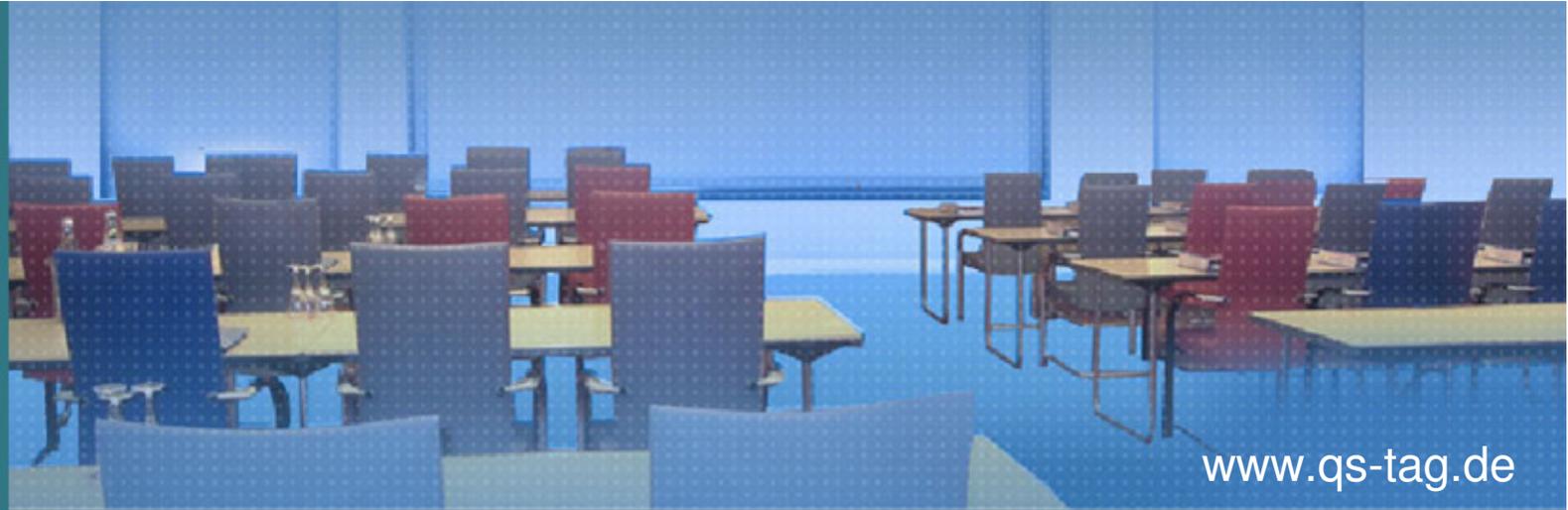


# Erfolgsgeheimnisse im Testmanagement

Auf die Testmanager kommt es an!



## Mit Stichproben-Reviews die Dauer von Testphasen vorhersagen

**Peter Rösler, Maud Schlich**  
**ReviewConsult**



ReviewConsult

# Tutorial Stichproben-Reviews

- **Einführung in Reviews und Inspektionen**
- Problematik  
Testaufwandsschätzung,  
Rework-Aufwand und Fehlerdichte
- Schätzung über Stichproben-Reviews
- Tipps für erfolgreiche Reviews



# Reifegradmodell CMM (Capability Maturity Model)

Schon CMM (eines der ersten Reifegradmodelle) forderte Reviews:

Level	Fokus	Schlüsselprozessbereiche
Level 5 Optimierend	Ständige Prozess- Optimierung	Prozessänderungsmanagement Technologieänderungsmanagement Fehlervorbeugung
Level 4 Beherrscht	Produkt- und Prozess- Qualität	Quantitatives SW-Qualitätsmanagement Quantitatives Prozessmanagement
Level 3 Definiert	Prozess- Standards	Prozess-Fokus (org.weit), Prozess-Definition (org.weit), <b>Peer-Reviews</b> , Trainingsprogramme Koordination von Entwicklungsgruppen, SW-Produkt- Entwicklung, Integriertes Projektmanagement
Level 2 Wiederholbar	Projekt- Management	Anforderungsmanagement, SW-Projektplanung SW-Projektverfolgung und -übersicht SW-Zulieferermanagement, SW-Qualitätssicherung SW-Konfigurationsmanagement
Level 1 Initial	"Helden"	---

Quelle: [www.software.org/  
quagmire/descriptions/swcmm.asp](http://www.software.org/quagmire/descriptions/swcmm.asp)

# Einführung in Reviews und Inspektionen

- **Einleitung / Begriffe**
- Rollen der Teilnehmer
- Phasen eines Reviews



## Begriffe (1)

- Im Folgenden verwenden wir die Begriffe “Review” oder „Software-Review“ synonym zu “Inspection” / “formale Inspektion” / “Fagan/Gilb style inspection”.
- *Achtung*: in der englischsprachigen Literatur bezeichnet “Review” oft eine zur “Inspection” deutlich abgegrenzte analytische QS-Maßnahme (s. nächste Folie).
- In diesem Sinne geht es im Vortrag ausschließlich um “Inspections”!

# Review-Varianten (nach Graham)

**Walkthrough:** [Presentation Review]      Aktivität: Verstehen

- Autor leitet die Gruppe durch das Dokument und die dahinter stehenden Gedankenprozesse, so dass alle dasselbe verstehen und Einigkeit erzielt werden kann, was zu ändern ist.

**Management-Review:** [Projektstatus-Review]      Aktivität: Entscheiden

- Gruppe diskutiert das Dokument und trifft Entscheidungen über den Inhalt, z.B. wie etwas gemacht werden soll, Ja/Nein-Entscheidungen.

**Inspektion:**      Hauptaktivität: Fehler finden

- Formale Einzel- und Gruppen-Prüfaktivität nach genau festgelegten Regeln, unter Verwendung von Vorgängerdokumenten und Standards.

Quelle: Dorothy Graham

## Begriffe (2)

“major defect” (im Gegensatz zu “minor defect”)

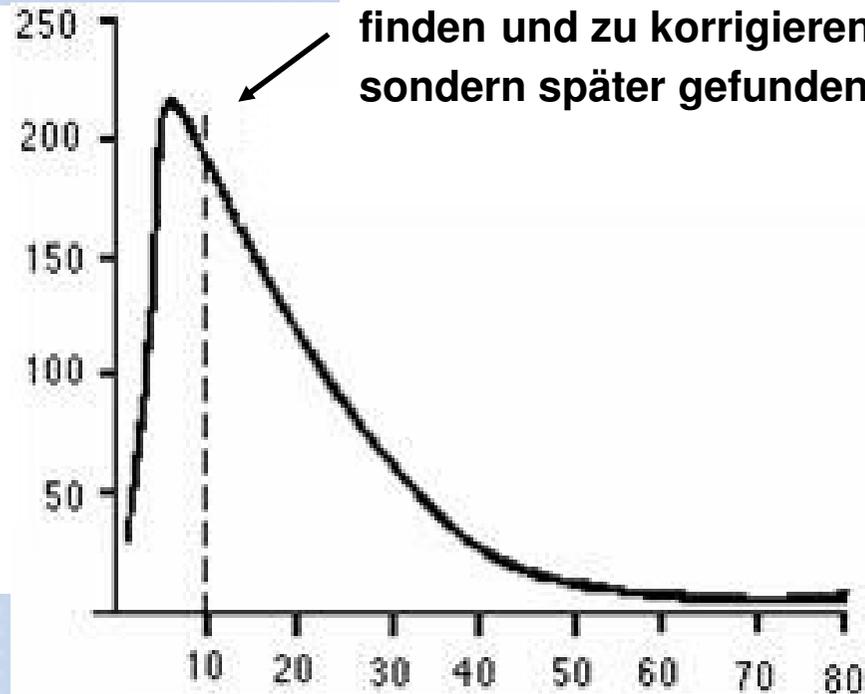
- Fehler, der möglicherweise erheblich höhere Kosten verursacht, wenn er später gefunden wird als jetzt

andere übliche Definitionen:

- Fehler, der durch Tests gefunden werden kann
- Fehler, der durch den Benutzer gefunden werden kann

# Kosten eines Major defects im Test oder im Betrieb

Anzahl Fehler aus 1.000 zufällig ausgewählten Major defects (aus allen Dokument-Typen)



Durchschnittlicher Aufwand, einen Major defect zu finden und zu korrigieren, wenn er nicht im Review, sondern später gefunden wird: 9,3 h.

Zum Vergleich: es kostet ca. nur 1 h Aufwand, um einen Major defect mit Reviews zu finden und zu korrigieren.

Geschätzter Aufwand, den Fehler im Test oder im Betrieb zu finden und zu korrigieren (Bandbreite 1 bis 80 h)

Quelle: MEL Inspektions-Daten  
Juni 1989 - Januar 1990,  
"Software Inspection", p 315

## Begriffe (3)

für Programme:

### NLOC

- non-commentary lines of code "Nichtkommentarzeilen", s. Notizansicht

für Textdokumente:

### “Seite”

oft auch "Netto-Seite" genannt (im Gegensatz zu "Druck-Seite")

- 300 “non-commentary” words

### “Inspektionsrate“

- Prüfgeschwindigkeit, wird bei Programmen in NLOC / h und bei Textdokumenten in Seiten / h angegeben.

