

— **Dr.Ing. Immanuel Höfer**

Konstruktive und analytische Qualitätssicherungsmaßnahmen im Projektverlauf



— Gliederung

- Einführung
- Qualitätssicherung in frühen Phasen
- Review
 - Definition und Ziele
 - Planung und Durchführung
(mit beispielhaften Dokumenten)
 - Erfahrungen
- Fazit

— Was ist Qualität?

- Umgangssprachlich ist Qualität ein Sammelbegriff für positive Eigenschaften eines Gegenstands.
- DIN ISO 9126 konkretisiert den Qualitätsbegriff für Softwaresysteme:
“Software-Qualität ist die Gesamtheit der Merkmale und Merkmalswerte eines Software-Produkts, die sich auf dessen Eignung beziehen, festgelegte oder vorausgesetzte Erfordernisse zu erfüllen.“
- DIN ISO 9000:2000 liefert eine allgemeingültige Definition des Begriffs Qualität:
“Grad in dem ein Satz inhärenter Merkmale Anforderungen erfüllt.“

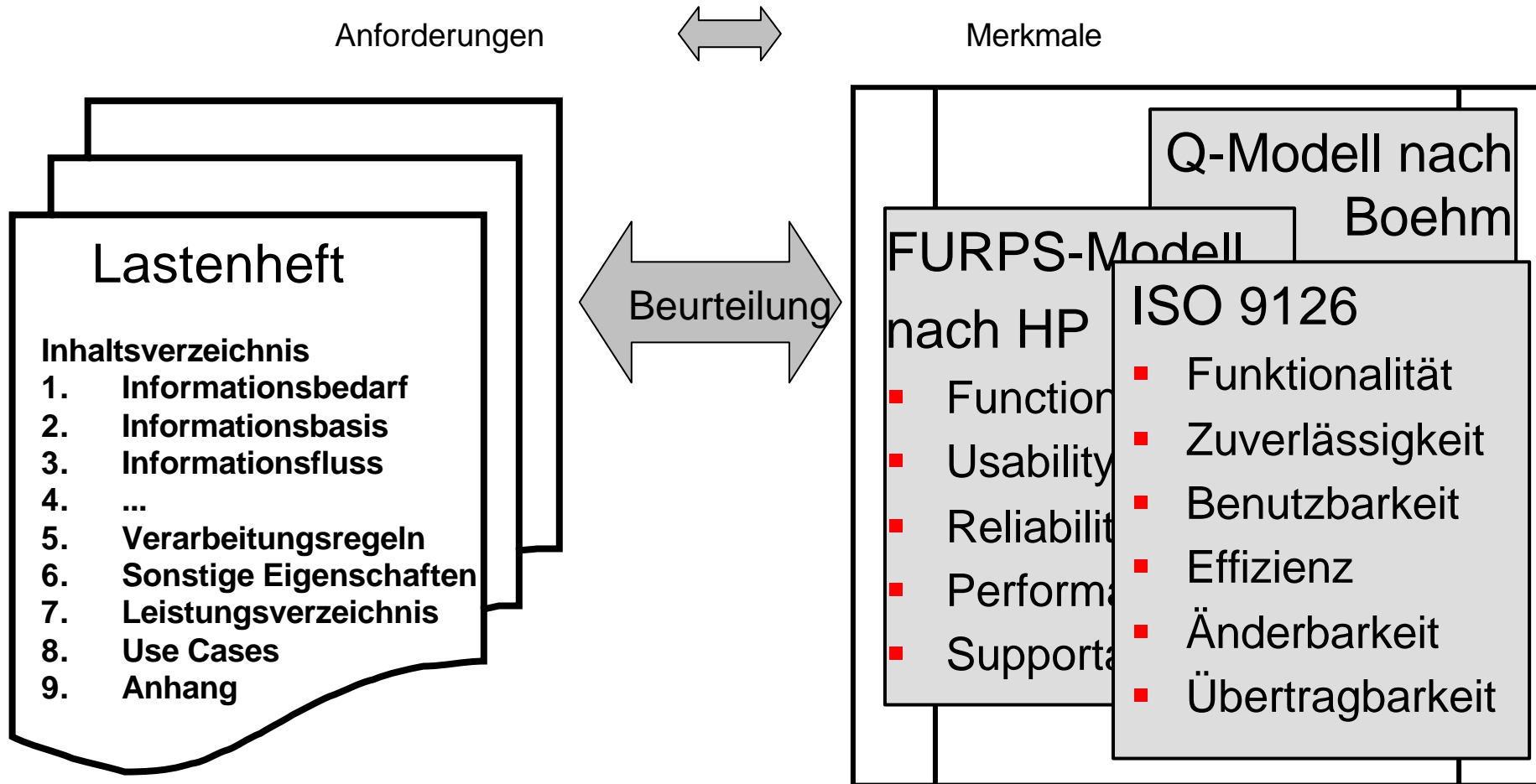
— Wie entsteht Qualität?

