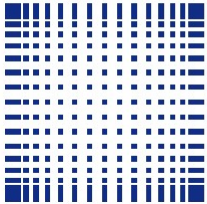




hochschule mannheim



Software- Qualitätsmanagement

SWQ – SS2019 – 2IM

Kapitel: Q-Aufgaben

Dr. Adam Trendowicz
Fraunhofer Institut für Experimentelles Software Engineering



1. Einführung
2. Verbesserungsprozesse
3. Q-Maßnahmen



Antonio Fiol

<http://www.flickr.com/photos/fiol/3455863437>



1. Einführung
2. Verbesserungsprozesse
3. Q-Maßnahmen



Antonio Fiol

<http://www.flickr.com/photos/fiol/3455863437>

Fehleinschätzung...

“640K ought to be enough for anybody.”

Unbekannt, vermutlich nicht Bill Gates

Hat die Software Fehler?

- Drei Meinungen über das gleiche SW-Paket

We've use the Simplex HR software in our Human Resource Department for three years and we have never had a software failure.

I started to use Simplex HR two month ago; we had so many failures that we are considering replacing the software package.

We have been using the same software package for almost four years. We were very satisfied... when we suddenly faced several severe failures...

Quelle: D. Galin: Software Quality Assurance, Pearson, Edinburgh, 2004

Übung & Diskussion

- **Problembeschreibung**
 - Drei Meinungen?
 - Kann das sein?
- **Aufgaben**
 - Falls drei Meinungen möglich sind, geben Sie entsprechende Szenarien an!
 - Falls nein, begründen Sie dies!



- Individuelle Arbeit: **5 min**
- Diskussion: **10 min**



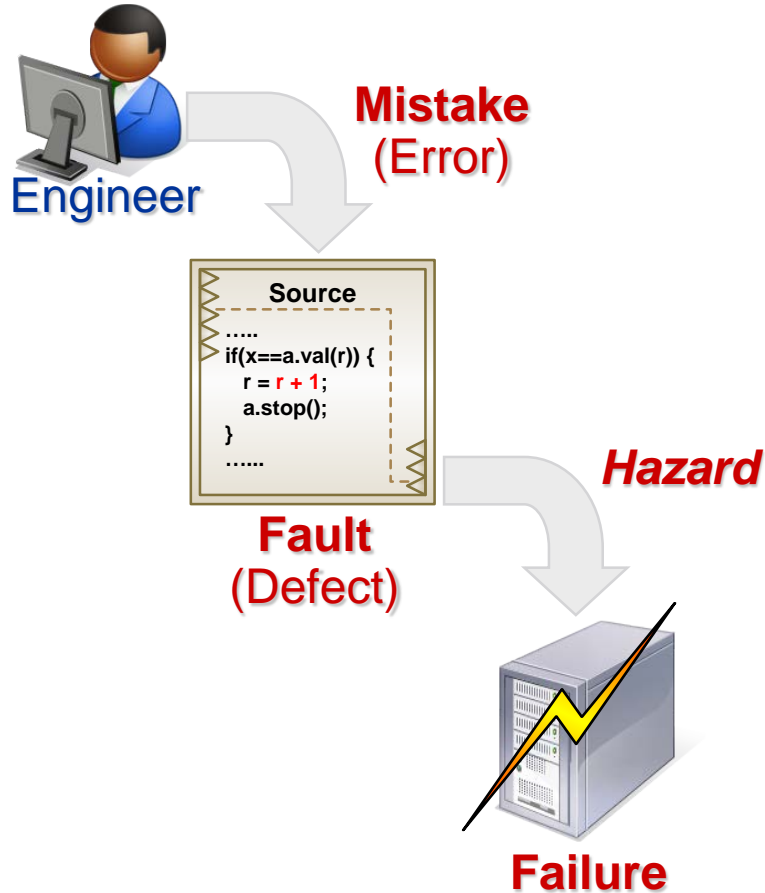
Arto Teräs

<http://ajt.iki.fi/travel/debconf5/page2.html>

Error, Fault, Failure

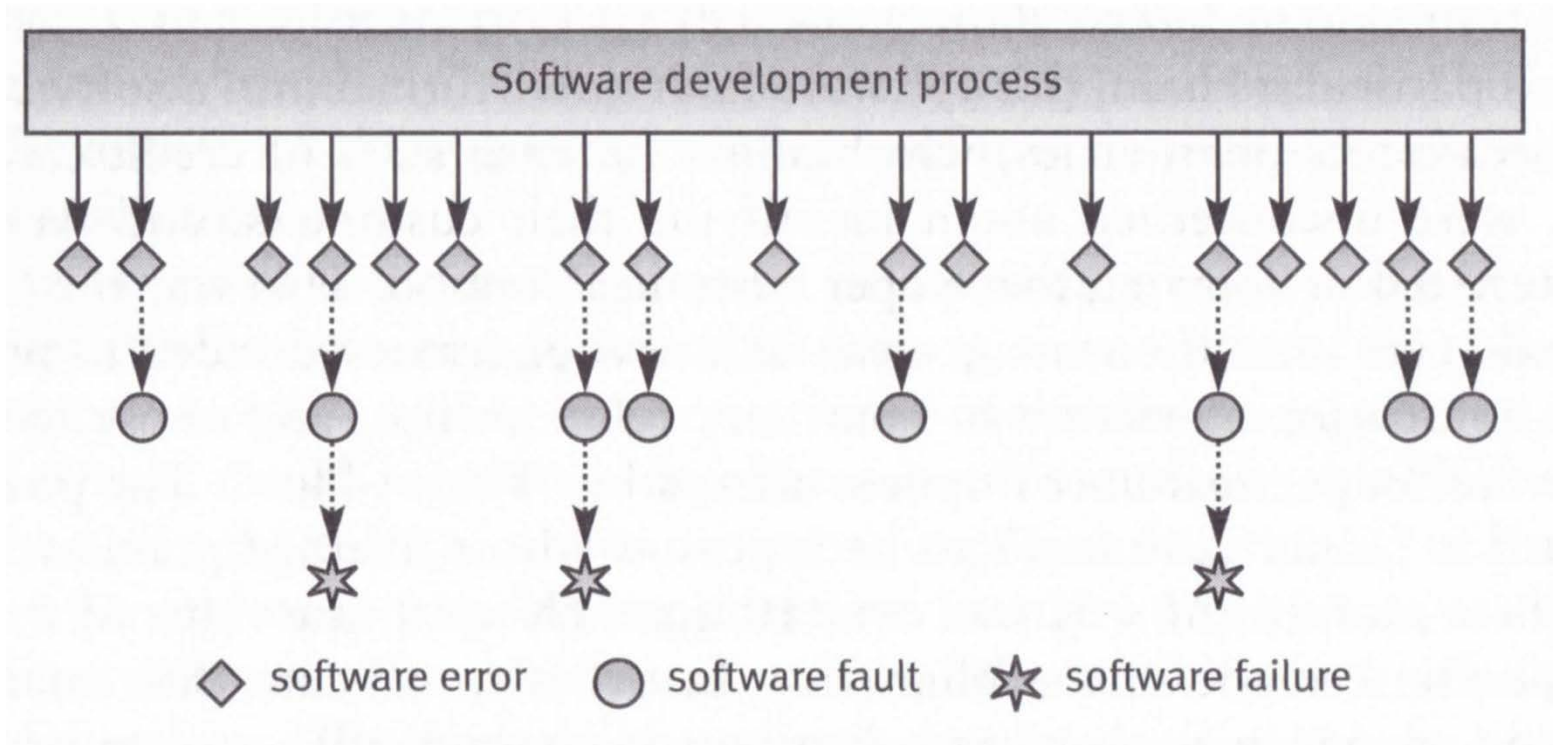
Erläuterung		Terminologie	
Ursache	Human action	Irrtum	Mistake Error
Vermittlung	Manifestation	Fehler	Fault Defect
Wirkung	Result	Fehlverhalten Ausfall	Failure