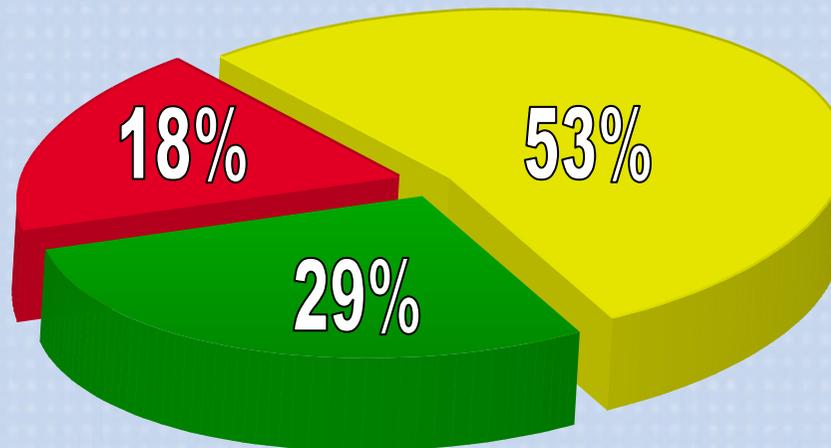
## Erfahrungen anderer Firmen

- Ein Projekt mit 200 Entwicklern bei AT&T Bell Laboratory führten mehrere Verbesserungen ein, u.a. Reviews. Die Produktivität stieg um 14% und die Qualität um Faktor 10.
- Aetna Insurance Company: Reviews fanden 82% der Fehler in einem COBOL-Programm, die Produktivität stieg um 25%. 
- Weiteres COBOL-Beispiel (Gilb, Software Metrics): 80% der Entwicklungsfehler wurden durch Reviews entdeckt, die Produktivität stieg um 30%.

Weitere Beispiele siehe  
Notizseite in der PowerPoint-Datei

Source: Humphrey 1989,  
Managing the software  
process, p186/187

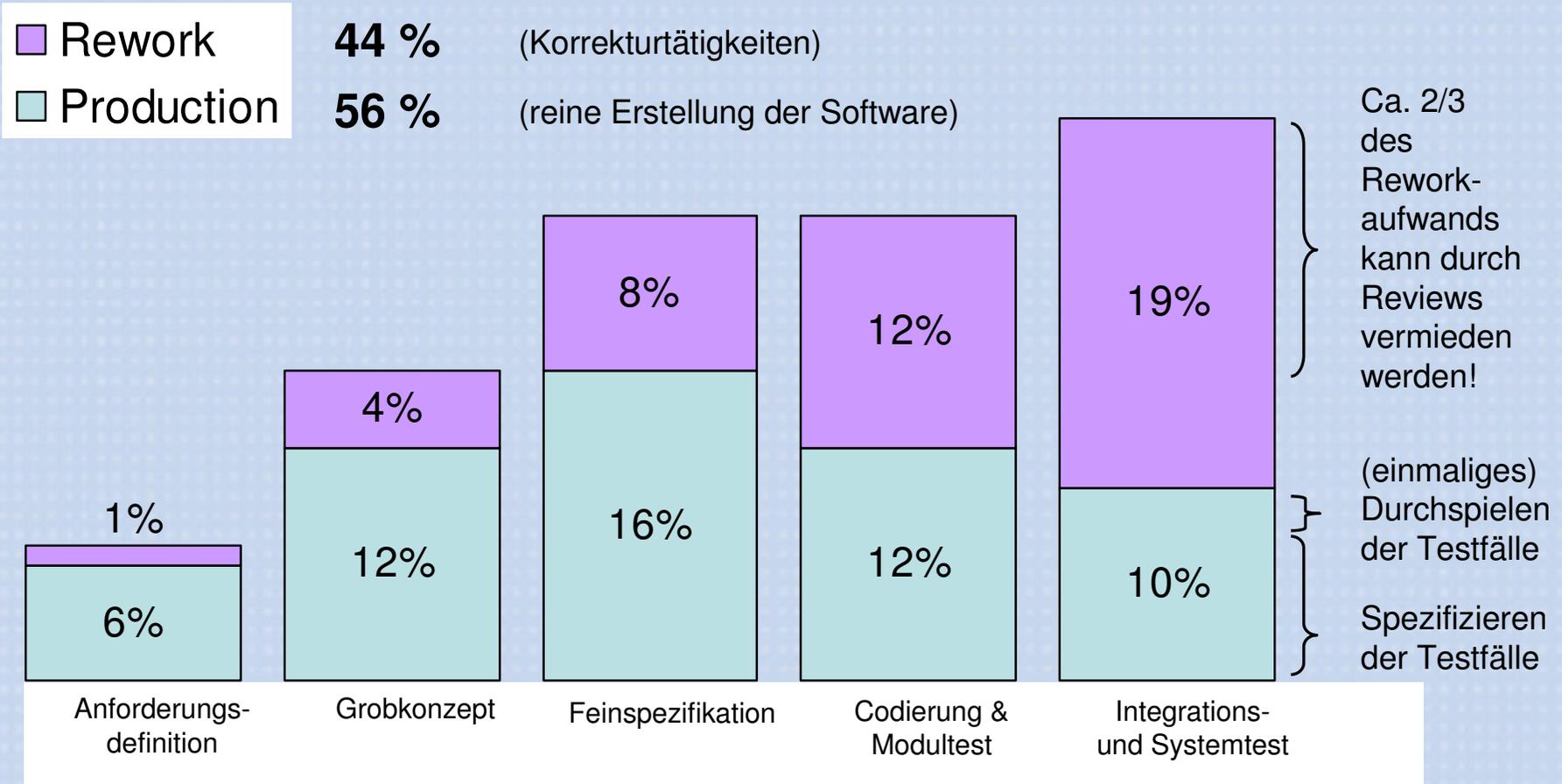
# Erfolgreiche Projekte sind nicht der Normalfall



Mit richtig durchgeführten Reviews wäre der Anteil erfolgreicher Projekte viel höher!

■ Abgebrochen    ■ Ziel verfehlt    ■ Erfolgreich

# Anteil von Korrekturtätigkeiten am Gesamtaufwand



Quelle: Wheeler 1996,  
Software inspection: an industry  
best practice, p 9

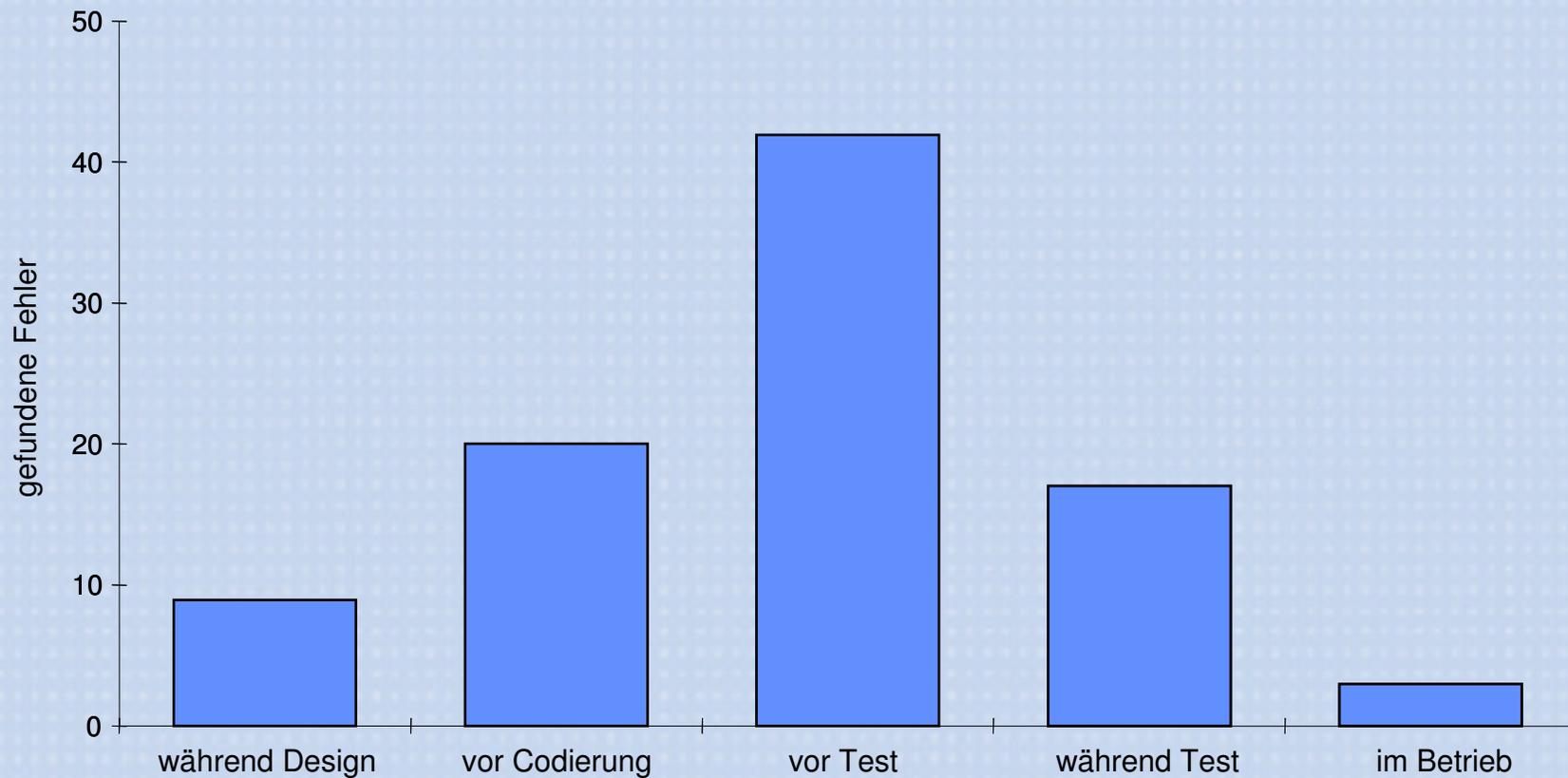
# Nutzen von Reviews

- geringere Entwicklungskosten (25-35%)
- kürzere Entwicklungszeiten (25-35%)
- geringere Wartungskosten (Faktor 10-30)
- höhere Zuverlässigkeit (10-100 mal weniger Fehler)