- Anforderungen festlegen
- Satz von inhärenten Merkmalen festlegen
- Grad der Erfüllung feststellen

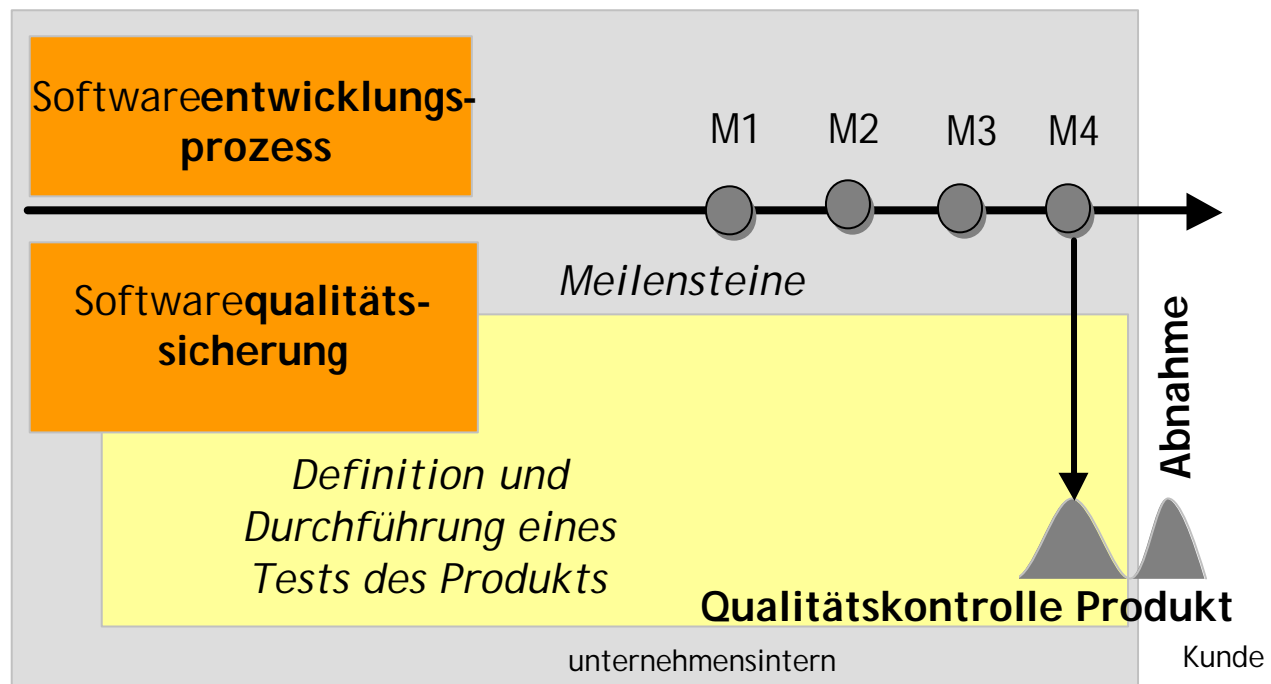
Probleme:

- Liegen definierte Anforderungen vor?
- Welche Anforderungen werden durch welche Merkmale abgedeckt?
- Wie können die Merkmale gemessen werden?