Error, Fault, Failure



- The source of any **fault (defect)** is an **error** ... usually a human **mistake**.
- A **fault** leads to a **hazardous situation**
- ... but not every **fault** leads to a **failure**
- **Pareto principle**: a very small portion of software defects lead to the majority of system failures

Zusammenhang: Error, Fault, Failure



Quelle: D. Galin, Software Quality Assurance, Pearson, Edinburgh, 2004

Beispiele: Fault, Failure, ...

- **Irrtum** (mistake, error)
 - Falsche Bestimmung des Schleifenindex
- **Fehler** (fault, defect)
 - Die falsche Initialisierung der Schleifenvariable
- **Fehlverhalten** (failure)
 - Ein Feld einer Tabelle in der Ausgabemaske bleibt leer
 - Berechnung einer Geschwindigkeit von 20km/h, wenn die tatsächliche Geschwindigkeit 40km/h beträgt
- **Ausfall** (failure)
 - Programmabsturz wg. Out-of-bounds-Exception

Example (Lufthansa Flight 2904): Fault, Failure



Fault

- logical fault in the braking system

Hazardous situation

- airplane bank angle: landing with aircraft banked to the right to compensate cross wind
- hydroplaning and imprecise (outdated) weather
- Too high landing speed and delayed wheels rotation despite touchdown

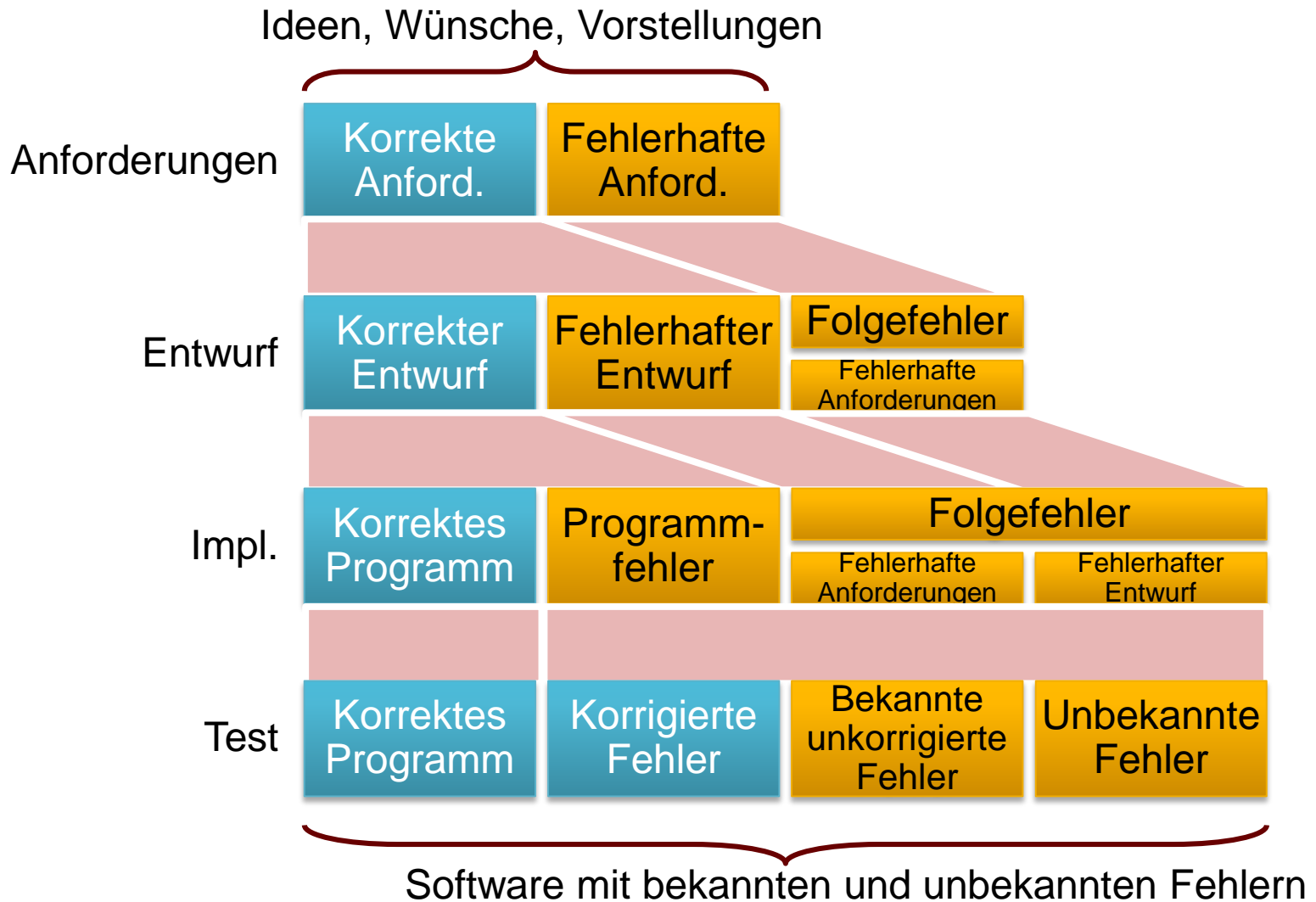
Effect

- 9-second delayed deployment of spoilers and thrust reversers
- 13-second delayed deployment of wheel brakes
- plane overshot runway at 72 knots (133 km/h)