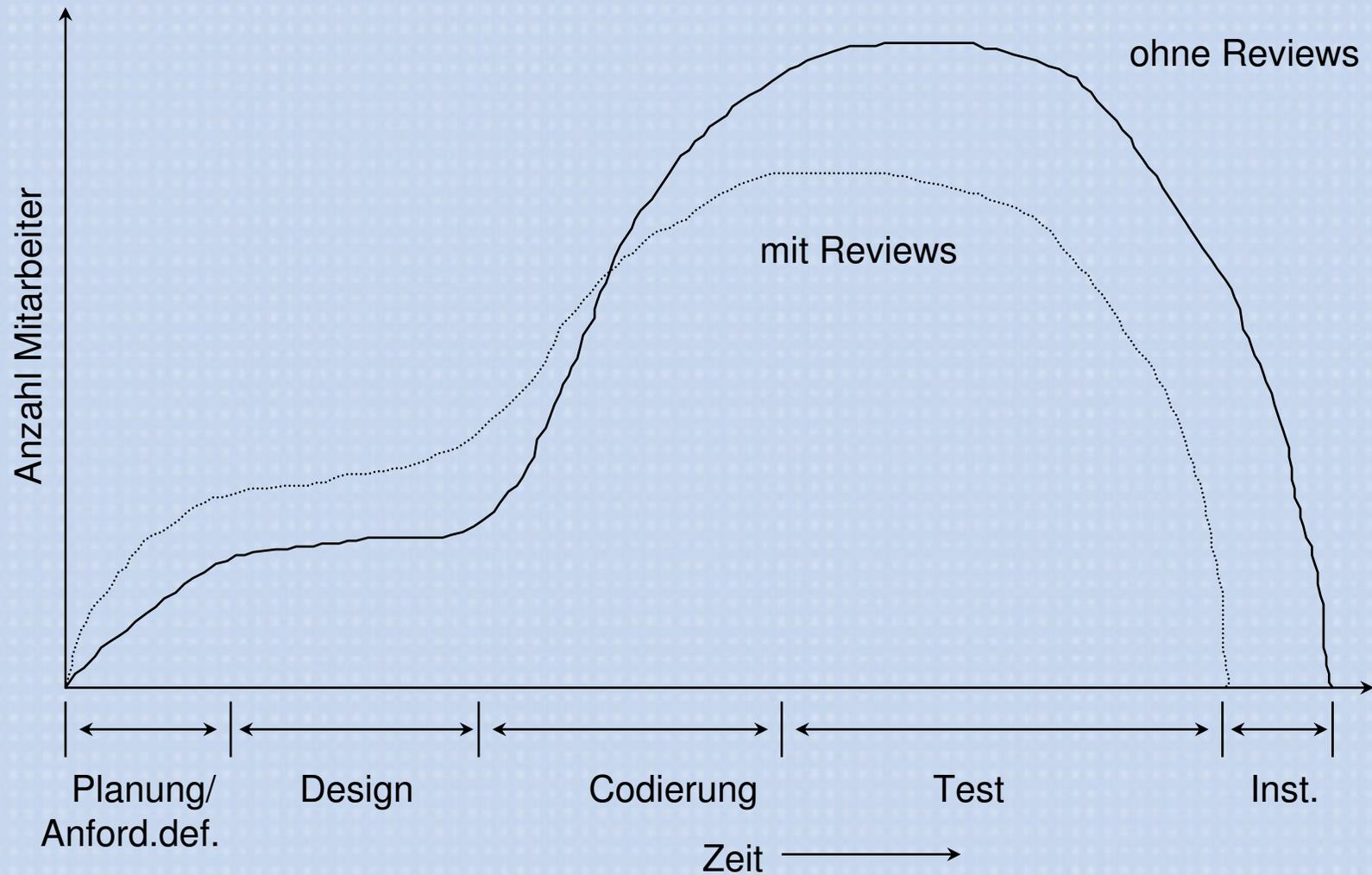
Source: Gilb1988, Principles of Software Engineering Management, Table 12.2  
vgl. auch SI p 24

# Zeitpunkt der Fehlerentdeckung (mit Reviews)



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

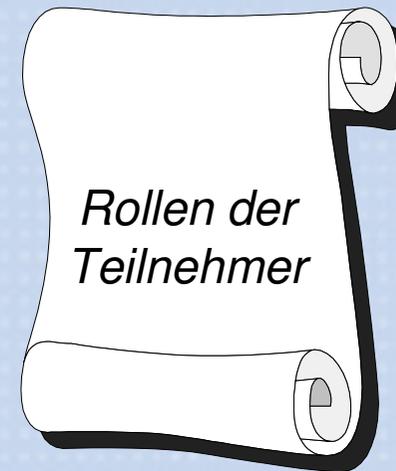
# “Mitarbeiterberg” mit und ohne Reviews



Source: Fagan 1986  
Schema

# Einführung in Reviews und Inspektionen

- Einleitung / Begriffe
- **Rollen der Teilnehmer**
- Phasen eines Reviews



# Rollen der Teilnehmer

- Moderator
- Autor
- Protokollant
- Gutachter/Reviewer
- Vorleser/Reader (nur wenn „double checking“ gemacht wird)

Ein Teilnehmer kann mehrere Rollen übernehmen.

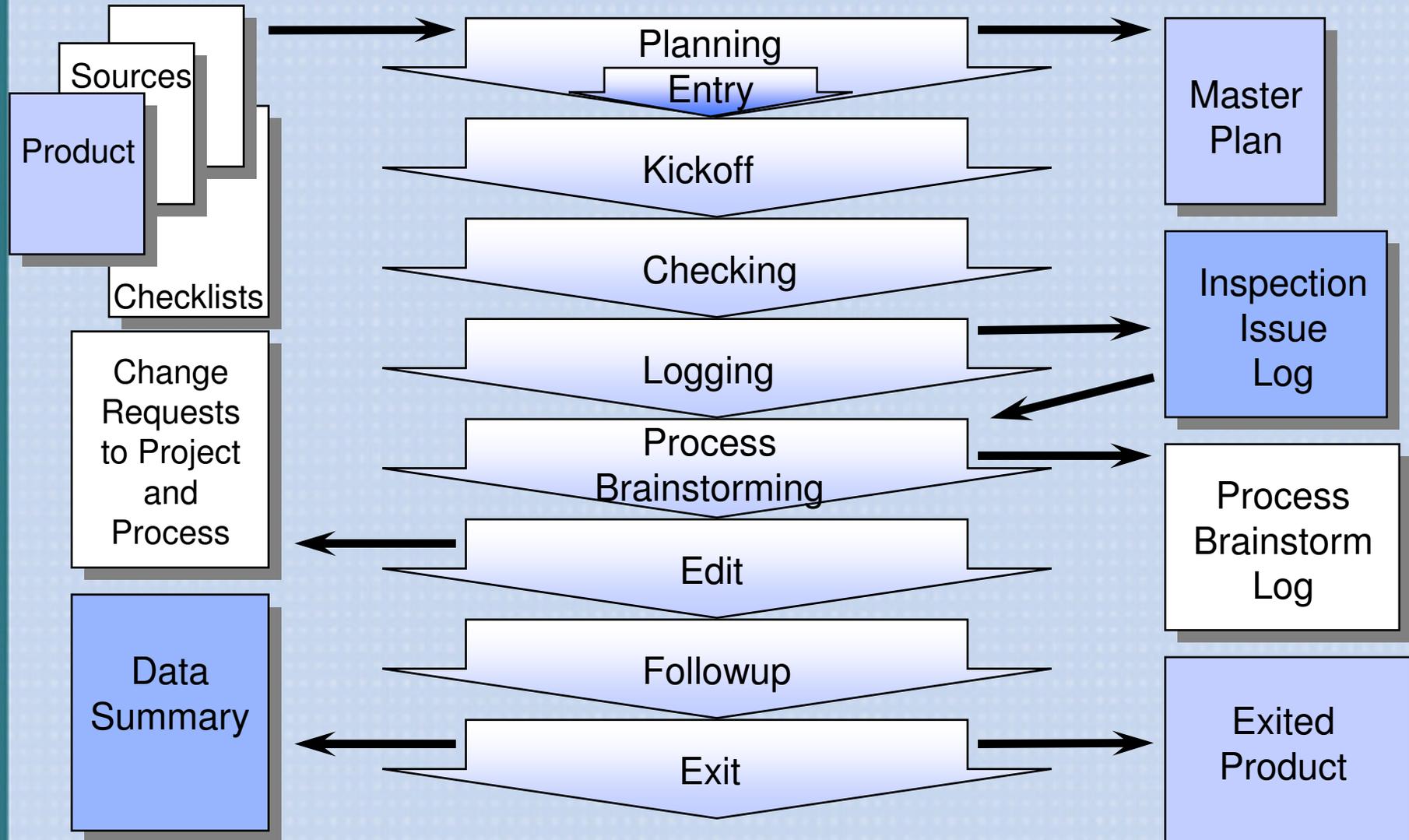
Einzige Einschränkung: der Autor darf zusätzlich höchstens die Rolle eines Gutachters übernehmen.