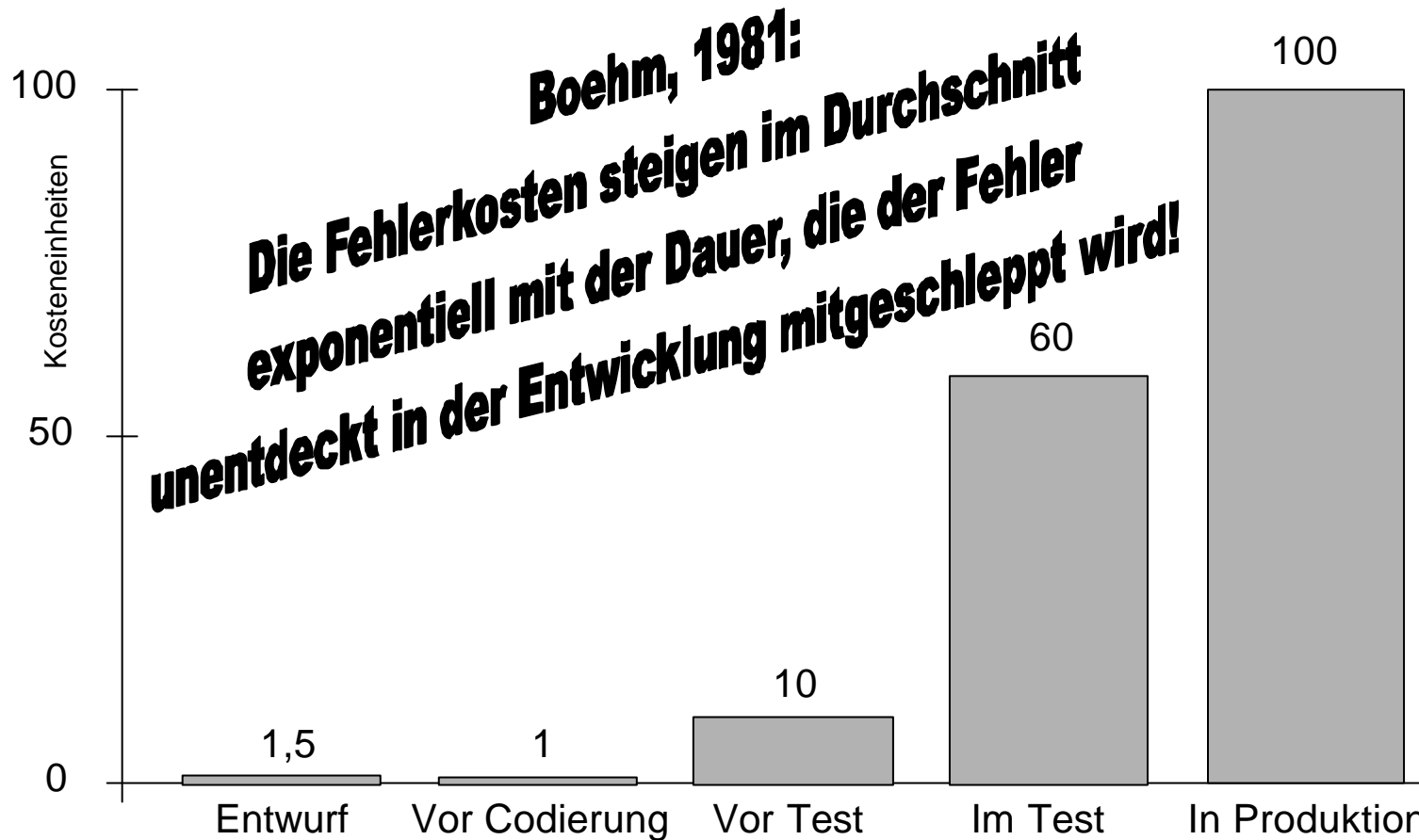
Qualität bei Softwareprodukten



Qualitätssicherung und Softwareentwicklung

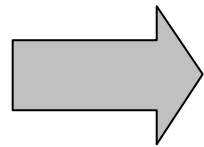


Relative Fehlerbehebungskosten

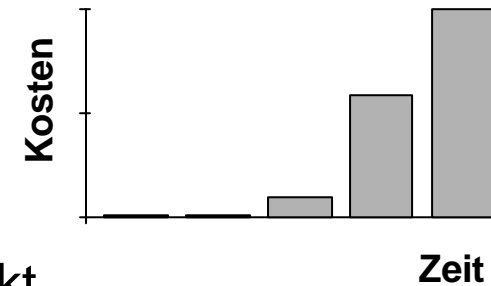


Quelle: Gilb, Tom, Software Engineering Management, Daten der Standard Chartered Bank

— Qualitätssicherung durch Testen



Risiken und Probleme



- Fehler werden erst relativ spät entdeckt

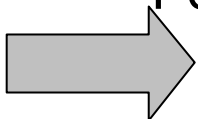


Fehlerbeseitigung teuer

- Qualitätsmerkmale können nicht überprüft werden

- Wartbarkeit

- Portabilität



Fehlerbeseitigung sehr teuer



— **Konstruktive vs. analytische QS-Maßnahmen**

Konstruktive QM-Maßnahmen

- Methoden, Sprachen Werkzeuge, Richtlinien, Standards und Checklisten die dafür sorgen, dass das Produkt bzw. der Prozess bestimmte Eigenschaften besitzt

Analytische QM-Maßnahmen

- Diagnostische Maßnahmen, die das existierende Qualitätsniveau „messen“
- Identifikation von Ausmaß und Grund der Defekte