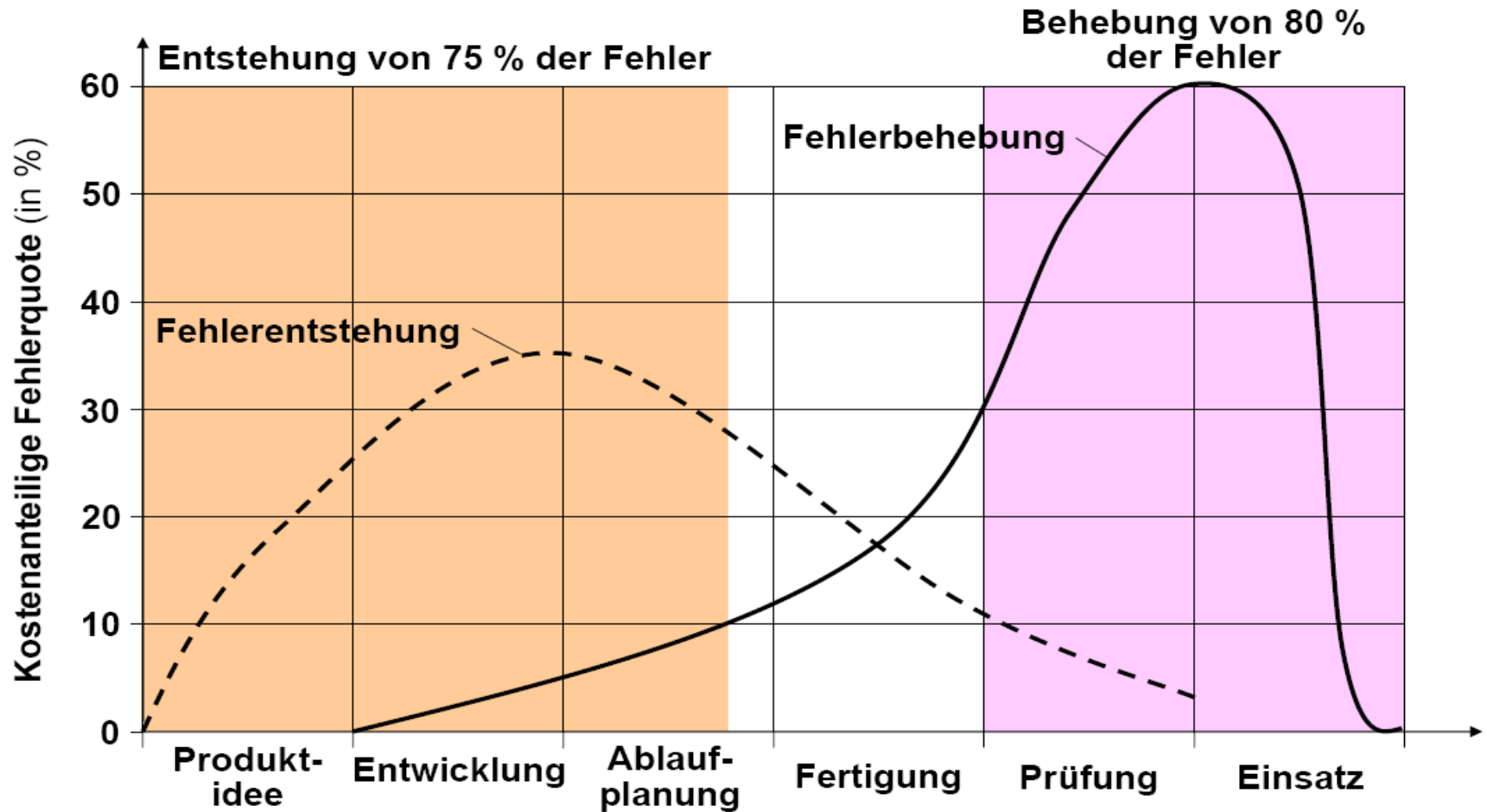
Cost

- Destroyed aircraft
- 2 deaths

Wo entstehen Fehler?



Wann entstehen Fehler



Quelle: Tilo Pfeifer, Qualitätsmanagement, Hanser, 1996

Ursachen für Fehler

- Faulty requirements definition
- Client-developer communication failures
- Deliberate deviations from software requirements
- Logical design errors / procedure errors
- Coding errors → nächste Folie
- Non-compliance with documentation and coding instructions
- Shortcomings of the testing process
- Documentation errors

Quelle: D. Galin: Software Quality Assurance, Pearson, Edinburgh, 2004



Beispiel Arten von Software Fehler

```
for i in range(10) # doppelpunkt fehlt  
    print i
```

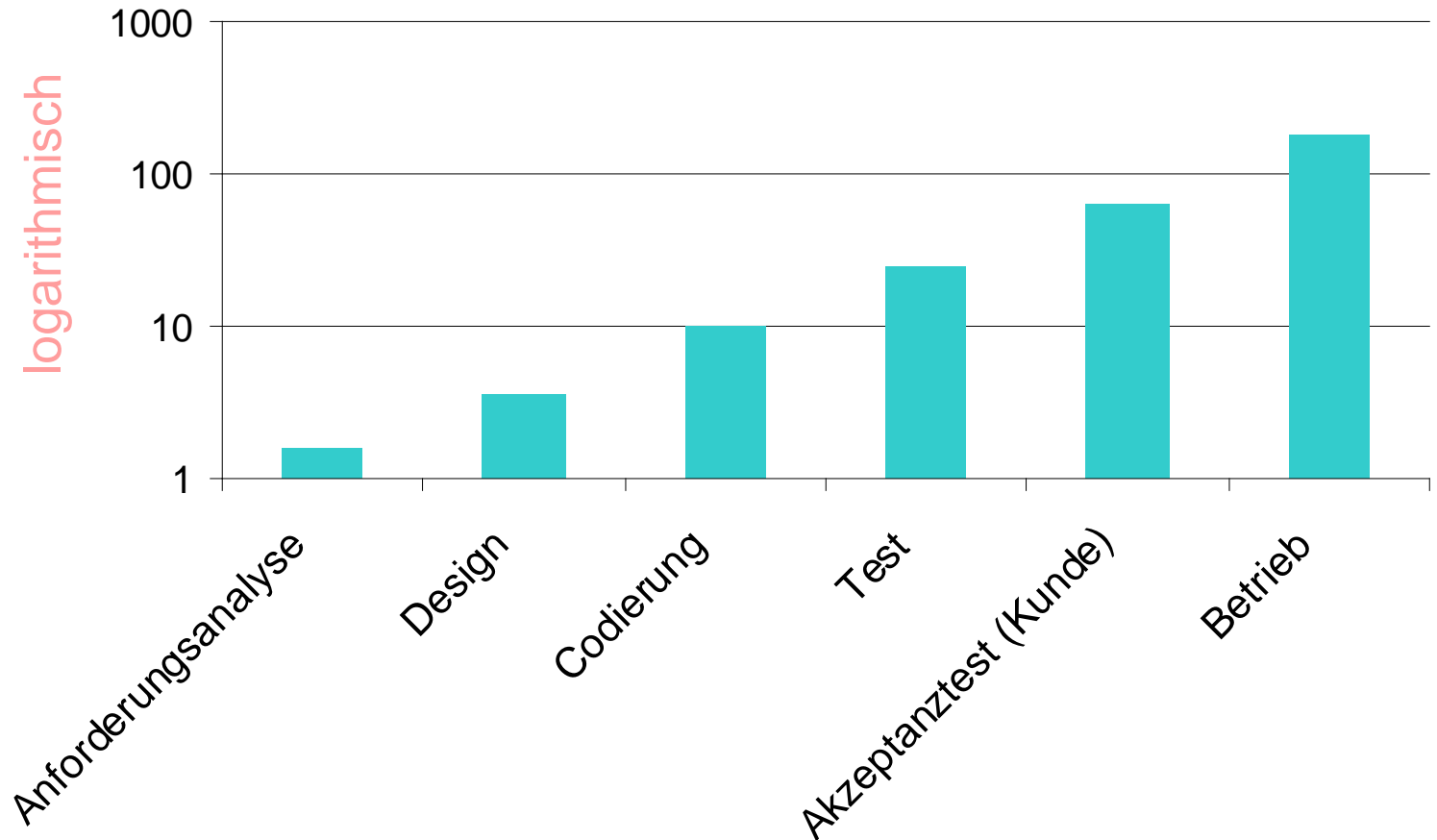
```
name = 'Bergakademie'  
name[12] # Name hat nur Indices 0..11
```

```
# Tausche die Inhalte von 'a' und 'b'  
a:=b; b:=a; # am Ende a==b
```

