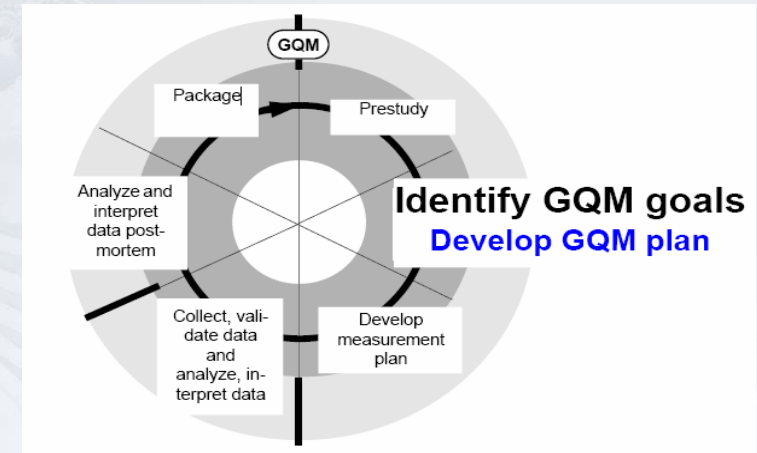
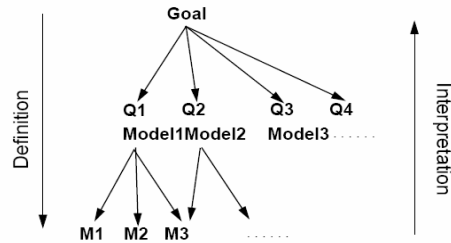
Das Entwicklungsziel des **Goal-Question-Metrik** Modells (GQM) nach Basili und Rombach ist die *Entwicklung eines Systems zur Messung von Softwarequalität.*

Das GQM-Modell besteht aus folgenden drei Stufen :

- Konzeptuelle Stufe (Goals)
- Operative Stufe (Question)
- Quantitative Stufen (Metric)

The Elements of GQM

GQM has four elements: Goals, Questions, Models, and Measures:



- Goal 1: Characterize development cost
 - Analyze the **software development process**
 - for the purpose of **characterization**
 - with respect to **effort**
 - from the viewpoint of the **management**
 - in the context **project abc/company xyz**
- Goal 2: Understand software quality wrt. reliability
 - Analyze the **software system**
 - for the purpose of **characterization**
 - with respect to **reliability**
 - from the viewpoint of the **software developer**
 - in the context **project abc/company xyz**

What information is required to achieve the measurement goal?

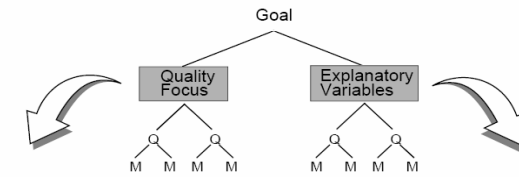
- Goal 1: If we want to characterize the development effort, then key characteristics include:
 - Distribution of effort among development phases/activities
 - Typical effort per module/component
 - Effort spent on rework
 - Hours spent on documentation
- Goal 2: If we want to understand the reliability of our systems, then key characteristics include:
 - Number and classes of defects found
 - How and when defects are found
 - Pass/fail rates for reviewing and testing

Associated with a particular goal is a set of information needed to achieve the goal:

- Questions for Goal 1:
 - What is the distribution of effort among development phases/activities?
 - How many hours are spent on rework depending on the introduction phase of the defect?
 - ...
- Questions for Goal 2:
 - What is the total number of defects found?
 - How are the defects distributed by criticality?
 - What is the distribution of defects found per detection mechanism?
 - ...