# Einführung in Reviews und Inspektionen

- Einleitung / Begriffe
- Rollen der Teilnehmer
- **Phasen eines Reviews**



# Overall Process Map



Source: Tom Gilb,  
Team Leader Course

Im Original-Tutorial folgten 13 Folien

ROS 06.11.2010

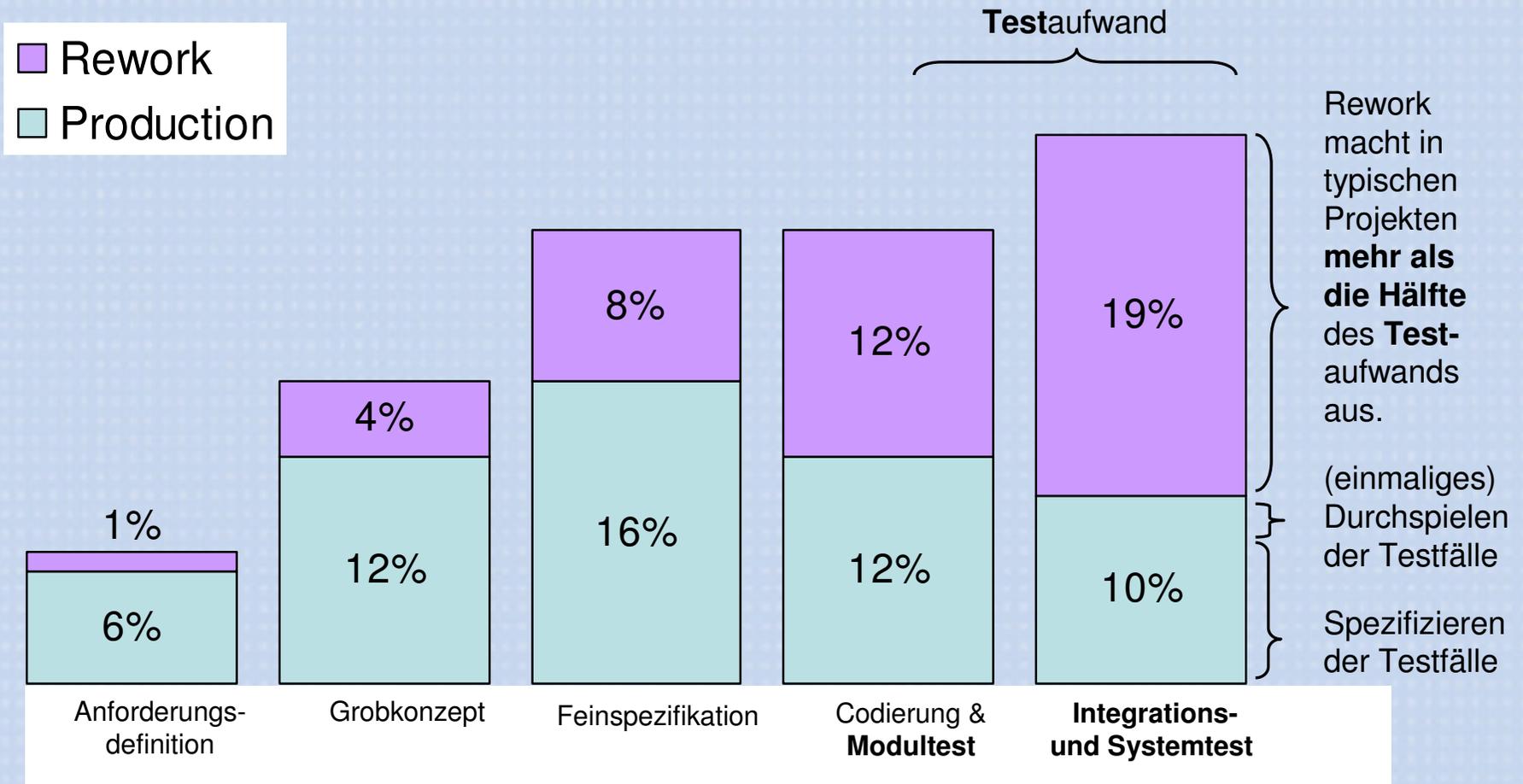
# Tutorial Stichproben-Reviews

- Einführung in Reviews und Inspektionen
- **Problematik**  
**Testaufwandsschätzung,**  
**Rework-Aufwand und Fehlerdichte**
- Schätzung über Stichproben-Reviews
- Tipps für erfolgreiche Reviews



# Anteil von Rework am Testaufwand

selbe Daten wie aus Folie 12, aber mit Blickwinkel "Testaufwand" statt "Gesamtaufwand"



Quelle: Wheeler 1996,  
Software inspection: an industry  
best practice, p 9

# Reworkaufwand und Fehlerdichte

- Rework sind alle Arbeiten, die aus Major Defects resultieren, z.B.
  - Lokalisieren von gefundenen Fehlern
  - Korrigieren der Dokumente
  - Ändern von Testspezifikationen
  - Wiederholen von Testläufen, etc.

(Minor Defects tragen definitionsgemäß nur unerheblich zum Reworkaufwand bei, s. Folie 7.)

- Der **Reworkaufwand** ist damit per Definition (zumindest annähernd) proportional zur **Anzahl der Major Defects** in den Dokumenten, also zur **Fehlerdichte** der Dokumente.

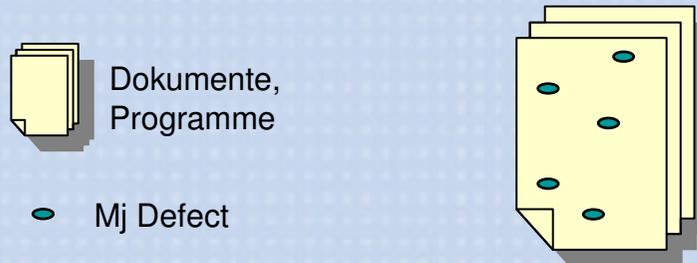
Fehlerdichten werden in Mj / p bzw. Mj / 100 NLOC angegeben, z.B.

Anforderungsspezifikation AnfSpez-CSS.doc: 3,5 Mj / p

Programm infcomp.c : 2,0 Mj / 100 NLOC

# Die Fehlerdichte steuert den Reworkaufwand

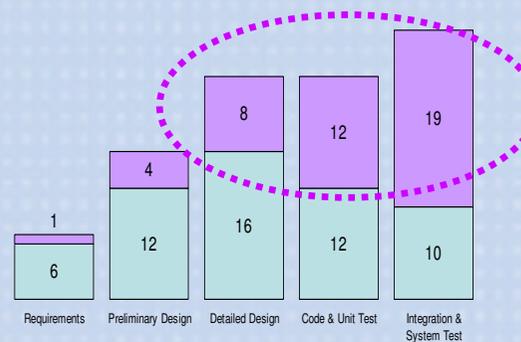
typisches Projekt



Die Anzahl der Major Defects entscheidet ...

Rework

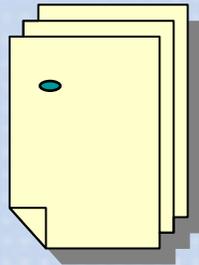
Production



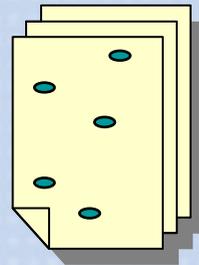
... wie hoch der Reworkaufwand ist.

# Die Fehlerdichte ist stark variabel ...

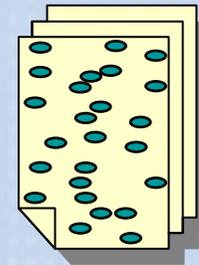
Projekt A,  
5 mal weniger Mj Defects



typisches Projekt



Projekt C,  
5 mal mehr Mj defects



← 3 Projekte mit um Faktor 25 unterschiedlichen Fehlerdichten →

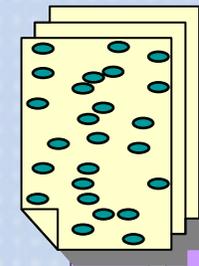
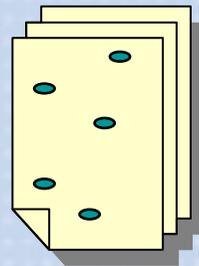
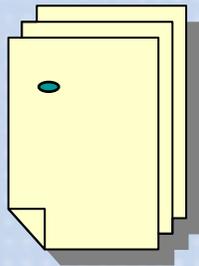
Die Reviewtechnik-Literatur berichtet sogar von Projekten, die im Vergleich zu anderen Projekten eine um Faktor 100 (!) geringere Fehlerdichte hatten.

# ... die Folgen für den Aufwand sind enorm

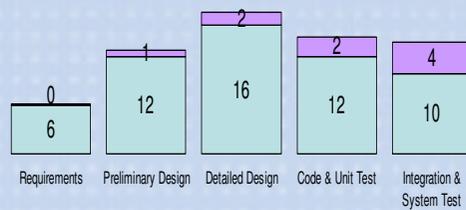
5 mal weniger Mj Defects

typisches Projekt

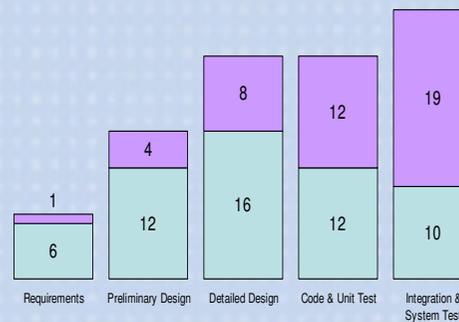
5 mal mehr Mj Defects



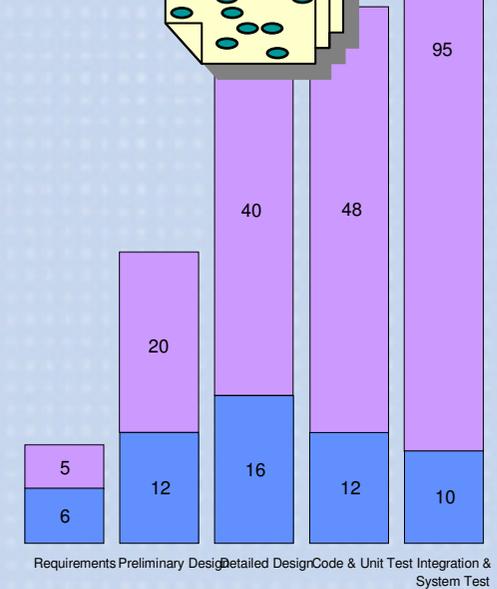
█ Rework  
█ Production



65 Wo



100 Wo



276 Wo

Gesamtaufwand in Wochen

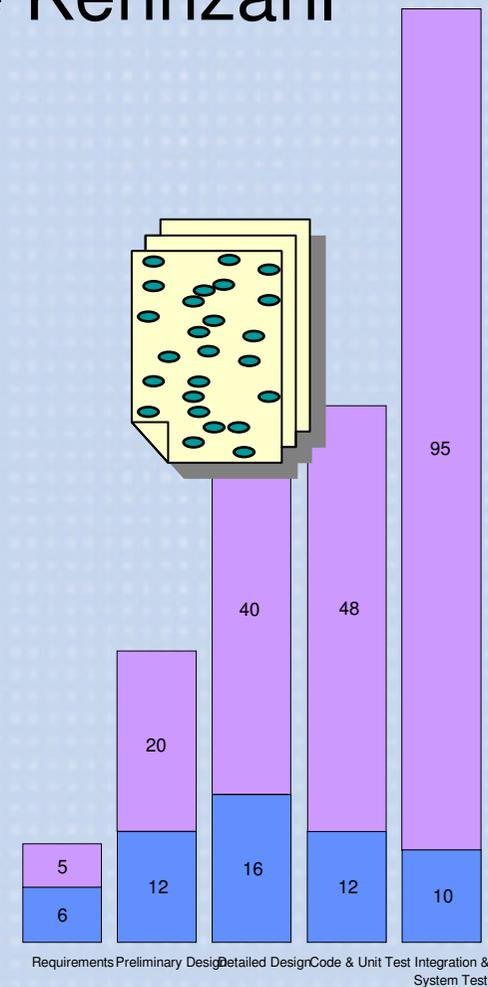
# Fehlerdichte als entscheidende Kennzahl