- **Syntax-Fehler**
 - Verstoß gegen grammatikalische Regel der Programmiersprache
 - Programm wird nicht ausgeführt
- **Laufzeitfehler**
 - z.B. Absturz des Programms bei unerwarteten Eingaben
 - z.B. nicht vorhergesehene Endlosschleifen
- **Logische Fehler**
 - falscher Problemlösungsansatz oder der Algorithmus ist nicht korrekt
- **Designfehler**
 - Fehler bei Anforderungsdefinition
 - Fehler beim Architektur Entwurf
- **Bedienfehler**
 - unübersichtliches Bedienungskonzept

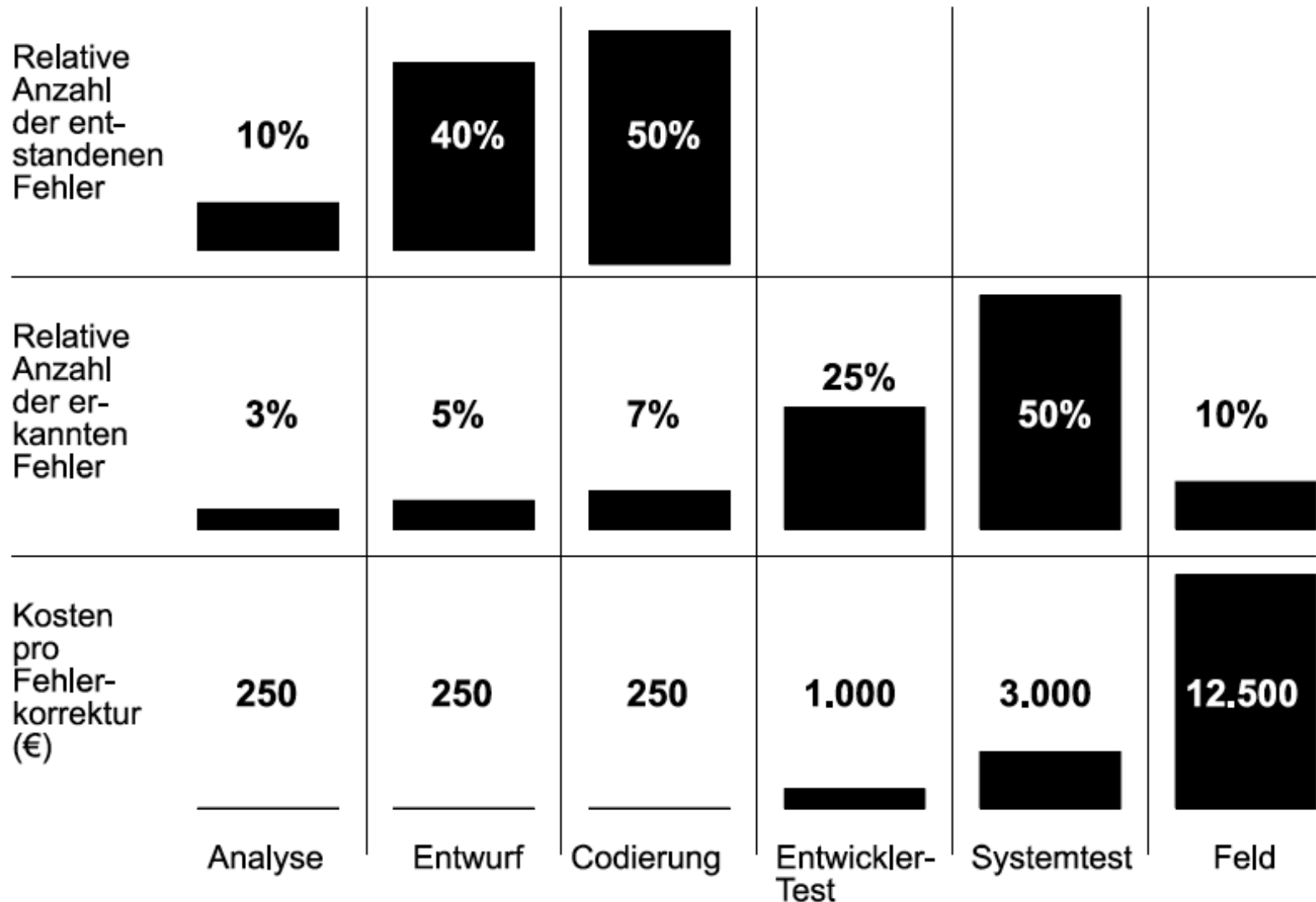
Was kosten Fehler?

Relative Fehlerkosten



Aus: Boehm, Software Engineering Economics, Englewood Cliffs, 1981

Fehleranzahlen & Korrekturkosten



Aus: Liggesmeyer, Software-Qualität, Heidelberg, 2002

Software faults indicate faulty processes

- Failures of software are typically caused by defects in the software products
- Defective software products indicate problems with the processes that are used for their creation
 - The **development processes** introduce (too many) defects into the software at the time of creation
 - The **quality assurance** processes are incapable of removing the (necessary) defects effectively and efficiently
 - ▶ **Effectively**: QA does not detect most critical defects before delivery of the product
 - ▶ **Efficiently**: QA does not detect defects early enough to remove them at relatively low cost
- **Products and processes are closely related!**

Wie werden Fehler vermieden?