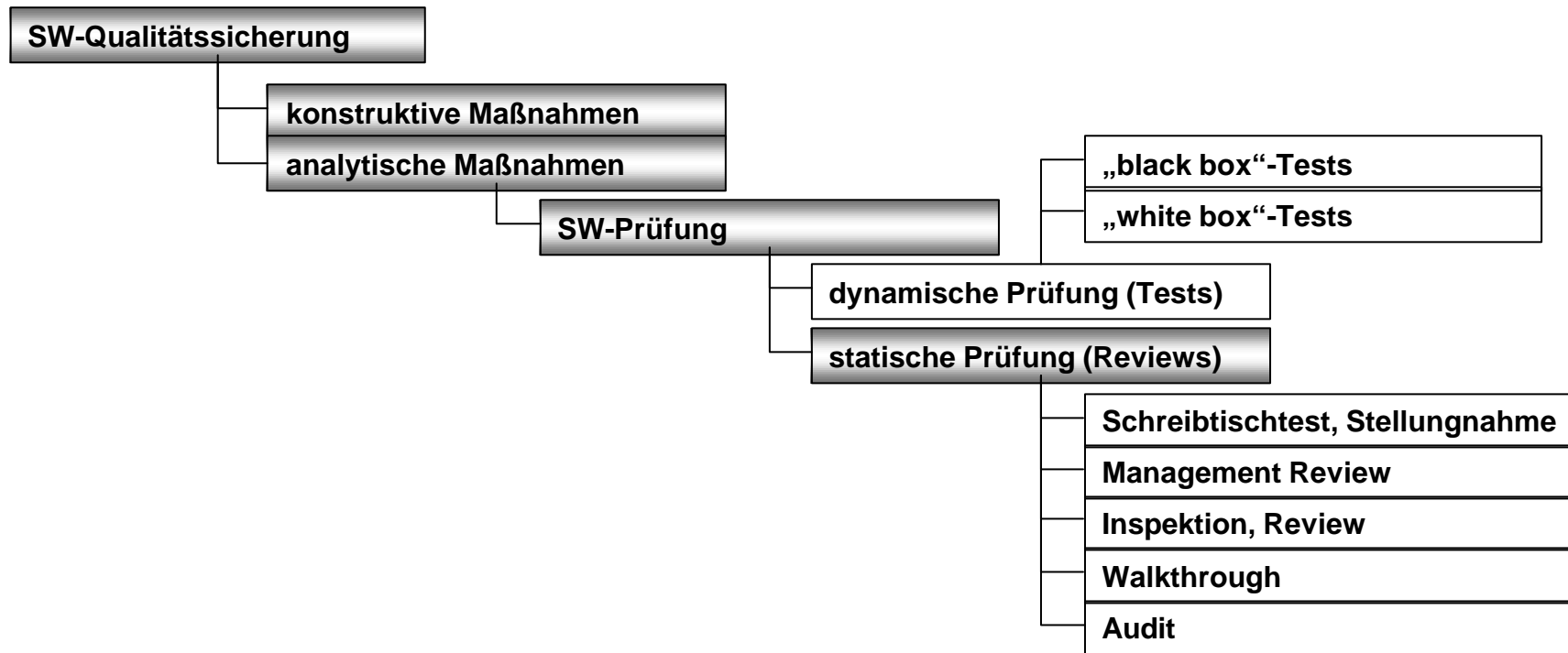
Beispiele

	Konstruktive QS-Maßnahmen	Analytische QS-Maßnahmen
Produktorientierte QS	<ul style="list-style-type: none"> ▪ Style Guides ▪ Entwurfsmuster (Patterns) 	<ul style="list-style-type: none"> ▪ Produkt-Tests ▪ Schreibtischtests ▪ Inspections, Reviews ▪ Walkthroughs
Prozessorientierte QS	<ul style="list-style-type: none"> ▪ Vorgehensmodelle (z.B. Teil des RUP) ▪ Maturity Model (z.B. CMM) 	<ul style="list-style-type: none"> ▪ Prozess-Tests ▪ Audits ▪ Assessments

— Strategie für QS-Maßnahmen

- Erfassung und Management der Anforderungen
- **Reviews** von Dokumenten und Ergebnissen, Audit von Prozessen
- Vorgehensmodell
- Iterativ inkrementelles Vorgehen mit frühen Prototypen
- Prüfbar Meilensteine
- Frühzeitige Erstellung eines Testkonzepts und der Testszenarien
- **Reviews** der QS-Maßnahmen des Auftragnehmers
- Festlegen von Q-Zielen, Q-Merkmalen, Q-Metriken
- Systematisches Testen, inkl. Testplanung und -protokollierung
- Konfigurations-, Fehler- und Änderungsmanagement

Überblick



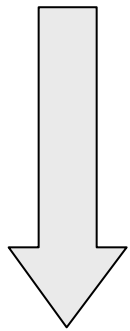
In Anlehnung an Ludewig, „SW-Prüfung“, Zürich 2000 und IEEE Std.1028-1997

— Was ist ein Review?

- “a process or meeting during which a software product is presented to project personnel, managers, users, customers, ... for comment or approval“
 - IEEE Std.610.12-1990, Standard Glossary of SW Terminology
 - IEEE Std.1028-1997, Standard for Software Reviews

- Ein Review ist die inhaltliche Prüfung eines Arbeitsergebnisses (Dokument, Zeichnung, Programm etc.) gegenüber Vorgaben und gültigen Richtlinien
 - in Anlehnung an Ludewig, „SW-Prüfung“, Zürich 2000

— Welche Ziele hat ein Review?