A Set of questions operationally define the goal

Question Categories



Quality Focus

- specifies the quality focus: a specific interpretation of "Reliability"

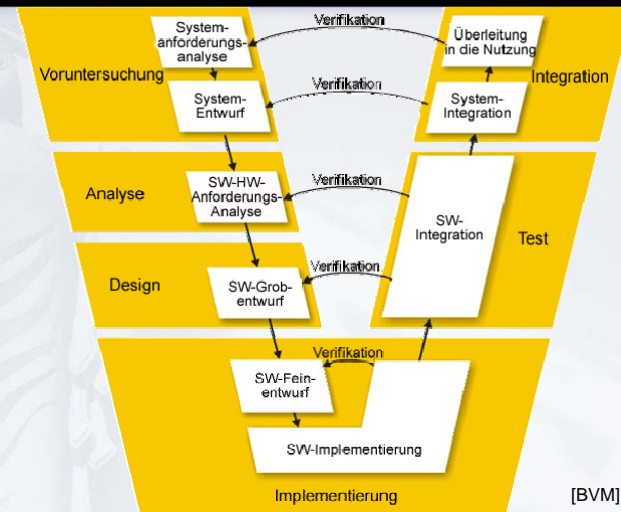
Explanatory Variables

- specifies the explanatory variables that may have an impact on the quality focus: e.g., degree of experience of the developer, used detection mechanism

SE-Prozessmodelle und QS

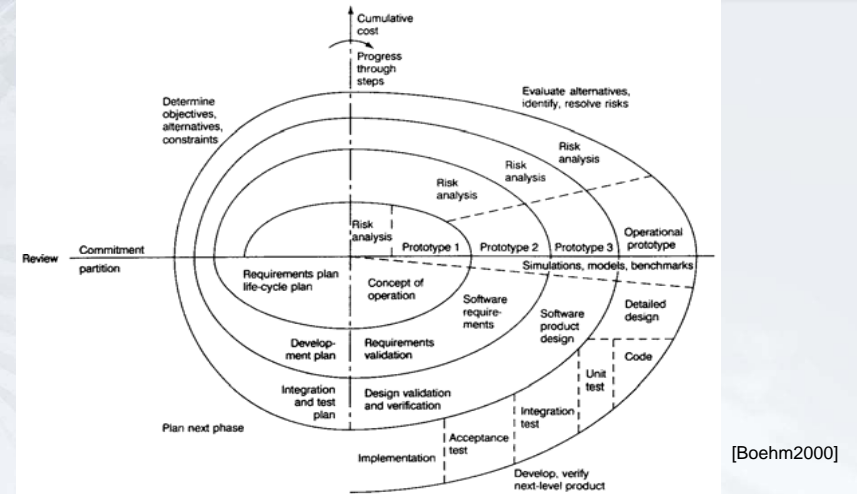
Integrierte Organisation des SE und der QS als Bauplan für ein Projekt

Sequentielles Prozessmodell: V-Modell97



[BVM]

- Spezifikation vs. Umsetzen and Testen
- Verschiedene Sichten (Rollen, Abstraktion): Applikation, Architektur und Implementierung
- Kontext von Produkten and Tests
- Problem: Fehler aus frühen Phasen der Entwicklung
- Voraussetzungen und Annahmen (z.B. Funktionen, Kernqualitäten) sollten stabil und gut verstanden sein