- **Verbesserungsprozesse**

- betrifft
 - ▶ Unternehmen (Unternehmensgrundsatz)
 - ▶ Über Projekte hinweg

- **Ansatz**

- ▶ Zyklisches Vorgehen
- ▶ Setzt Rahmen für Q-Sicherung

- **Q-Sicherung**
Q-Maßnahmen

- betrifft
 - ▶ Projekt

- **Ansätze**

- ▶ Prozessorientiert
- ▶ Konstruktiv
- ▶ Analytisch
- ▶ Personell
- ▶ Organisatorisch



1. Einführung
2. Verbesserungsprozesse
3. Q-Maßnahmen

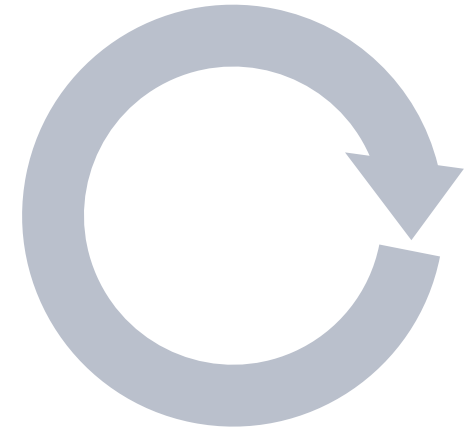


Antonio Fiol

<http://www.flickr.com/photos/fiol/3455863437>

Verbesserungsprozesse

- Grundsätzliche Idee
 - Zyklisches Vorgehen
 - ▶ „es gibt immer etwas besser zu machen“
 - ▶ Kleine Schritte
 - Orientierung an Zielen
 - ▶ Top-down:
Kunde/Markt → Produkt/Projekt
 - Ständige Überprüfung
 - ▶ Keine Beeinträchtigung lfd. Geschäfte
 - ▶ Erfolgskontrolle
 - Ergebnissicherung



Grundlegende Techniken

- Charakterisierung

- Grundlegend
d.h. immer notwendig
- Verschieden formal
d.h. informell/Bauchgefühl \leftrightarrow formal/Prozess
- unterstützen Verbesserungsansatz

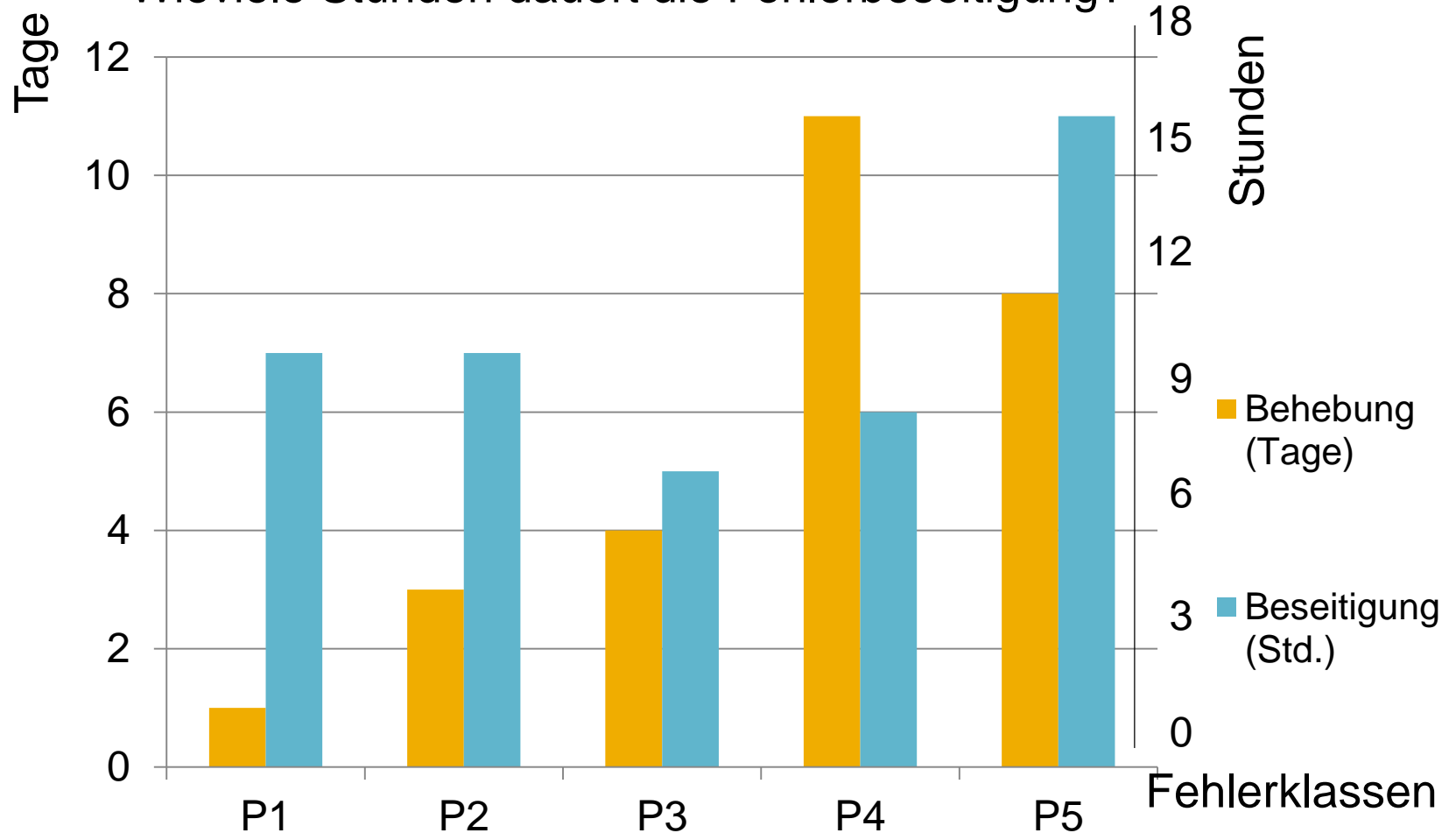
- Techniken

- Prozessmodellierung
- Messen und Bewertung
- Risikomanagement
- Technologietransfer