Dieses Projekt wird mit hoher Wahrscheinlichkeit nicht termingerecht fertig oder wird sogar ganz abgebrochen.



- Die Fehlerdichte gehört zu den wichtigsten Kennzahlen, die erfolgreiche von gescheiterten Projekten unterscheiden.
- Die Fehlerdichte ist für Projektleiter und Testmanager erst einmal unbekannt. Sie zu ermitteln ist eine (Qualitäts-) Managementaufgabe mit höchster Priorität!

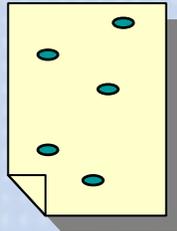
Sie zu senken ist noch wichtiger!  
(Aber nicht Thema dieses Vortrags.)



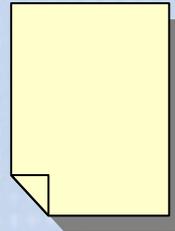
276 Wo

# Fehlerdichte und Effektivität eines Reviews

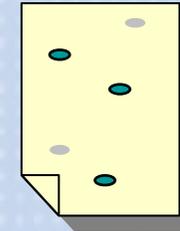
So viele Mj Defects  
sind vorhanden.



Vor dem Review sieht das  
Dokument fehlerfrei aus.  
Die Fehlerdichte ist unbekannt.



Das Review findet einen  
Teil der Mj Defects:



Formeln:

$$\text{Effektivität} = \frac{\text{gefundene Mj}}{\text{alle Mj}}$$

$$\text{unentdeckte Mj} = \text{gefundene Mj} * (1/\text{Effektivität} - 1)$$

Die Mindest-Fehlerdichte ist  
damit bekannt. Offen ist:  
wie viele Mj Defects sind  
unentdeckt, also wie hoch  
ist die Fehlerdichte genau?

# Die Effektivität muss abgeschätzt werden

$$\text{unentdeckte Mj} = \text{gefundene Mj} * (1/\text{Effektivität} - 1)$$



Leider auch erstmal unbekannt.  
Je nach Reifegrad des Review-Prozesses  
stark streuend (von unter 3% bis über 90%).

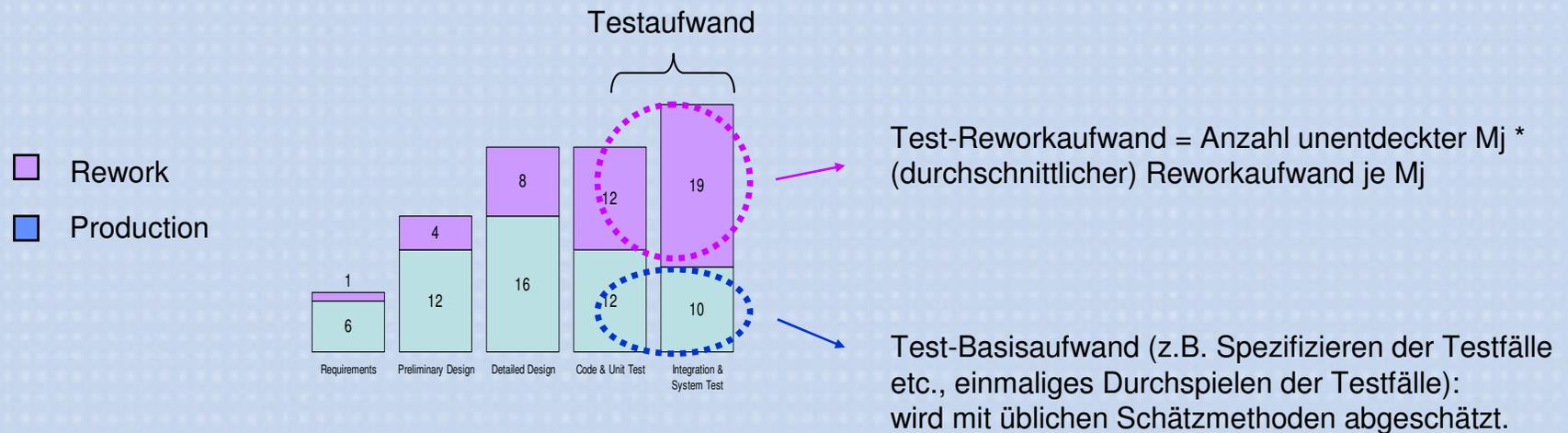
- Je genauer man die Effektivität abschätzen kann, desto genauer weiß man die Anzahl der unentdeckten Fehler und damit die Fehlerdichte.

Beispiel:

Die Effektivität des Reviews einer 20-seitigen Anforderungsspezifikation wird mit 50-66% geschätzt und es wurden durch das Reviewteam 100 Mj Defects entdeckt. Dann sind zwischen 50 und 100 Mj Defects unentdeckt, das (unkorrigierte) Dokument enthält zwischen 150 und 200 Mj Defects, die Fehlerdichte des unkorrigierten Dokuments liegt zwischen 7,5 und 10 Mj / p, die Fehlerdichte des korrigierten Dokuments liegt zwischen 2,5 und 5 Mj / p.

- Mehrere Verfahren zum Schätzen der Effektivität werden **später** vorgestellt (im nächsten Kapitel).

# Testaufwand vorhersagen



Test-Reworkaufwand = Anzahl unentdeckter Mj \*  
(durchschnittlicher) Reworkaufwand je Mj

# Reworkaufwand je Major Defect

Methoden, den Reworkaufwand je Major Defect zu bestimmen	Kommentar zur Methode
Nehmen Sie als erste grobe Schätzung an, die 9,3 h von Folie 8 gelten auch für die Mj Defects in Ihrem Projekt.	Schnell machbar, aber der Wert kann bei Ihnen auch 2 bis 3 Mal höher oder niedriger liegen.
Erfahrene Entwickler schätzen separat für jeden gefundenen Mj, in welcher Phase er sonst aufgetreten wäre und wie zeitaufwändig die Korrektur geworden wäre. Dann wird der Mittelwert gebildet.	Aufwändig, liefert aber eine relativ gute Schätzung. (Die Daten aus Folie 8 sind übrigens genau so entstanden.)
Verwenden Sie historische Daten aus eigenen ähnlichen Projekten, z.B. aus einer Fehlerdatenbank, in der auch die Korrekturaufwände erfasst sind.	Ideal (Ist Ihre Firma vielleicht sogar schon auf CMM(I) Level 4?)

# Tutorial Stichproben-Reviews

- Einführung in Reviews und Inspektionen
- Problematik  
Testaufwandsschätzung,  
Rework-Aufwand und Fehlerdichte
- **Schätzung über Stichproben-Reviews**
- Tipps für erfolgreiche Reviews



# Schätzung über Stichproben-Reviews

- **Stichproben-Reviews**

Effektivitäts-Schätzverfahren:

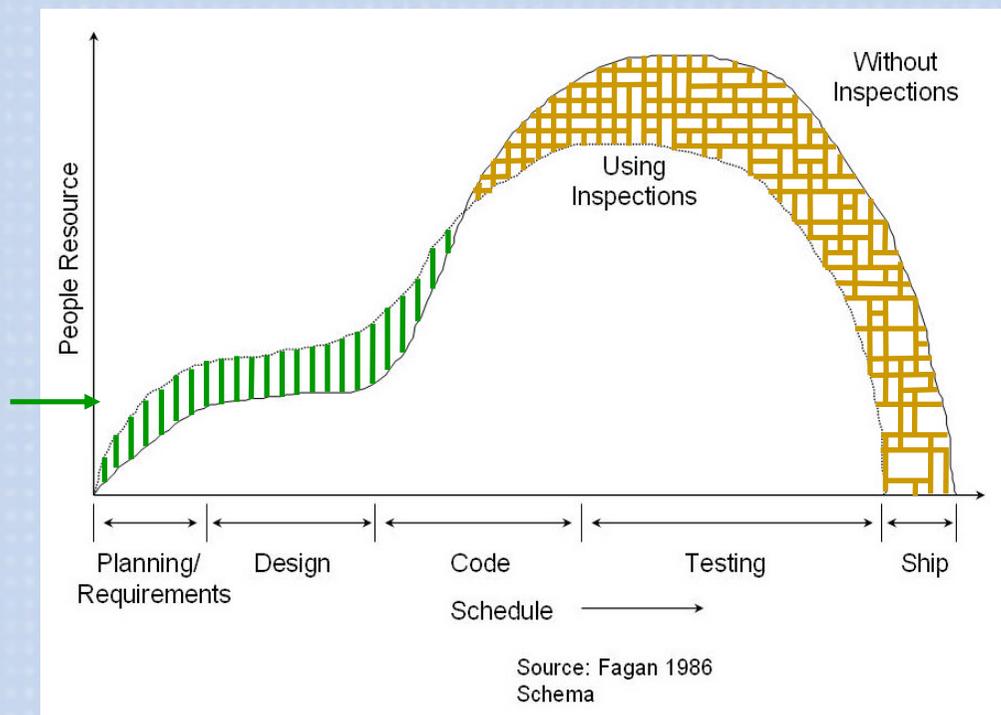
- Vergleich mit opt IR
- Capture-Recapture
- Curve Fitting
- Subjektiv durch Experten