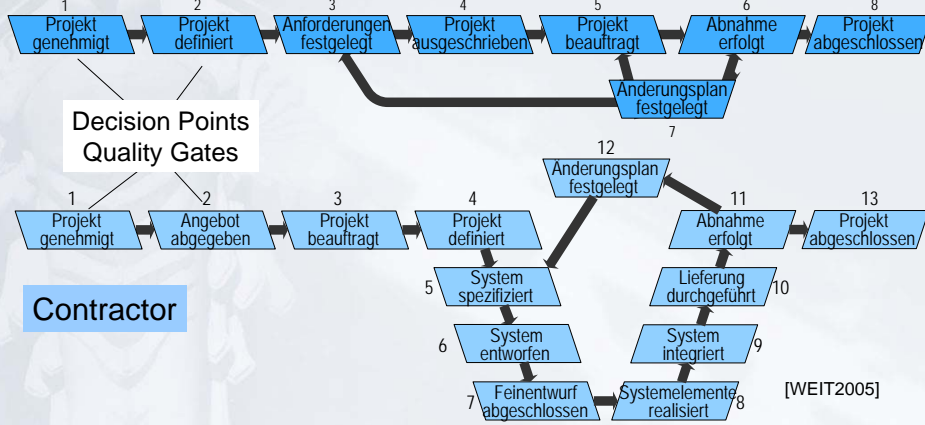


- Orientiert an Investitionen und Risikomanagement
- Sieht jede Phase als Subprojekt
 - Entscheidung über Projektumfang pro Phase
- Vier Schritte in jedem Zyklus
 1. Kick-off: Bestimme Ziele und Alternativen
 2. Einschätzung der Risiken der Alternativen
 3. Entwicklung und Test
 4. Plan der nächsten Phase; weiter bei 1.

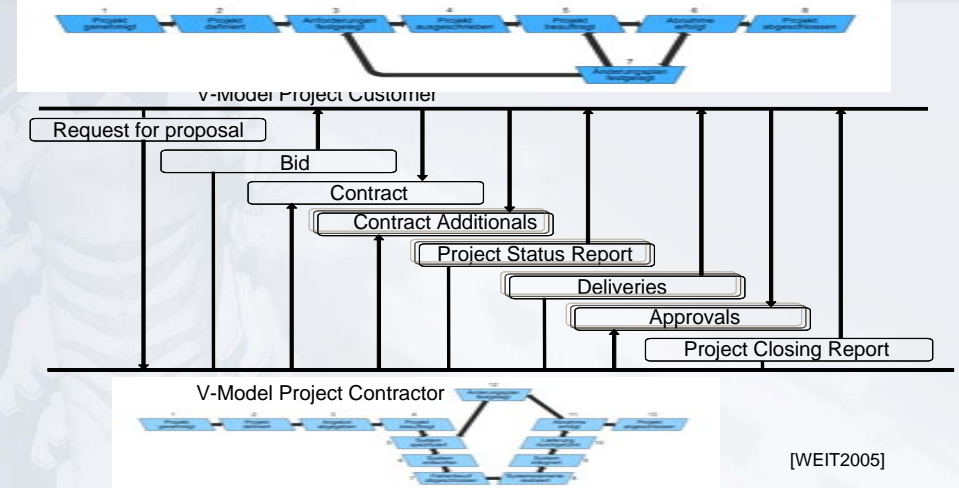
This block illustrates a 'System' for an airport, centered around an Airbus A380 aircraft. The system is supported by various infrastructure and services:

- Counter:** A check-in counter with a staff member.
- Passenger and Luggage Check In:** A person at a computer terminal.
- Terminals:** An aerial view of the airport terminal building.
- Parking:** A multi-story parking garage.
- Fire Brigade:** A fire truck at an airport.
- Service:** The interior of an airport terminal.
- Tower:** An air traffic control tower.

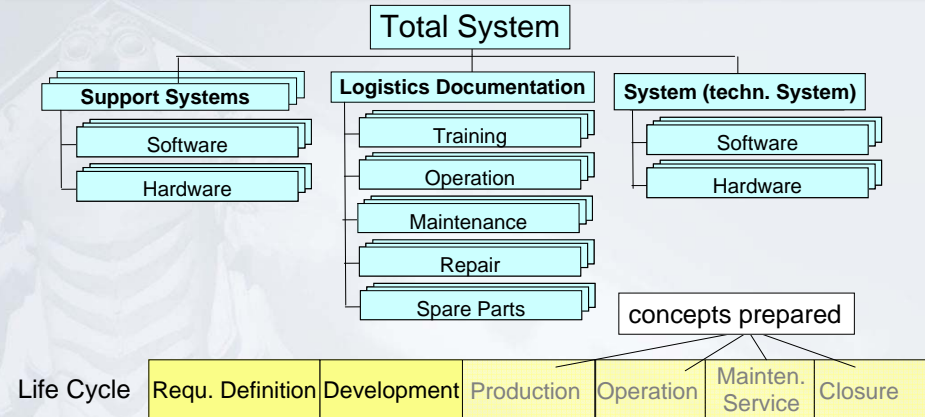
Auftraggeber



[WEIT2005]