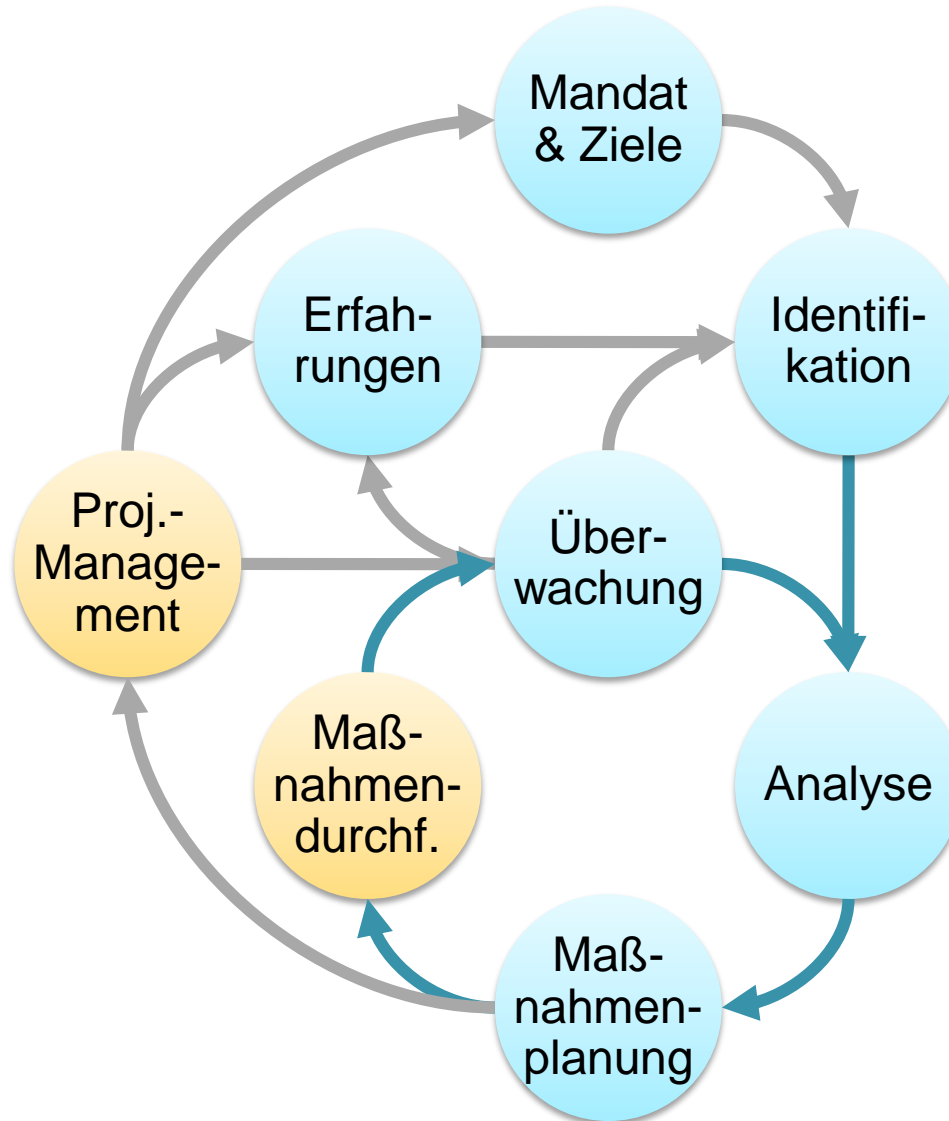
→ SWE

Beispiel: Messen und Bewertung

Wieviele Tage dauert es, bis ein Fehler behoben ist
 Wieviele Stunden dauert die Fehlerbeseitigung?



Beispiel: Risikomanagement



Wie erfolgt Prozessverbesserung?

- in großen Firmen
 - SEPG
 - ▶ Personal, Kosten
 - Expliziter Verbesserungsprozess (z.B. PDCA, QIP, Lean Start-Up)
 - Explizites Assessment (z.B. SPICE, CMMI)
- in kleinen Firmen
 - Keine/wenig Mittel für nicht-Produkte
 - Ad-hoc-Verbesserung
 - Personen statt Prozesse

vereinfachte,
übertriebene
Darstellung

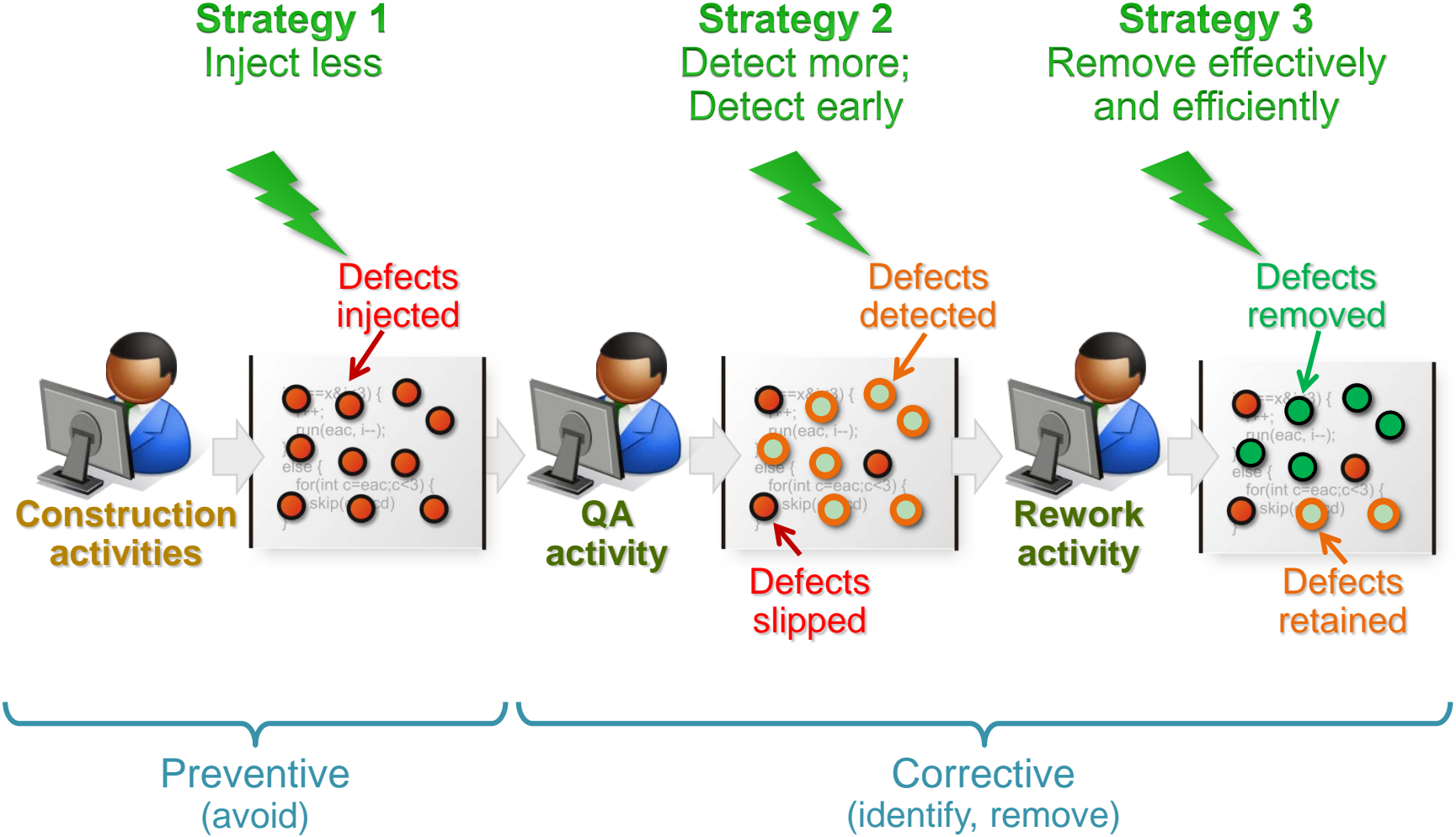
1. Einführung
2. Verbesserungsprozesse
3. Q-Maßnahmen