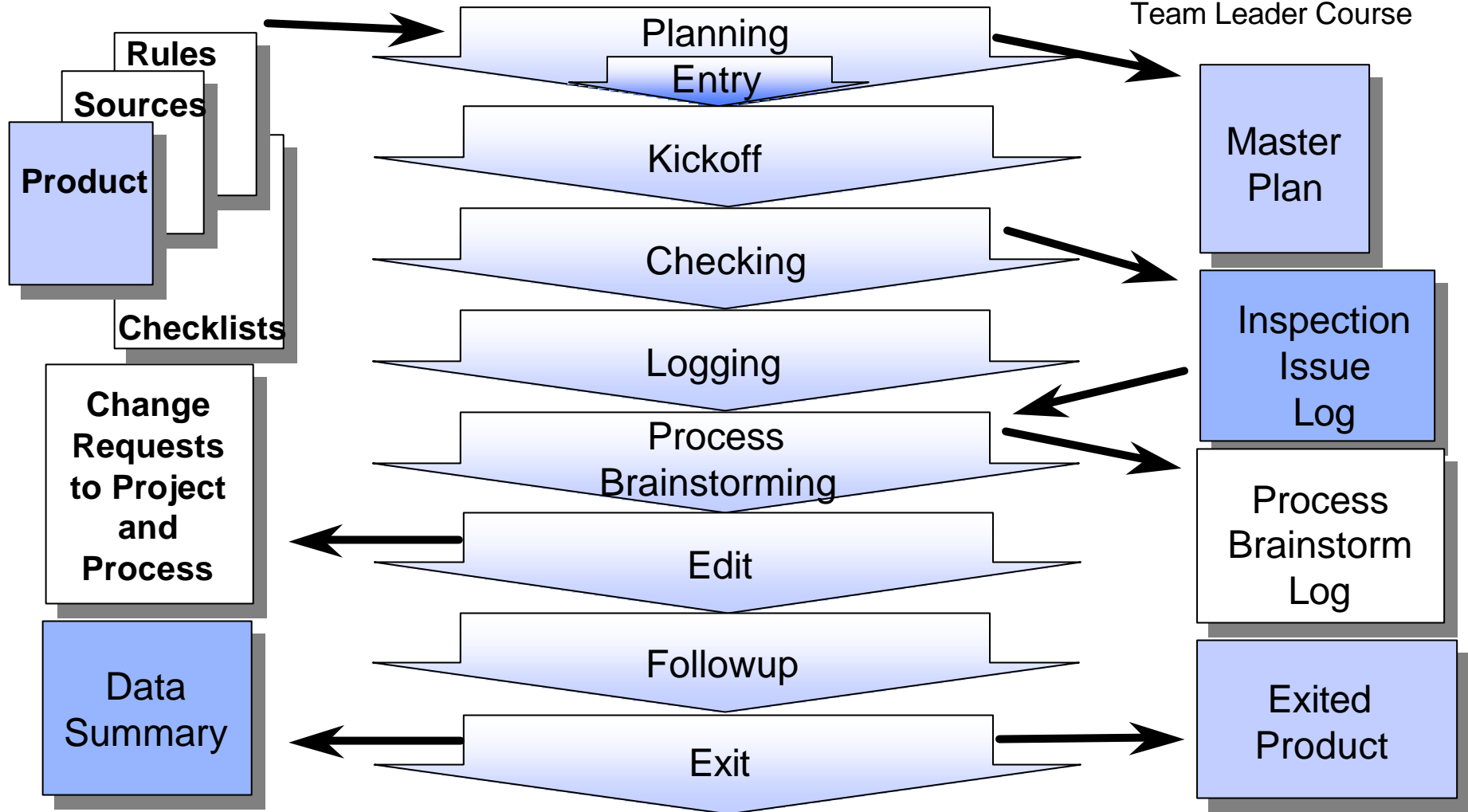
Dokumente
Konzepte
Code
Planung
.....

- Aufdecken von Fehlern eines Prüfobjektes
 - möglichst früh im Entwicklungsprozess
 - mit Hilfe von unabhängigen Experten

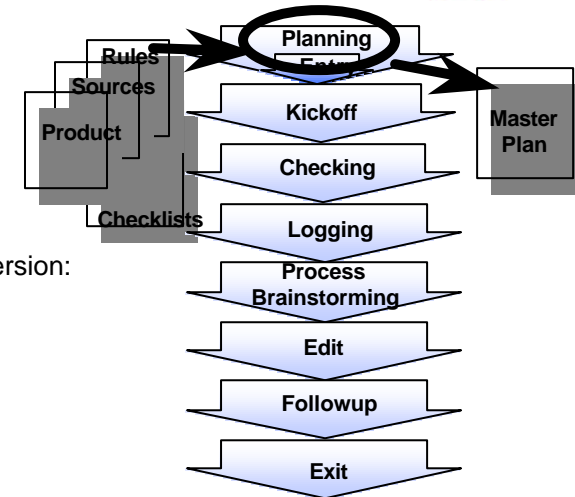
- Nicht Ziel des Reviews ist es,
 - das Problem zu lösen,
 - den Fehler zu korrigieren,
 - über eine Lösung zu diskutieren oder
 - einen Fehlerverantwortlichen anzuklagen

Überblick Review Prozess

Source: Tom Gilb,
Team Leader Course



Review - Protokoll



Prüfprotokoll

Produkt(e):

Version:

Prüfzeitraum:

Prüfort:

Nr	Name des Prüfers	Rolle	Unterschrift	Bereich	Bemerkungen
1.					
2.					

