Test und Verlässlichkeit Foliensatz 1: Gefährdungen, Gegenmaßnahmen und Kenngrößen

Prof. G. Kemnitz

Institut für Informatik, TU Clausthal (TV_F1)

May 12, 2021

Organisation

Web-Seite Vorlesung: http://techwww.in.tu-clausthal.de/TestVerl_2020
Kontaktdaten Dozent: gkemnitz@in.tu-clausthal.de

- Foliensätze, Handouts, je Veranstaltung eine zip-Datei mit Audio-Dateien auf der Web-Seite.
- Abarbeitung der Vorlesungen und großen Übungen im Selbststudium:
  - Ausdruck der Handouts für Notizen.
  - Entpacken der zip-Dateien und öffnen des Foliensatzes.
  - Vor Abspielen jeder Audio-Datei zugehörige Folie anzeigen und nach Abspielen Anmerkungen notieren.
  - Für Fragen und Diskussion: Stud-IP-Forum zur Vorlesung.
- HA-Abgabe per Mail an ha-tv@in.tu-clausthal.de als pdf. Übungsblätter und Abgabetermine siehe Web-Seite.
- Ergebnispunkte der HA siehe Tabelle hier auf der Web-Seite.

Prüfung

- Prüfung ab 10 Teilnehmer schriftlich.
- Prüfungszulassung: 50% der erreichbaren Hausübungspunkte. Für mehr erreichte Punkte gibt es bis zu 2 Bonuspunkte für die Prüfungsklausur.
- Erlaubte Hilfsmittel Prüfungsklausur: Eigene Ausarbeitung incl. Handouts mit eigenen Kommentaren und die eigenen Hausübungen, Taschenrechner.
- Erlaubte Hilfsmittel mündlichen Prüfung: ein A4-Blatt (einseitig) mit eigenen Ausarbeitungen.

Alle weiteren Infos sie Web-Seite.

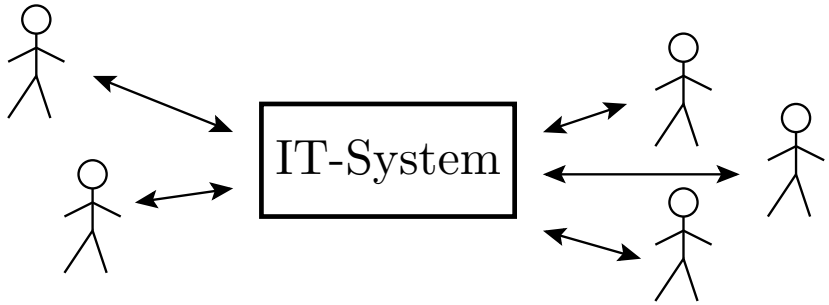
Inhalt Foliensatz TV_F1

Einführung

Vertrauen und Verlässlichkeit

IT-Systeme automatisierten intellektuelle Aufgaben:

- betriebliche Abläufe,
- Steuerung von Prozessen und Maschinen,
- Entwurfsaufgaben, ...



Einsatzvoraussetzung ist Vertrauen, dass

- das System, wenn es gebraucht wird, funktioniert,
- seine Service-Leistungen korrekt und pünktlich ausführt,
- keine unkalkulierbaren Schäden und Kosten verursacht.

Das Vertrauen in eine IT-System setzt Verlässlichkeit des Systems voraus.