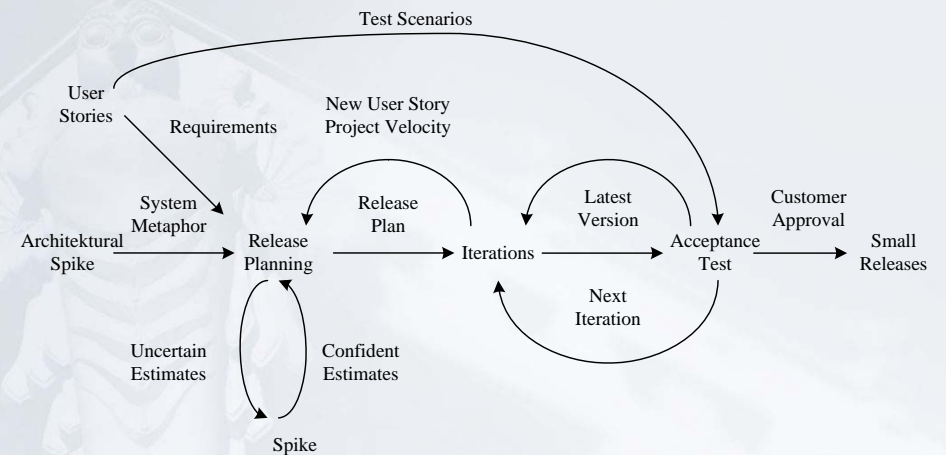


[WEIT2005]



V-Model XT supports „Systems of Systems“

[WEIT2005]

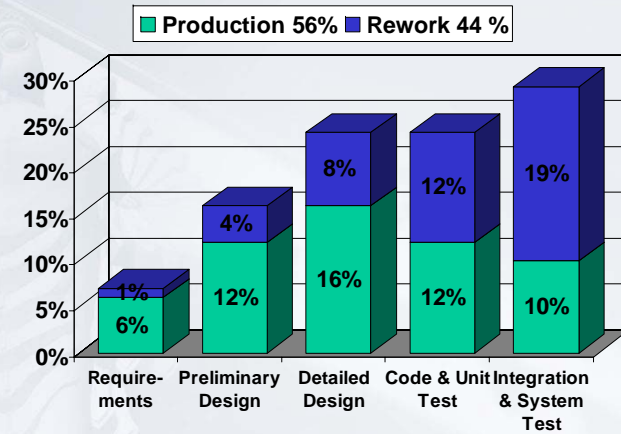


Reviews als QS-Ansatz

Stefan Biffli
Institut für Softwaretechnik und Interaktive Systeme

<http://qse.ifs.tuwien.ac.at>

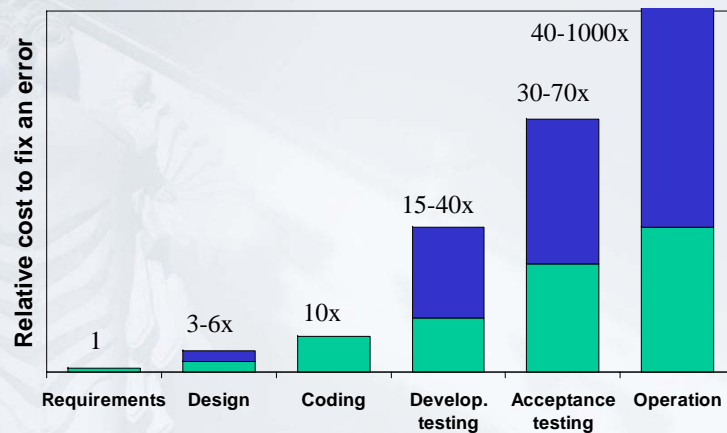
Software Development - Effort



Distribution of rework during development

Wheeler D., Brykczynski B., Meeson R., "Software Inspection An Industry best practice", IEEE Computer Society Press, 1996 Figure 1-5