Individuelle Vorbereitung der Prüfer

Geprüfte Seitenzahl	Gesamtzeit Vorbereitung	Gesamtzahl Fehler Bewertung A	Gesamtzahl Fehler Bewertung B	Gesamtzahl Fehler Bewertung C	Gesamtzahl Fragen

Statistische Auswertung der Prüfung

Fehlerrate (min/Fehler)	Fehler/Seite	Anzahl fachliche Fehler	Anzahl semantische Fehler	Anzahl syntaktische Fehler

Freigegeben von:

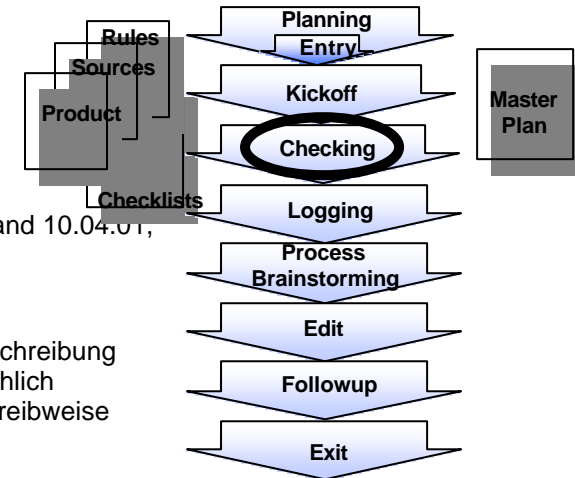
Datum / Unterschrift:

Fehler finden mit einer Checkliste

Fragen zu formalen und inhaltlichen Kriterien

Nr.	Kriterien
1.	Ist eine Änderungshistorie vorhanden und ausgefüllt?
2.	Ist der Text in seiner Breite (Umfang) und Tiefe (Detaillierungsgrad) auf die Zielgruppe des Dokumentes abgestimmt? Dabei versteht man unter der Breite den inhaltlichen Vollständigkeitsgrad mit dem ein Thema behandelt wird (Ausmaß der Behandlung von Ausnahme- und Sonderfällen).
3.	Gibt es fehlerhafte Inhalte?
4.	Gibt es Widersprüche?
5.	Stimmt die Beschreibung in der Dokumentation mit dem jeweils geltenden Zustand überein?
6.	Sind die vermittelten Informationen missverständlich?
7.	Sind die Überschriften aussagekräftig?
8.	Ist die Argumentation für die beschriebenen Alternativen ausreichend?
9.	Sind alle technischen Risiken berücksichtigt?
10.	
11.	Usw.

Review - Befundliste



Befundliste (Ausschnitt aus der Revieweinheit Oberflächendaten erfassen), Stand 10.04.01,

Referenz: z.B. auf Seite oder Abschnitt

Befund: Kurzbeschreibung der Abweichung

Bewertung: abnahmeverhindernd (A), kritisch (B), unkritisch (C), Frage (?)

Fehlerart: fachlich, funktional (1), wie z.B. falsche/ unvollständige/ fehlende Beschreibung
Semantisch, inhaltlich (2), wie z.B. nicht eindeutig, unklar, widersprüchlich
Syntaktisch, formal (3), wie z.B. falsche Zeichensetzung, falsche Schreibweise

Erledigt: Datum und Kürzel des Mitarbeiters

Nr.	Referenz	Befund	Bewertung	Fehlerart	Wird erledigt bis	Wurde erledigt am
1	S. 41 UC-1003: Objekt erfassen	Befundkategorie: BO Der Begriff Objekt wurde nicht definiert	?	2	13.05.01	
2	S. 41	Befundkategorie: EH Plausibilitätsprüfung: „Offensichtlich falsche Daten des Benutzers werden dabei abgewiesen und der Benutzer wird entsprechend informiert.“ Welche Benutzer ist hier gemeint, der Benutzer, der die Daten ursprünglich angelegt hat? Oder der Benutzer, der z. Zt. die Daten modifiziert? Oder beide? Oder der aktuelle Datenerfasser?	A	1	13.05.01	
3	S.41	Befundkategorie: BO Wichtige Attributen der Systemstammdaten fehlen. Kein Verweis auf die Attributen der Systemstammdaten oder Pflichtstammdaten wurde erwähnt.	A	1	13.05.01	

Beispiel TÜViT

Nr.	Anzahl Worte	Anzahl Seiten, bereinigt (500 W/P)	Dokumentart	Bemerkungen
1.	13500	27	Grobkonzept Teil 1	-
2.	5800	12	Grobkonzept Teil 2	Dialogbeschreibung mit 19 Masken
3.			Kein Dokument zur Systemarchitektur im Repository	-
4.	4500	9	Fachkonzept Iteration1 Teil 1	-
5.	8500	17	Fachkonzept Iteration1 Teil 2	Dialogbeschreibung mit 10 Masken