# Aufwand für 100%ige Reviewabdeckung

Gilb: ca. 10-15% des Projektaufwands (Radice: ca. 8-20%)

**Aufwand:**  
z.B. 500 h  
verteilt auf 25  
Reviews



**Einsparung:**  
z.B. 2000 h  
eingespart durch  
250 gefundene  
Mj defects

**ROI = 4:1**

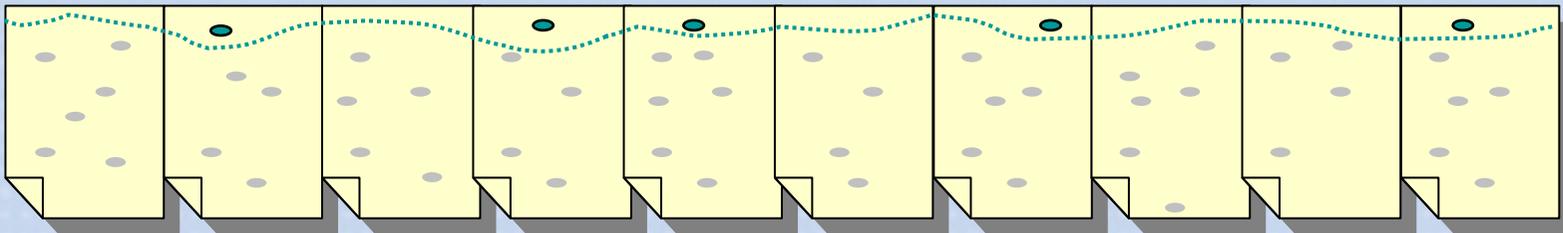
Details siehe Notizseite

## Stichproben-Reviews

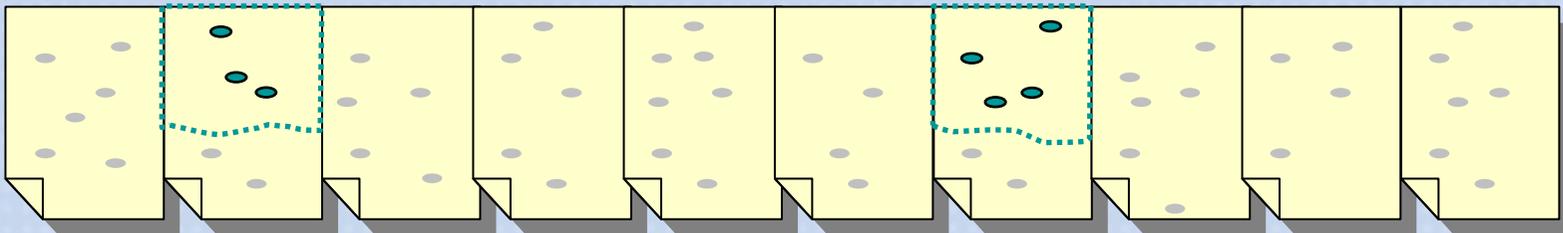
- Ein Testmanager, wenn er "nur" eine Vorschau auf den zu erwartenden Testaufwand erhalten will, wird in diesem Fall kein Budget für eine 100%ige Reviewabdeckung der Anforderungs-, Design-, und Code-Dokumente erhalten.
- Es bleibt dann nur die Möglichkeit, mit Stichproben die Fehlerdichte der Dokumente festzustellen (mit allen Unsicherheiten, die bei Stichprobenverfahren unvermeidlich sind).
- Mit Stichproben zu arbeiten, ist in der Software-Industrie eher unüblich, wurde im Zusammenhang mit Reviews aber schon im Buch "Software Inspection" (Gilb/Graham 1993) empfohlen.

# Vorteil von Stichproben

Wenn ein komplettes Dokument mit 100 oder mehr Seiten mit einem "gewöhnlichen" Review geprüft wird, wird nur ein kleiner Teil der Fehler entdeckt. Eine Abschätzung der realen Fehlerdichte ist kaum möglich.



Ein oder zwei gut ausgewählte und gründlich geprüfte Stichproben erlauben eine viel bessere Abschätzung der realen Fehlerdichte des Dokuments.



# Aufwand für eine Stichprobe

Textdokument: **1,5 Seiten**, Aufwand: **10 h**

Annahmen:

Teamgröße 3 (Autor + 2 Reviewer)

Inspektionsrate 1,5 p/h (auch Autor prüft)

Phase Checking nimmt 30% des Aufwands ein

Erwartete Effektivität: ca. 50%

The load control value times, that is once a after each start of be entered for all the following days.

For each such flight examined line by line. the condition part mat the new load control

The operator may enter (the so called genera ensures that every fli

```
void consumer ()
{
    int item;
    while (1)
    {
        if (count == 0)
            sleep();
        remove_item(&item);
        count = count - 1;
        if (count == N - 1)
            wakeup(producer);
        consume_item();
    }
}
```

Programm: **125 NLOC**, Aufwand: **10 h**

Annahmen:

Teamgröße 3 (Autor + 2 Reviewer)

Inspektionsrate 125 NLOC/h (auch Autor prüft)

Phase Checking nimmt 30% des Aufwands ein

Erwartete Effektivität: ca. 50%

Mit **agilen Inspektionen** ist der Aufwand nur ca. 3 h, siehe Notizseite

# Effektivitäts-Schätzverfahren

- Stichproben-Reviews

## Effektivitäts-Schätzverfahren:

- **Vergleich mit optimaler Inspektionsrate**
- Capture-Recapture
- Curve Fitting
- Subjektiv durch Experten



## Vergleich der realen Inspektionsrate mit der optimalen Inspektionsrate

Bei dieser Methode wird die Effektivität des Reviews eines Dokuments wie folgt abgeschätzt:

- Mit Angaben aus der Literatur oder eigenen historischen Daten wird die für diesen Dokumenttyp "optimale Inspektionsrate" geschätzt.
- Die "erreichbare Effektivität" wird abgeschätzt, also die Effektivität, die bei Einhalten der optimalen Inspektionsrate zu erwarten ist.
- Die reale Inspektionsrate des Reviewteams wird mit der optimalen Inspektionsrate verglichen. Wenn sie höher ist (also "schlampiger" geprüft wurde), wird die reale Effektivität entsprechend niedriger angesetzt als die erreichbare Effektivität.

# Optimale Inspektionsrate

Angaben aus der Literatur:

- Optimale Inspektionsrate für **Textdokumente**:  
1 bis max. 2 Seiten / h (nach Gilb/Graham, 1993)  
(Der Wert ist kontraintuitiv, Entwickler vermuten eher 5-10 Seiten / h,  
siehe aber Folien 72-81)
- Optimale Inspektionsrate für **Programme**:  
100 – 150 NLOC / h

# Erreichbare Effektivität

"Daumenwert" für erreichbare Effektivität: 50%.  
Ca. 50% der im Dokument vorhandenen Fehler werden durch das Reviewteam entdeckt, sofern die optimale Inspektionsrate eingehalten wird.

Der "Daumenwert" kann noch feinjustiert werden, wenn Sie den Reifegrad Ihres Review-Prozesses bzw. Ihrer Softwareentwicklung berücksichtigen:

Angaben von  
Michael Fagan:  
("Erfinder" der  
SW-Inspektionen)

Maturity	Effectiveness
inspections just implemented	50-60%
refined processes	75-85%
refined processes, exceptional cases	>90%
Source: Fagan 2001	

Genauere Quellenangaben  
siehe Notizseite

## Erreichbare Effektivität (2)

weitere Literaturangaben:

SW-CMM	Effectiveness
Level 1	<50%
Level 2	50-65%
Level 3	65-75%
Level 4	75-90%
Level 5	90-100%
Source: Radice 2002	

Die Anzahl der Reviewer wird in den Literaturstellen nicht berücksichtigt, vermutlich weil sie die Effektivität weit weniger beeinflusst als es z.B. die Inspektionsrate vermag.