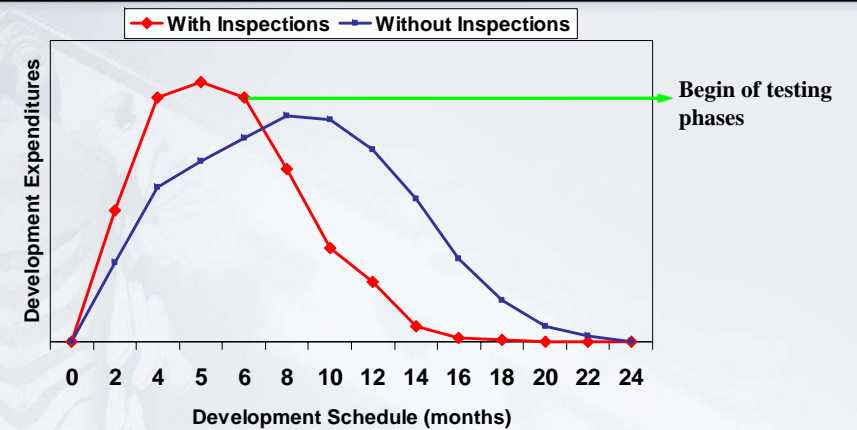
Software Development effort of rework



The cost of fixing errors escalates as we move the project towards field use.

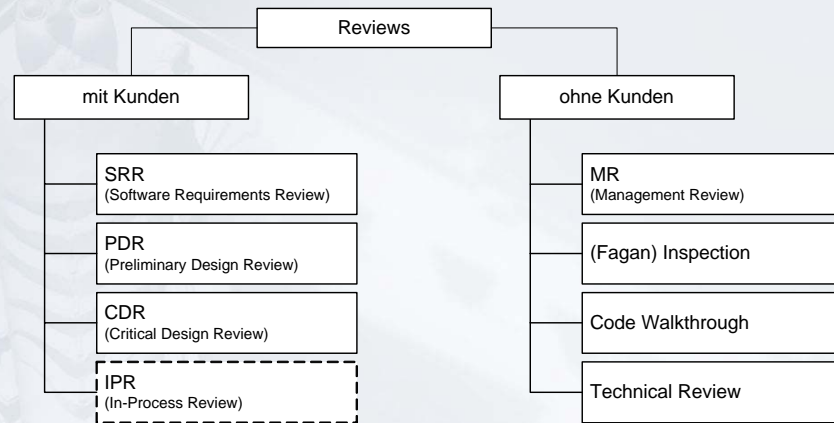
From an analysis of sixty-three projects cited in Boehm Barry, „Software Engineering Economics“, Prentice-Hall, 1981

Effort along Life cycle



Software Development Spending Profiles

Wheeler D., Brykczynski B., Meeson R., "Software Inspection An Industry best practice", IEEE Computer Society Press, 1996, Figure 1-7