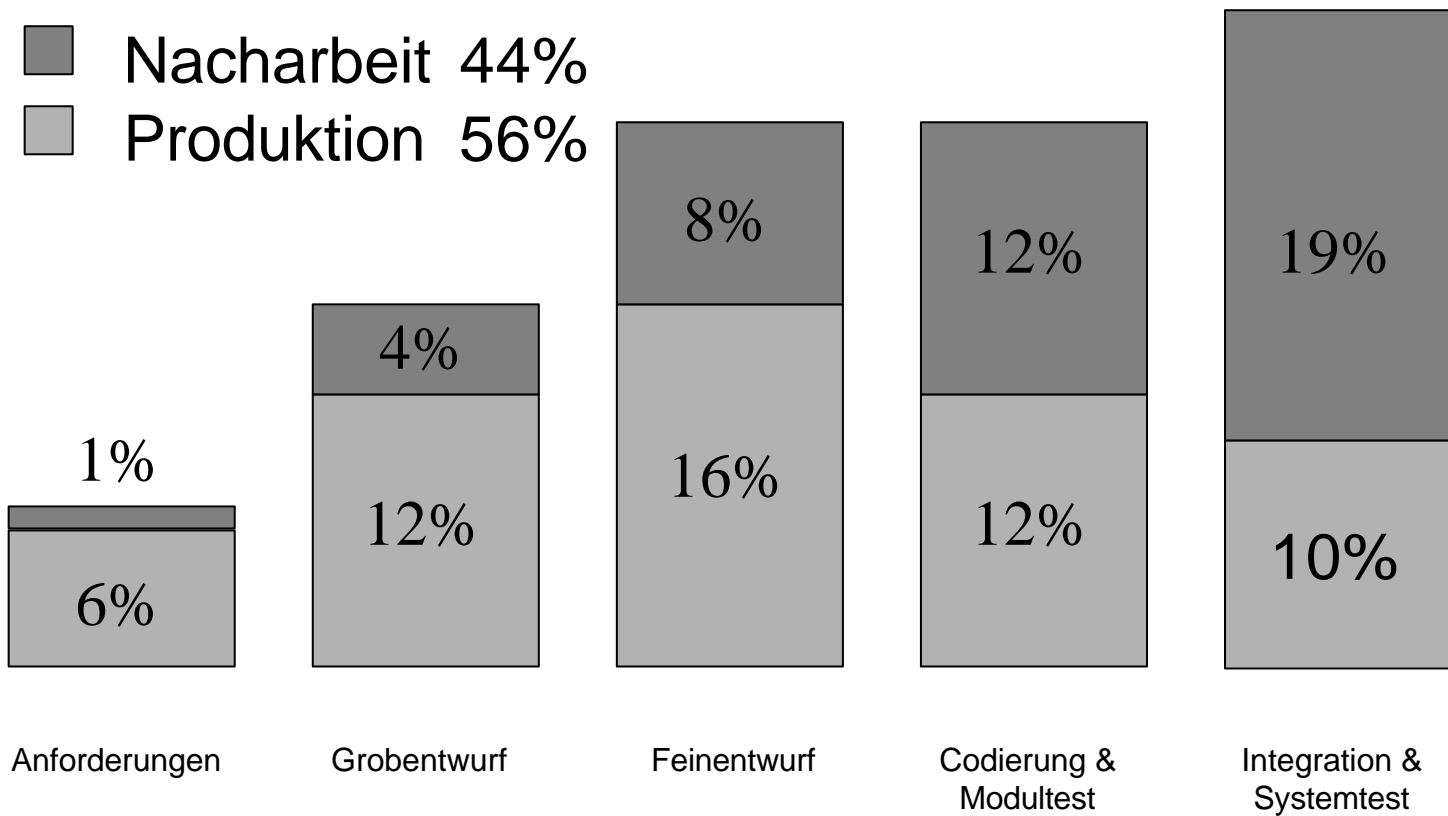
Nr.	Bezug zu o.g. Dokumenten	Reviewprotokoll	A	B	C	?	Abn./Int.	Fehler (A,B)/P
6.	1, 2	Reviewprotokoll Grobkonzept	14	27	15	34	A	1.1
7.	4	Reviewprotokoll Fachkonzept Iteration1 Teil 1	2	12	12	6	A	1.6
8.	5	Reviewprotokoll Fachkonzept Iteration1 Teil 2	5	38	47	16	A	2.5
9.	Siehe 3	Reviewprotokoll Systemarchitektur	20				I	

— andere Firmen

- AT&T Bell Laboratory
 - Projekt mit 200 Mitarbeitern setzt Reviews ein => Steigerung der Produktivität um 14%, der Qualität um Faktor 10.
- Aetna Insurance Company
 - Durch Reviews wurden 82% der Fehler in einem COBOL Programm gefunden, die Produktivität um 25% gesteigert.
- Weiteres COBOL – Beispiel (Gilb, Software Metrics)
 - In der Entwicklung wurden 80% der Fehler gefunden, die Produktivität wurde um 30% gesteigert.