Eine Firma berichtete, dass 4 oder mehr Reviewer bei Code-Reviews nur unwesentlich mehr Fehler fanden als 3 Reviewer. Empfehlung: feinjustieren Sie die Effektivität nur dann anhand der Anzahl der Reviewer, wenn es nur sehr wenige Reviewer sind (einer oder evtl. zwei).

immature inspection process	Effectiveness
default estimation (for all types of document)	30%
mature inspection process	Effectiveness
source code	60%
pseudocode	80%
module and interface spec.	88%
Source: Gilb, SI p23, p383, Lindner 1992	

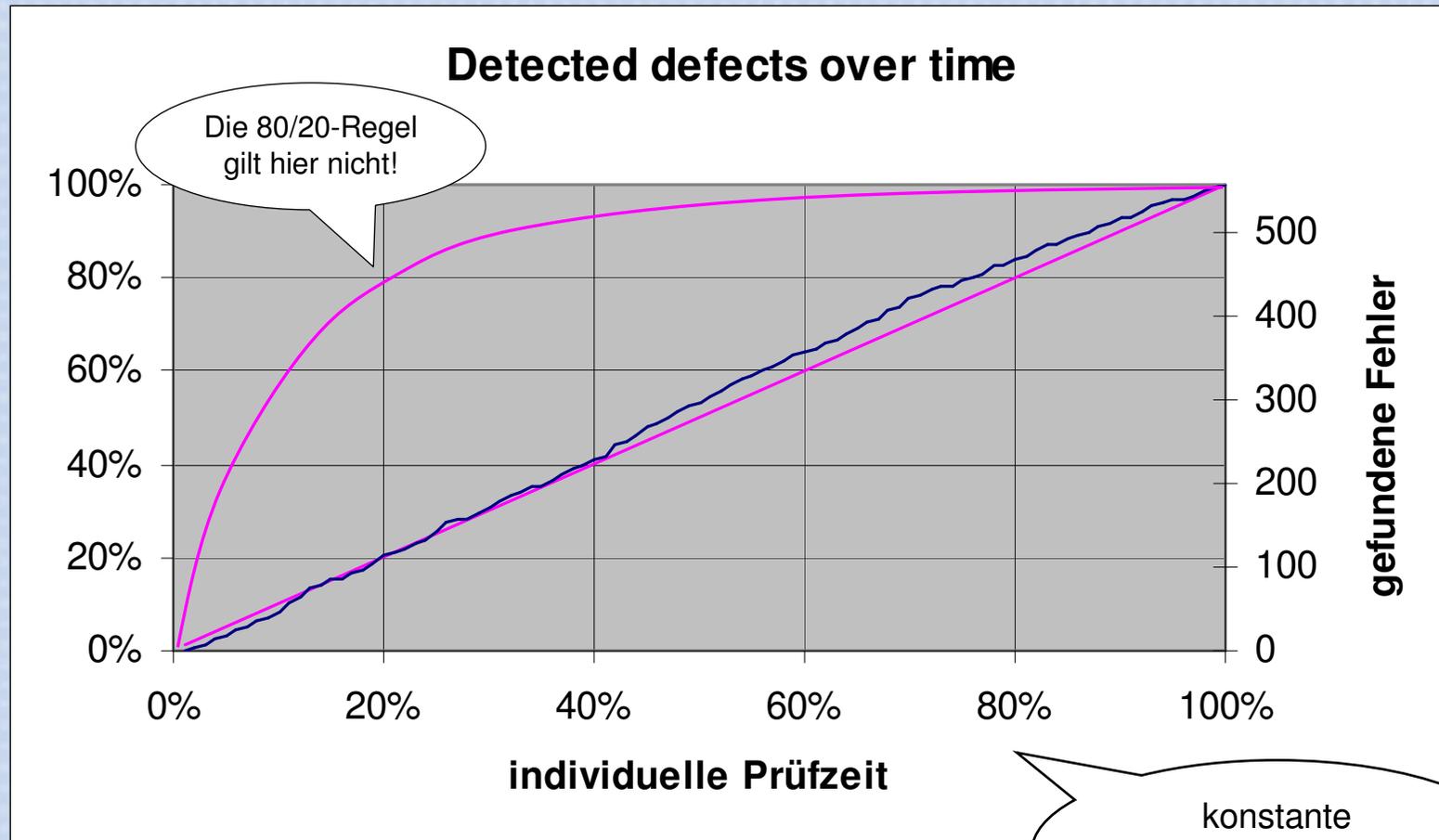
Genauere Quellenangaben  
siehe Notizseite

## Berücksichtigen der realen Inspektionsrate

- Wenn Ihr Reviewteam die optimale Inspektionsrate eingehalten hat, dann setzen Sie als reale Effektivität natürlich die erreichbare Effektivität an.
- Wenn aber zu schnell geprüft wurde, rechnen Sie die Effektivität **linear** herunter.
- "Linear", denn es gibt Hinweise, dass in der Phase Individual Checking die Fehler zeitlich ungefähr gleichverteilt gefunden werden und es daher eine **konstante Fehlerentdeckungsrate** gibt, mit der Folge:  
*„Das Verfehlen der optimalen Inspektionsrate um Faktor  $x$  bewirkt ein Sinken der Effektivität um Faktor  $x$ “ ( $x \geq 1$ ).*

(Ausführliche Begründung siehe [www.reviewtechnik.de/Reviewtechnik\\_GI\\_080923.rtf](http://www.reviewtechnik.de/Reviewtechnik_GI_080923.rtf))

# Konstante Fehlerentdeckungsrate in Phase "Individual Checking"



Kumuliertes Histogramm für 28 Reviewer

konstante  
Fehlerentdeckungs-  
rate!

Source: Peter Rösler,  
[www.reviewtechnik.de](http://www.reviewtechnik.de), 2006  
s.a. Notizansicht der Folie

# Beispiel

- Aus einer 50-seitigen Anforderungsspezifikation werden 3 Seiten geprüft, die nach Angaben von Autor und Projektleiter eine gute Stichprobe darstellen.
- Da keine historischen Daten vorliegen, werden 1,5 Seiten / h als optimale Inspektionsrate (also 2 h Prüfzeit) und eine erreichbare Effektivität von 50% angenommen.
- Die Reviewer fanden insgesamt 3 Mj Defects, also 1 Mj Defect / Seite. Der Mittelwert der Prüfzeiten lag bei 60 Minuten, was einer realen Inspektionsrate von 3 Seiten / h entspricht.
- Weil die Reviewer durchschnittlich um Faktor 2 zu schnell geprüft hatten, wird als reale Effektivität  $50\% / 2 = 25\%$  angenommen. Die Fehlerdichte der Anforderungsspezifikation liegt also bei ca. 4 Mj / Seite. In den 50 Seiten sind daher ca. 200 Mj verborgen.

## Beispiel (2)

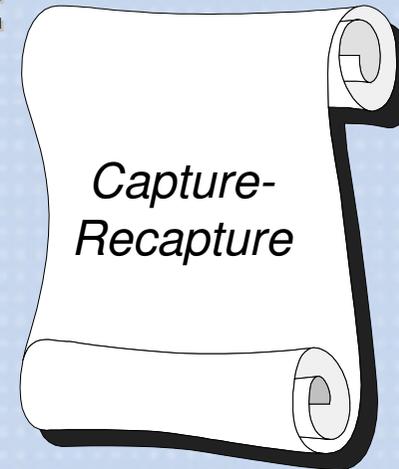
- Erfahrene Entwickler des Projektes vermuten, dass ein Anforderungsfehler später in den Testphasen ca. 10 h Aufwand verursacht. Daher wird der Reworkanteil der Testphasen, der durch Anforderungsfehler verursacht ist, ca.  $200 * 10 = 2.000$  h betragen (dazu kommt noch der durch Design- und Codierungsfehler verursachte Reworkanteil).
- Testmanager und Projektleiter sind durch die 2.000 h beunruhigt. Da sie vermuten, dass die Genauigkeit der obigen Schätzung bestenfalls bei Faktor 2 liegt, werden noch 2 weitere Stichproben-Reviews durchgeführt und die Fehlerdichte mit einem genaueren Verfahren (z.B. mit Capture-Recapture) ermittelt.

# Effektivitäts-Schätzverfahren

- Stichproben-Reviews

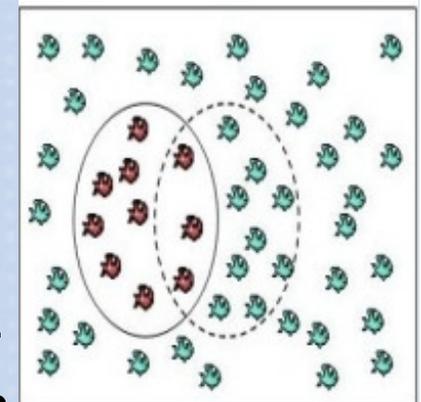
## Effektivitäts-Schätzverfahren:

- Vergleich mit optimaler Inspektionsrate
- **Capture-Recapture**
- Curve Fitting
- Subjektiv durch Experten



# Capture-Recapture

- Bei dieser Methode wird ermittelt, wie viele Fehler von zwei (oder mehreren) Reviewern entdeckt wurden ("**Doubletten**"). Aus diesen Daten wird dann die Effektivität des Reviews geschätzt.
- Diese Methode ist aus der Biologie bekannt und wird verwendet, um die Größe von Tierpopulationen zu ermitteln.
- In der Softwareentwicklung kann Capture-Recapture ganz analog eingesetzt werden und wurde als Bestandteil von **TSP** (Team Software Process) von **Watts Humphrey** vom Software Engineering Institute vorgestellt.

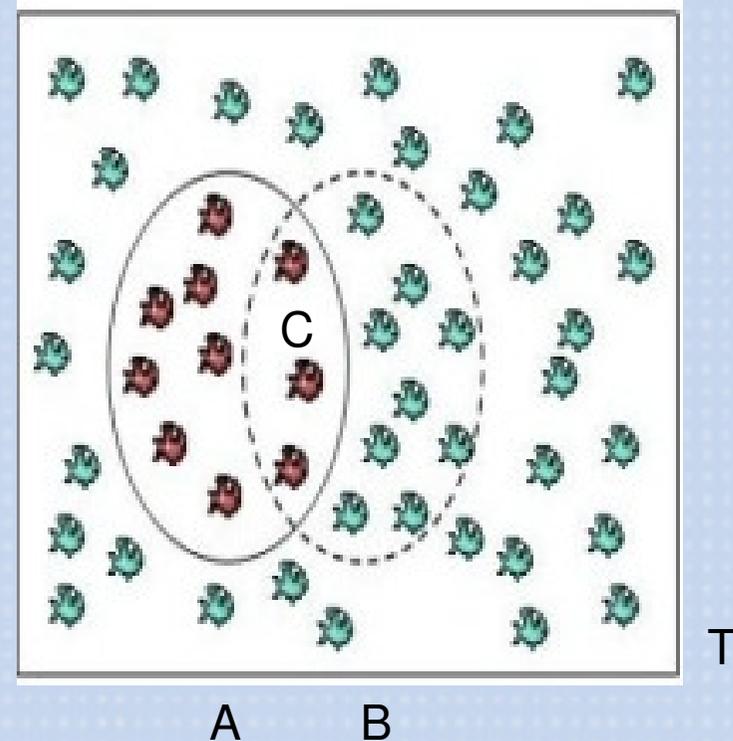


# Ein Teich mit unbekannter Anzahl von Fischen

- Fang A ergibt 10 Fische, die markiert und wieder freigelassen werden.
- Fang B (irgendwann später) ergibt 12 Fische, darunter 3 markierte (C).
- Gesamtzahl der Fische:

$$T = A * B / C$$
$$10 * 12 / 3 = 40 \text{ Fische}$$

(Real sind es 50 Fische. So ist das eben mit Schätzmethoden.)