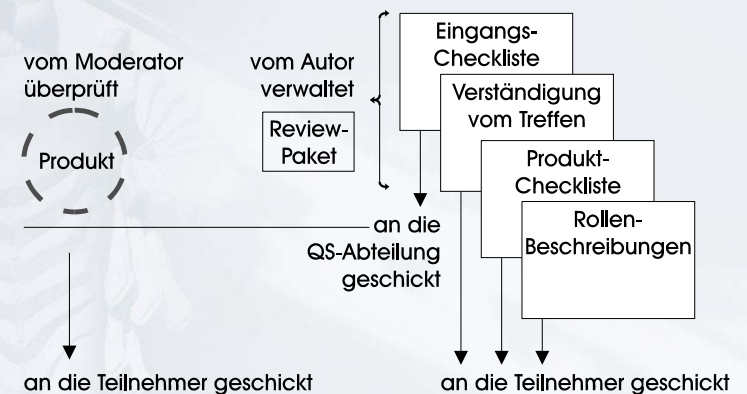


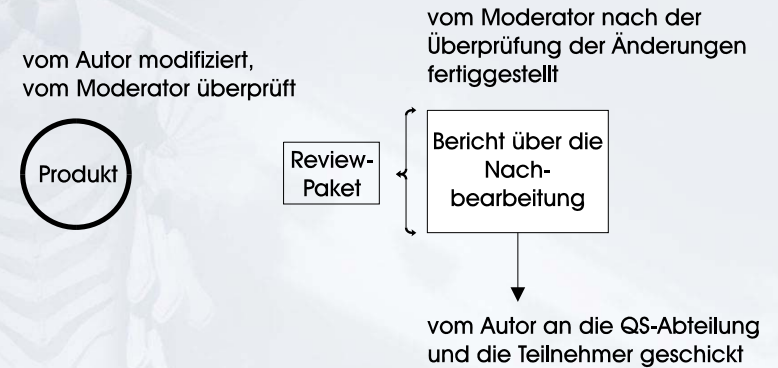
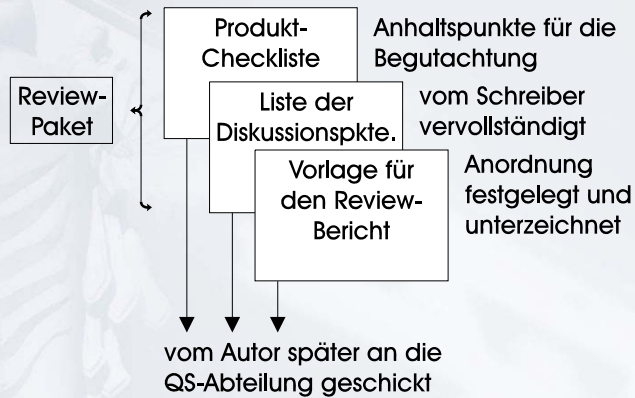
Die typische Größe eines Reviewteams ist (abhängig von der Projektart, -größe, usw.) im Bereich 3 bis 6 Personen, die sich auf folgende **Rollen** aufteilen:

- Moderator („keeper of the process“)
- Leser („keeper of focus and pace“)
- Schreiber („preserver of knowledge“)
- Gutachter („Reviewer“)
- Autor („author“)

Um brauchbare Ergebnisse zu erzielen, ist der Ablauf einer Review klar definiert und umfasst folgende **Phasen**:

- Planung („planning phase“)
- Initialisierung („initialisation phase“)
- Vorbereitung („preparation phase“)
- Reviewsitzung („review“)
- Sammlung im Team („meeting“)
- Reviewbericht („reporting“)
- Nacharbeit („re-work“)





Personen	Produkte							
	Projektplan Analyse	Anforderungen	Analysemodell	Testplan	Projektpl. Realisierung	Entwurf	Implementierung	Installation
Kunde	●	●			●			●
Projektleiter	●	●			●			●
Analytiker		●		●		●		
Integrator						●	●	●
Entwickler						●	●	
Tester				●			●	
Qualitätssicherer				●	●			

- **Situation:** Die Aufzeichnungen bei Reviews sind oft handschriftlich, was die Weiterverwendung erschwert; insbesondere in der Gruppendiskussion und für die Nachbearbeitung.
- **Ziel:** Durch den Einsatz von geeigneten Werkzeugen die Effizienz bei Reviews systematisch verbessern, falls möglich und sinnvoll.
- **Aufgabe:** Gehen Sie die Schritte einer Review durch. Überlegen Sie für jeden Schritt den Zweck des Schrittes, mögliche Ineffizienzen und Optionen für einen sinnvollen Werkzeugeinsatz.

1. Das **Produkt** reviewen, nicht den Autor.
2. Verwenden Sie einen **Arbeitsplan**.
3. Diskussionen sachlich und kurz halten.
4. Problembereiche identifizieren, aber nicht jedes Problem gleich lösen.
5. Schriftliche **Aufzeichnungen** führen.
6. Anzahl der Teilnehmer begrenzen; gute Vorbereitung aller Teilnehmer.
7. Für jedes Produkt eine passende **Checkliste** verwenden.
8. Ausreichende **Ressourcen und Zeitbudget** zur Verfügung stellen.
9. Vorab **Training** für alle Reviewer.
10. Reviews im Nachhinein beurteilen für **Verbesserung der Reviews**.