Antonio Fiol

<http://www.flickr.com/photos/fiol/3455863437>

Basic strategies to cope with faults (resp. quality deficits)



Klassifikation – 2. Variante

- Prozessbezogen
 - ▶ *Vorlesung SWE*
- Konstruktiv
 - ▶ Durch Techniken und Methoden Fehler verhindern
- Analytisch
 - Statisch / Dynamisch
 - ▶ Fehler finden und beheben
- Personell
 - ▶ Durch kompetentes und motiviertes Personal Fehler verhindern und finden
- Organisatorisch
 - ▶ Durch angemessene Organisation Fehler verhindern und finden

- Definition

- Unter konstruktiven Q-Maßnahmen verstehen wir den Einsatz technischer Maßnahmen und Hilfsmittel mit dem Ziel, ein **Produkt mit vorgegebenen Eigenschaften so zu entwickeln** oder zu erweitern, dass so viele Mängel und **Fehler** wie möglich **vermieden** werden

nach Wallmüller

- Wirkung

- präventiv

- Definition

- Unter analytischen Q-Maßnahmen verstehen wir all jene Maßnahmen, die zur **Erkennung und Lokalisierung von Mängeln und Fehlern in einem Produkt** dienen.

nach Wallmüller

- Wirkung

- bewertend

- Definition

- Unter personellen Q-Maßnahmen verstehen wir die **Verbesserung der persönlichen Arbeitsleistung**, wie auch der Teamleistung mit dem Ziel, ein Produkt mit vorgegebenen Eigenschaften so zu entwickeln oder zu erweitern, dass so viele Mängel und **Fehler** wie möglich **vermieden** werden

- Wirkung

- präventiv
(auch bewertend)

- **Definition**

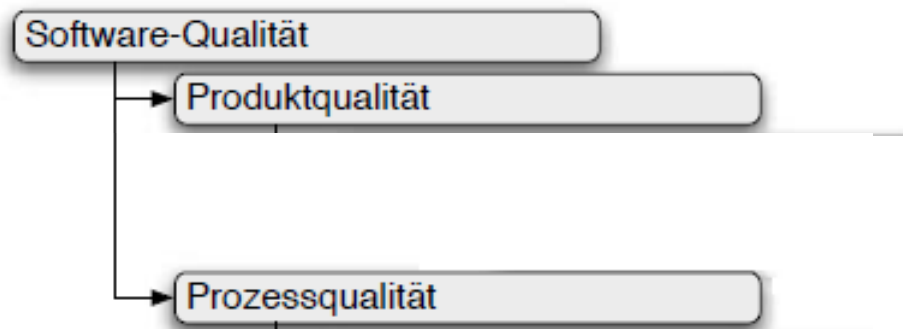
- Unter organisatorischen Q-Maßnahmen verstehen
 - ▶ wir den **Aufbau, die Einführung und die Pflege eines Qualitätsmanagementsystems**, und
 - ▶ die (Re-)Organisation eines Unternehmens mit dem Ziel, **Qualität zu fördern/verbessern**

- **Wirkung**

- präventiv
(auch bewertend)

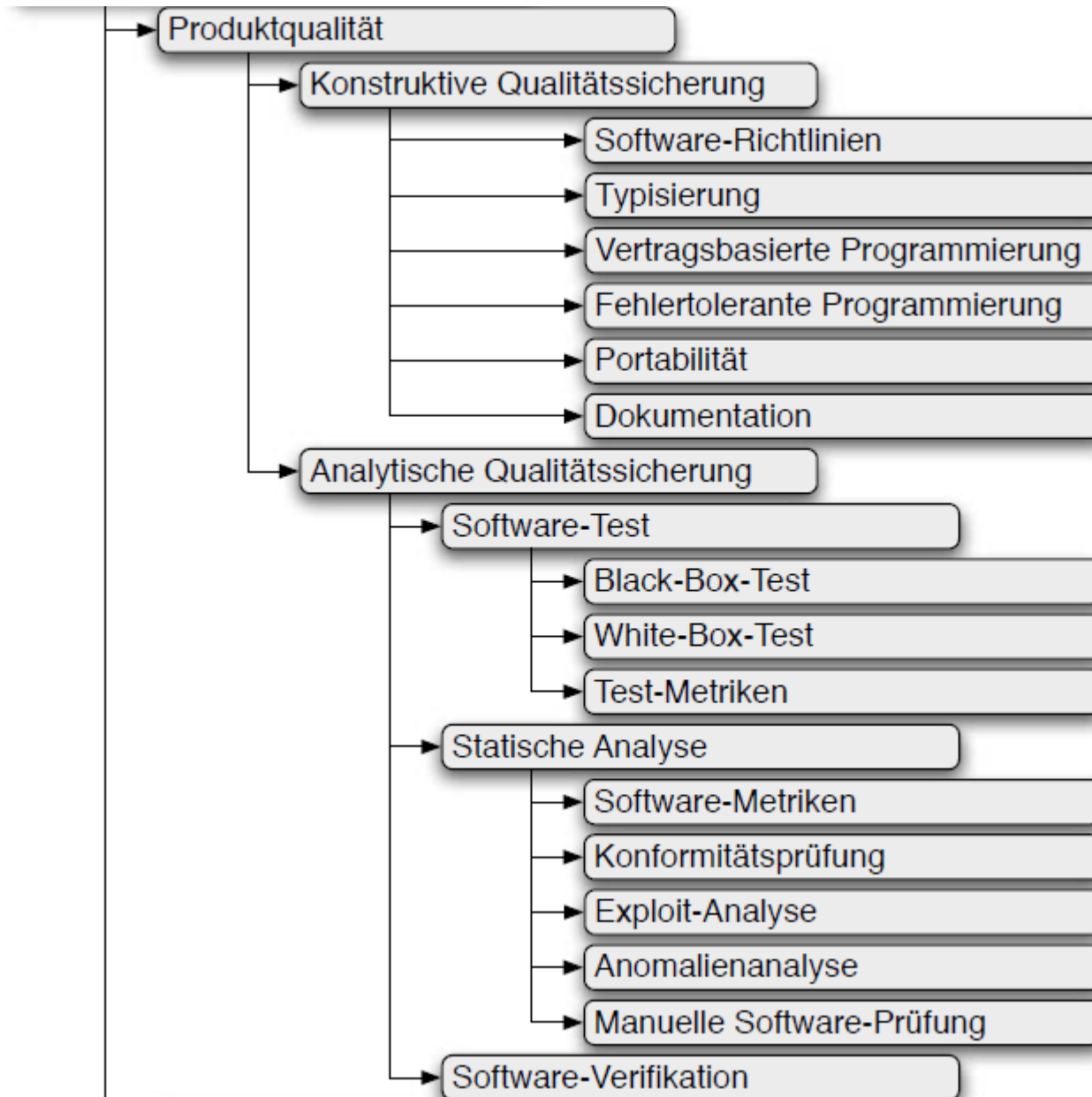


Klassifikation – 3. Variante (1)