# Formeln

- Anzahl gefundener Fehler (defects found)

$$F = A + B - C$$

- Anzahl aller Fehler (total defects)

$$T = A * B / C \quad (\text{Schätzung})$$

- Effektivität

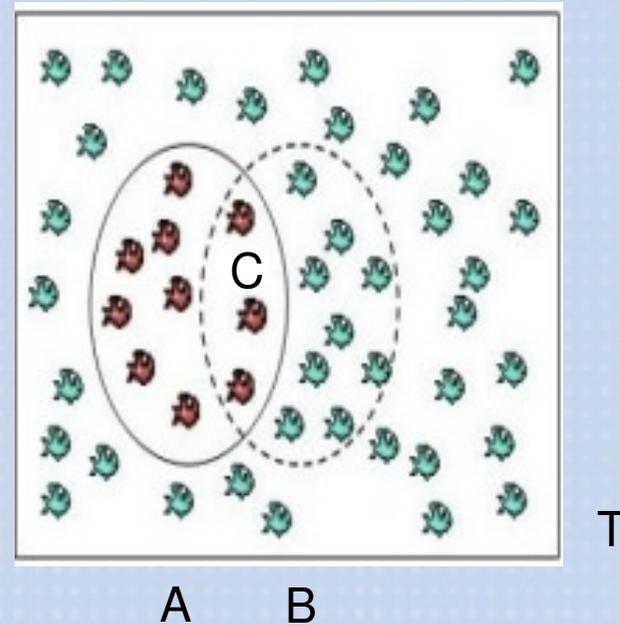
$$E = F / T \quad (\text{Schätzung})$$

Wobei:

A = Anzahl Fehler, die von  
Reviewer A gefunden wurden

B = Anzahl Fehler, die von  
Reviewer B gefunden wurden

C = Anzahl Fehler, die von  
beiden gefunden wurden



## Tipps zur Anwendung

### Feinjustierung der Schätzung:

Wenn beide Reviewer unterschiedliches Spezial-Know-How haben und damit verschiedene Fehler des Dokuments nur mit sehr unterschiedlichen Wahrscheinlichkeiten finden können, müssen Sie die Effektivitätsschätzung entsprechend korrigieren.

### Capture-Recapture bei mehr als 2 Reviewern:

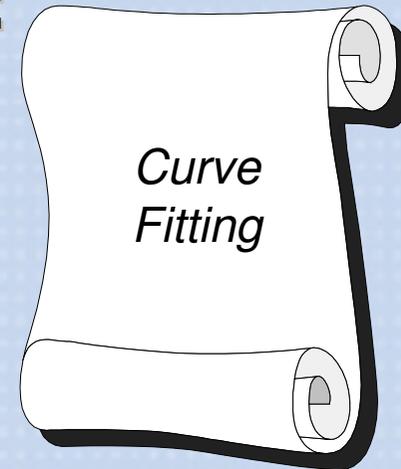
- Reviewer A ist der Reviewer, der am meisten Fehler gefunden hat.
- Die anderen Reviewer stellen "Reviewer B" dar, wobei ein Fehler, der von mehreren gefunden wurde, natürlich nur einfach zählt.

# Effektivitäts-Schätzverfahren

- Stichproben-Reviews

## Effektivitäts-Schätzverfahren:

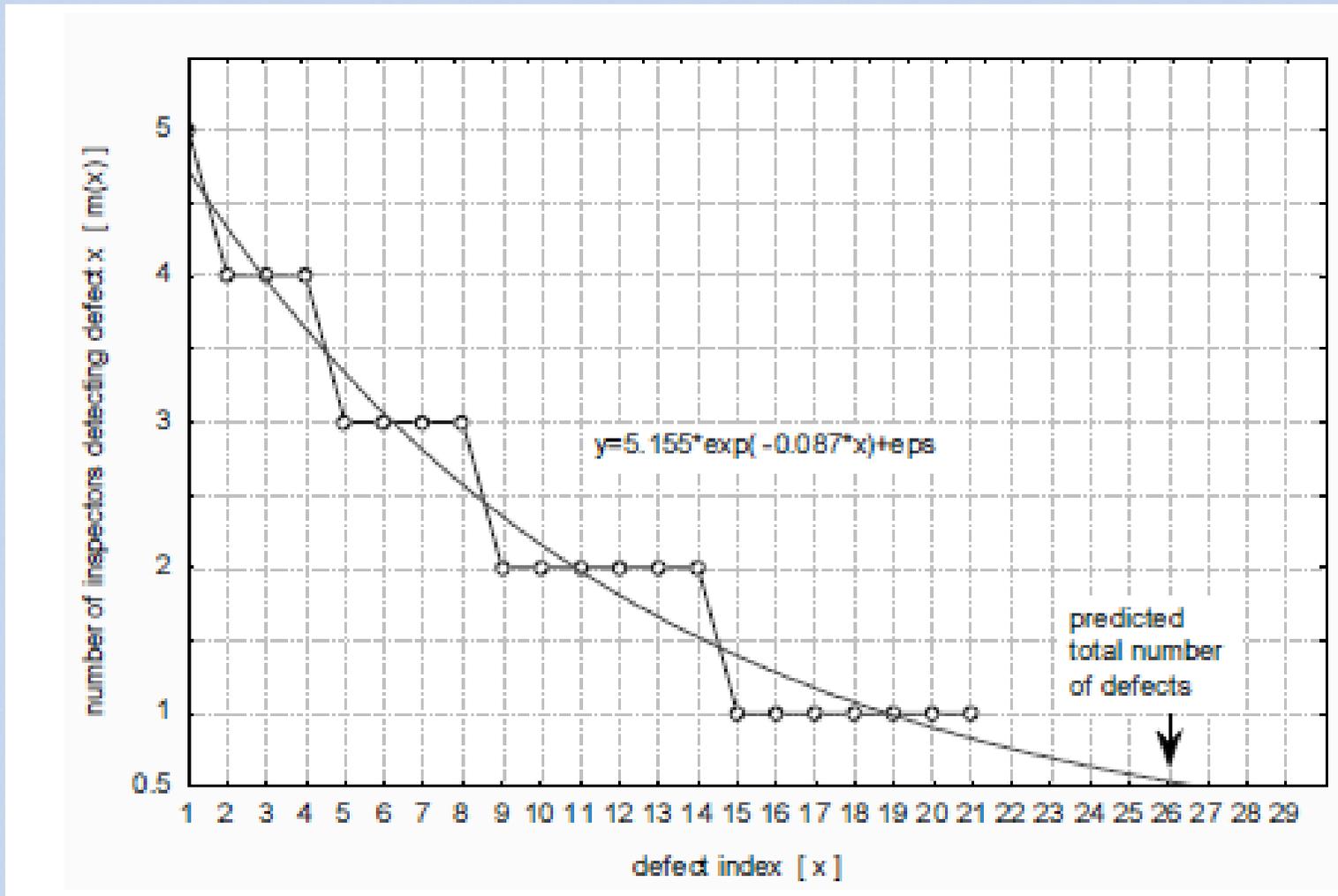
- Vergleich mit optimaler Inspektionsrate
- Capture-Recapture
- **Curve Fitting**
- Subjektiv durch Experten



# Curve Fitting

- Sortierung der Daten nach bestimmten Kriterien
- Anwendung von Regressionstechniken zum Erarbeiten einer Kurve (to fit a curve)
- Hochrechnen auf Basis der ermittelten Kurve
- Bei Reviews am häufigsten: Detection Profile Method (DPM)

# Detection Profile Method (DPM)



Quelle: [Freimut et al 2001], Figure 2: Example of the Detection Profile Method

[Mehr Details siehe Volltext](#)

# Effektivitäts-Schätzverfahren

- Stichproben-Reviews

## **Effektivitäts-Schätzverfahren:**

- Vergleich mit optimaler Inspektionsrate
- Capture-Recapture
- Curve Fitting
- **Subjektiv durch Experten**



## Subjektive Schätzung durch Experten

- Schätzung der gefundenen vs. der Gesamtzahl im Dokument vorhandenen Fehler

[Freimut et al 2001]: Laut mehrerer Studien überraschend genaue Schätzung

- Oder Schätzung von Gesamtzahl der Fehler
  - Minimale Anzahl
  - Wahrscheinliche Anzahl
  - Maximale Anzahl

Aber [Freimut et al 2001]: In 50% aller Reviews liegt die reale Gesamtzahl der Fehler sogar noch über der geschätzten "maximalen Anzahl".

# Tutorial Stichproben-Reviews

- Einführung in Reviews und Inspektionen
- Problematik  
Testaufwandsschätzung,  
Rework-Aufwand und Fehlerdichte
- Schätzung über Stichproben-Reviews
- **Tipps für erfolgreiche Reviews**



Im Original-Tutorial folgten 24 Folien

ROS 06.11.2010



## Kontakt Daten

Name: Peter Rösler  
Firma: ReviewConsult  
Hermann-Gmeiner-Weg 12  
81929 München  
E-Mail: [ros@reviewconsult.de](mailto:ros@reviewconsult.de)  
Tel.: +49 (0) 172/ 8188 154

Name: Maud Schlich  
Firma: ReviewConsult  
E-Mail: [msc@reviewconsult.de](mailto:msc@reviewconsult.de)  
Tel.: +49 (0) 162/ 27 949 20