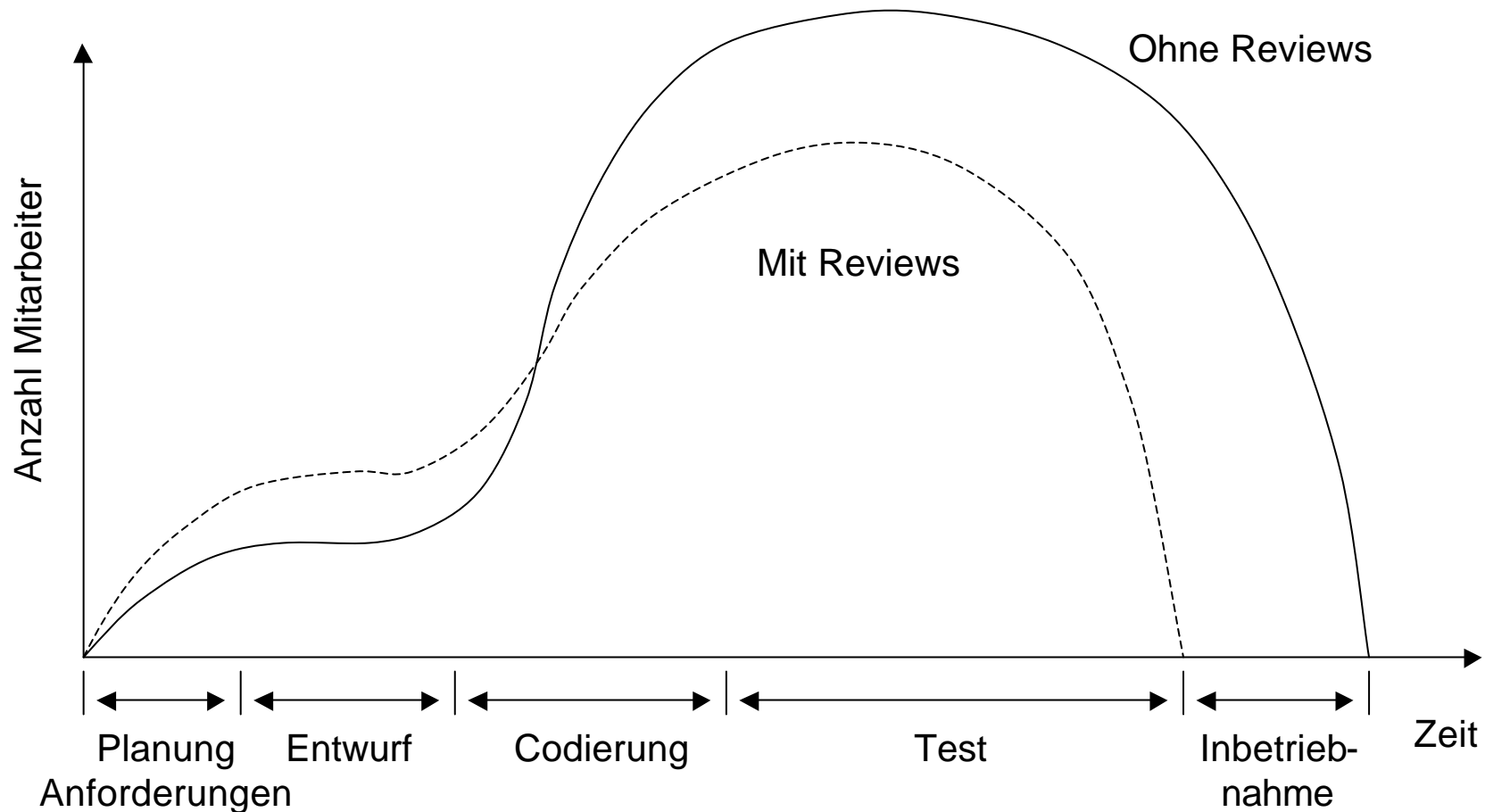
Quelle: Humphrey 1989, Managing the Software process, S. 186/187

Anteil Nacharbeit am Gesamtaufwand



Quelle: Wheeler 1996, Software Inspection: An Industry best Practice, S.9

— Einsparung durch Reviews (Zeit, Kosten)



Quelle: Fagan 1986

— Einige Tipps für ein erfolgreiches Review

- Auf die Zusammensetzung der Review-Gruppe kommt es an!
 - Autor, Prüfer, Review-Leiter
 - Fachexperten mit einbinden
 - Methodenexperte
- Gute Checklisten
- Vor dem Meeting die Anmerkungen der Prüfer zuschicken lassen
- “Formalkram” vorab erledigen
- Problem und Lösung trennen

— Fazit

- Reviews erfordern keine hohen Investitionen
- Reviews müssen geplant werden
- Reviews lassen sich über den gesamten Software-Entwicklungsprozess hinweg anwenden
- Reviews helfen bereits in frühen Phasen des Software-Entwicklungsprozesses Fehler zu finden
- Reviews sind keine Diskussionsrunden um Probleme zu lösen
- Durch Reviews werden der Know-how-Transfer und gegenseitiges Verständnis in der Organisation gefördert
- Reviews bergen auch gewisse Risiken (Etablieren einer Kultur!)