- **Inspektion**
Ziel: Behebung von konkreten Mängeln
Bed.: Leser ist nicht Autor
Unb.: Alternativen, Stil
- **Walkthrough**
Ziel: Behebung von konkreten Mängeln
Bed.: Autor ist Leser & Moderator
Bed.: Alternativen, Stil

- **Audit**
Ziel: Überblickende Kontrolle
Bed.: Planung durch externe Personen
Bed.: Projektmitarbeiter nur als Information

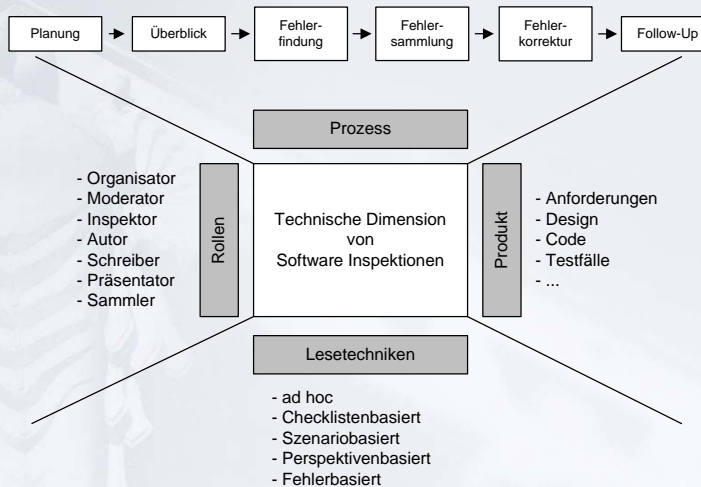
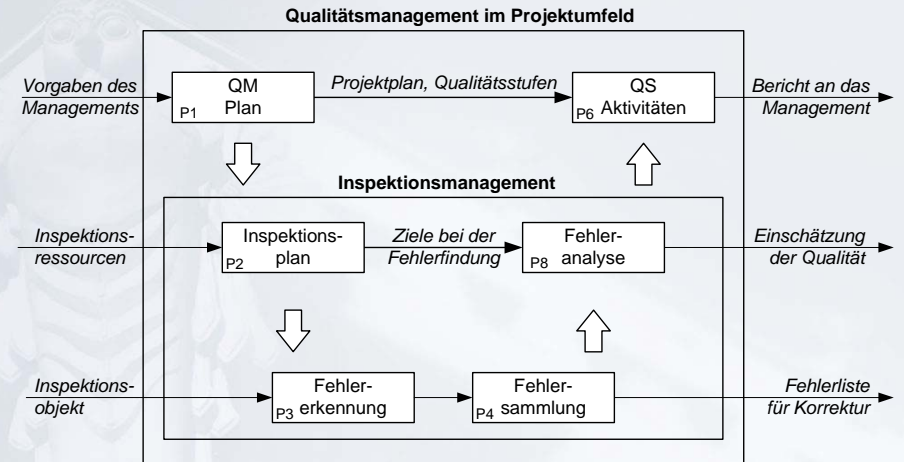
- **Situation:** Zu Beginn des Projekts „Warenversand“ soll ein Projektplan erstellt werden. In diesem Projektplan sollen auch Reviews als Mechanismen der Projektkontrolle und Qualitätssicherung vorgesehen werden.
- **Ziel:** Durch den Einsatz von Management-Reviews soll sichergestellt werden, dass der Projektplan eingehalten wird. Technische Reviews sollen die Qualität des Produkts erhöhen und die Entwicklungskosten senken.
- **Aufgabe:** Treffen Sie eine Einteilung der Reviews in einen groben Projektplan und finden Sie Richtlinien, wann Reviews durchgeführt werden sollen.

- **Situation:** Nach der Fertigstellung eines ersten Systementwurfs soll im Zuge einer technischen Review der Entwurf geprüft werden bevor mit der Umsetzung des Entwurfs begonnen wird.
- **Ziel:** Durch die frühzeitige Überprüfung des Entwurfs sollen Fehler und Mängel erkannt werden, bevor durch ihre Beseitigung hohe Folgekosten durch ein unbrauchbare Systemteile entstehen.
- **Aufgabe:** Definieren Sie die Kriterien, die der Systementwurf erfüllen muss, damit er im Zuge einer Review geprüft werden kann. Erstellen Sie zudem eine Checkliste mit Punkten, die während der Review auf jeden Fall überprüft werden müssen. Führen Sie anschließend diese Review durch.