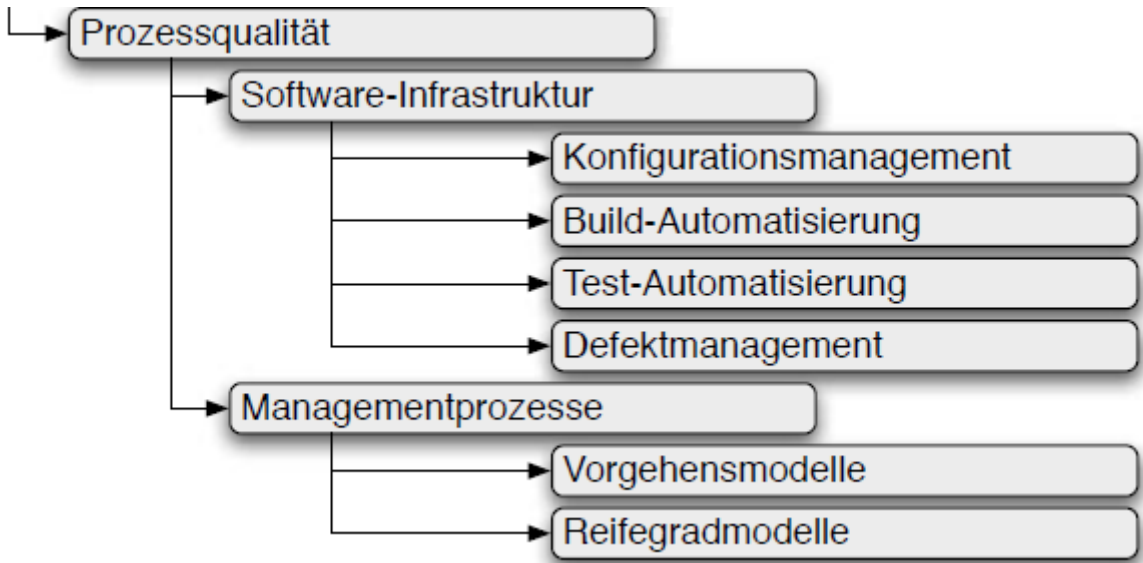


Hoffmann, Software-Qualität, Springer, 2008

Klassifikation – 3. Variante (2)



Klassifikation – 3. Variante (3)



Hoffmann, Software-Qualität, Springer, 2008

Übung & Diskussion

● Aufgabe

- Finden Sie je 2 konkrete Beispiele für folgende Q-Sicherungs-Maßnahmen:
 - ▶ konstruktive
 - ▶ analytische
 - ▶ personelle
 - ▶ organisatorische

- Software Fehler von welchem Typ könnten mit den Maßnahmen (wie?) adressiert werden?
 - ▶ Syntax, Laufzeit, Logik, Design, UI



- Individuelle Arbeit: **15 min**
- Diskussion: **15 min**



Arto Teräs

<http://ajt.iki.fi/travel/debconf5/page2.html>

Beispiel Arten von Software Fehler

```
for i in range(10) # doppelpunkt fehlt  
    print i
```

```
name = 'Bergakademie'  
name[12] # Name hat nur Indices 0..11
```

```
# Tausche die Inhalte von 'a' und 'b'  
a:=b; b:=a; # am Ende a==b
```

- **Syntax-Fehler**
 - Verstoß gegen grammatikalische Regel der Programmiersprache
 - Programm wird nicht ausgeführt
- **Laufzeitfehler**
 - z.B. Absturz des Programms bei unerwarteten Eingaben
 - z.B. nicht vorhergesehene Endlosschleifen
- **Logische Fehler**
 - falscher Problemlösungsansatz oder der Algorithmus ist nicht korrekt
- **Designfehler**
 - Fehler bei Anforderungsdefinition
 - Fehler beim Architektur Entwurf
- **Bedienfehler**
 - unübersichtliches Bedienungskonzept

Beispiele: Konstruktiv

- Auswahl/Vorgabe von Methoden
 - z.B. **use-case-basierte Anforderungsbeschreibung**
- Auswahl/Vorgabe von Formalismen
 - z.B. **UML**
- Auswahl/Vorgabe von Tools
 - z.B. **Rational Rose**

Beispiele: Analytisch

- Inspektion
 - z.B. Walkthrough
- Test
 - z.B. Modultest
- (Code-)Analyse
 - z.B. Anzahl von Codezeilen
 - z.B. zyklomatische Komplexität

Beispiele: Personell

- Weiterbildung
 - z.B. **Schulung**
- Unternehmenskultur
 - z.B. **Teambildung**
 - z.B. **Arbeitsplatzgestaltung**

Beispiele: Organisatorisch

- Planen von Q-Maßnahmen
 - z.B. **Planung von Inspektionen**
- Verteilung von Aufgaben auf Org.-Einheiten
 - z.B. **Trennung Entwicklung-Test**
- Etablieren einer Q-Abteilung
 - z.B. **Stabsstelle in Linie und Projekt**
- Auswahl/Vorgabe von Projekt-Vorgehensmodel
 - z.B. **Agile Entwicklung mit Scrum**

Verantwortlichkeit

Ebene Q-Maßnahmen	Global	Projektbezogen	Phasenbezogen
Prozessorientiert	x	(Anpassung)	
Konstruktiv	(Rahmenvorgaben)	x	x
Analytisch	(Rahmenvorgaben)	(Planung)	x
Personell	x	x	
Organisatorisch	x	x	

x inhaltliche Hauptverantwortung

Was sollten Sie mitnehmen

- **Sie kennen...**
 - Prinzipielle Q-Sicherungsmaßnahmen, mit Beispiele
 - ▶ Konstruktive, analytische, personelle, organisatorische
 - Beispiel Fehlerursachen
 - ▶ Falsche Anforderungen, schlechte Kommunikation, Kode Fehler, etc.
- **Sie können erklären...**
 - Vor-/Nachteile von unterschiedlichen Q-Sicherungsmaßnahmen (bzw. -Strategien)
 - Unterschied zwischen Irrtum (mistake, error), Fehler (fault, defect), Fehlverhalten/Ausfall (failure)
 - Prinzipien von Verbesserungsprozesse, bzw. kontinuierliche Verbesserung (zyklisch, ziel-orientiert, ...)



F R A G E N



photography: woodleywonderworks
<http://www.flickr.com/photos/wwworks/2350106729>
art work: Peter Kaiser