Inspektionen sind formale, effiziente und wirtschaftliche Methoden um Fehler im Design und Code zu finden.

Ablauf von Inspektionen:

- Planungsphase (Planning Phase)
- Einführung (Tutorial)
- Vorbereitungsphase (Preparation Phase)
- Ausführung (Operation)
- Überarbeitung (Rework)
- Überprüfung der Richtigkeit (Verification)



Lesetechniken sind strukturierte Ansätze oder Dokumente, die den Fehlerfindungsprozess während einer Inspektion oder einer Review unterstützen (z.B. Checklisten, Szenarios, Usage-based, usw.)

Charakteristiken von **Lesetechniken**:

- Anwendungsspektrum
- Anleitung
- Anpassbarkeit
- Wiederholbarkeit
- Überdeckung
- Überlappung

- Kosten
 - Aufwand Inspektoren (Personenstunden).
 - Opportunitätskosten.
 - Projektverzögerung.
- Nutzen
 - Einsparungen durch **frühes Finden und Entfernen von Fehlern**.
 - Genauere Informationen über das Produkt und den Entwicklungsprozess zur Planung der Aktivitäten und Ressourcen (PM und QM).
 - Besseres **Produktwissen** der Inspektoren
 - Gemeinsame Sicht des Inspektionsteams auf das Produkt.

Klassifikation	Maßnahmen
analytische oder fehleridentifizierende Maßnahmen	
dynamische Maßnahmen, d.h. das Objekt wird ausgeführt	Testen
statische Maßnahmen, d.h. das Objekt wird nicht ausgeführt	Review
	formale Verifikation
	Messung von Qualitätsindikatoren
	...
konstruktive oder fehlervermeidende Maßnahmen	Verwendung von Hochsprachen

- Verschiedene Testebenen: Abhängig von der **Testebene** und dem aktuellen Anwendungsfall verfolgen Tests unterschiedliche Ziele:
- Modultest
 - Systemtest
 - Alphatest
 - Betatest
 - ...
- Funktion (Funktionstest)
 - Last (Lasttest)

Je nachdem, ob beim Testen die innere **Struktur der Software** berücksichtigt wird oder nicht, spricht man von **White Box** oder **Black Box** testen

- Diese Einheit behandelt die wesentlichen Aspekte der **Qualitätssicherung**, stellt Zusammenhänge zum **Qualitätsmanagement** her, und stellt einige Methoden und Werkzeugen der Qualitätssicherung vor.
- Die Qualitätssicherung ist für die Umsetzung der Vorgaben des Qualitätsmanagements, d.h. für die Auswahl, Steuerung und Überprüfung zuständig, und bildet somit die Schnittstelle zwischen Management und operativer Umsetzung.

- [Basili et al, 1994] Basili, Victor R.; Caldiera, Gianluigi; Rombach, H. Dieter: "The Goal Question Metric Approach"; Encyclopedia of Software Engineering - 2 Volume Set, pp 528-532; John Wiley & Sons, Inc.; 1994
- [Biffl, 2001] Biffl, Stefan: „*Software Inspection Techniques to support Project and Quality Management*“, Shaker Verlag, 2001, ISBN: 3-8265-8512-7
- [IEEE, 1990] IEEE-STD 610.12-1990 - Standard Glossary of Software Engineering Terminology, 1990
- [IESE, 1997][IESE, 2003] V-Modell: Entwicklungsstandard für IT-Systeme des Bundes, 1997, <http://www.iese.fhg.de/VModell/>
- [Laitenberger, 2000] Laitenberger, Oliver: "Cost-Effective Detection of Software Defects Through Perspective-based Inspections"; PhD Theses in Experimental Software Engineering Vol. 1, 2000
- [Thaller, 2000] Thaller, Georg Erwin: „Software Qualität“, VDE Verlag, 2000, ISBN: 3-8007-2494-4