Verlässlichkeit

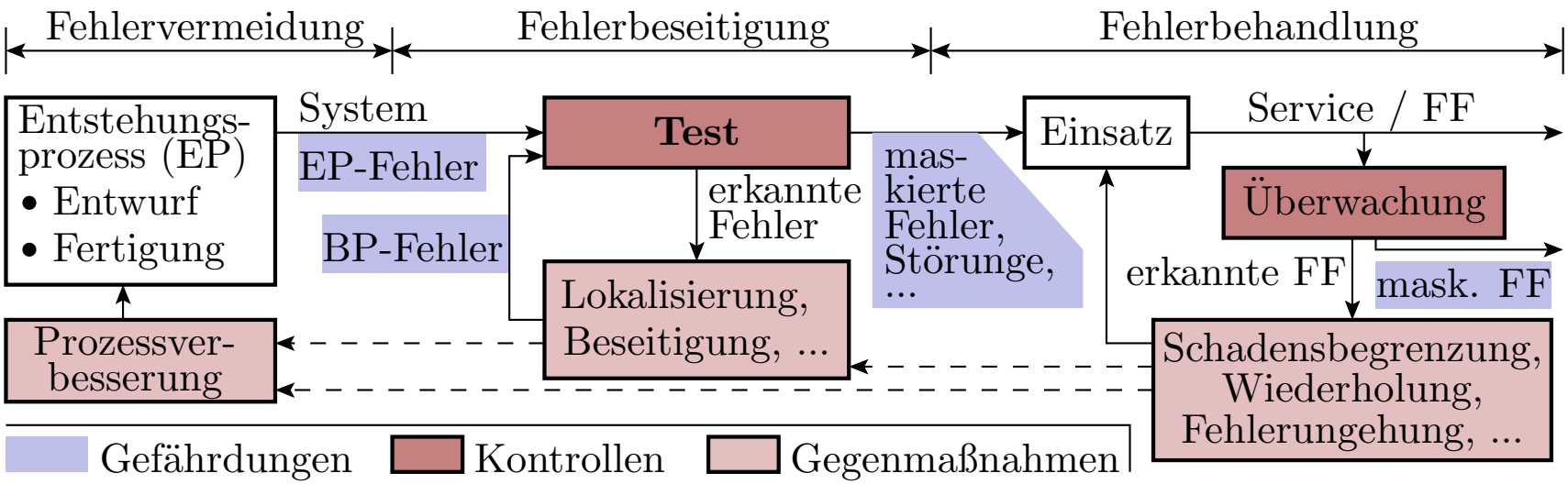
Umgangssprachlich beschreibt Verlässlichkeit (von Personen, Rechnern, ...), dass man ihnen trauen kann. Dabei treffen unterschiedliche Aspekte zusammen (Wünsche, Erwartungen, ...).

Die Verlässlichkeit von IT-Systemen wird durch eine Vielzahl von Aspekten beschrieben. Lapri[1] unterscheidet:

- Gefährdungen (Threats): Fehler, Fehlfunktionen (FF), Störungen, Ausfälle, ...
- Gegenmaßnahmen zur Gefährdungsminderung (Means):
  - Überwachung, bei FF Wiederholung, Fehlerumgehung, ...
  - Test, Fehlerdiagnose und -beseitigung,
  - Fehlervermeidung durch Verbesserung der Entstehungsprozesse.
- Kenngrößen (Attributes) zur Quantifizierung der Gefährdungen und Gegenmaßnahmen.

[1] J.C. Laprie. "Dependable Computing and Fault Tolerance: Concepts and Terminology," 15th IEEE Int. Symp. on Fault-Tolerant Computing, 1985

Die Gefährdungen und Gegenmaßnahmen



1. Fehlervermeidung: Bei Entwurf und Fertigung entstehen Fehler. Ursachenbeseitigung $\Rightarrow$ weniger entstehende Fehler.
2. Fehlerbeseitigung: Beseitigung erkannter Fehler. $\Rightarrow$ weniger Fehler und FF im Einsatz.
3. Fehlerbehandlung: Schadensbegrenzung, aufrechterhalten der Funktion, bei unvorhergesehenen Eingaben und internen FF.

Was kostet Verlässlichkeit?

Zusätzliche Entwurfs- und Betriebskosten für

- Fehlervermeidung, Test, Fehlerbeseitigung,
- Funktionen zur Prozess- und Systemüberwachung,
- Wartungspersonal, ...

Zusätzliche Systemfunktionen für

- prüfgerechten Entwurf, Selbsttests,
- Diagnose, Überwachung, ...

Wenn Verlässlickeit wichtig ist, weit mehr als 50% der Gesamtkosten.

Mit wachsender Systemkomplexität nimmt der Kostenanteil für die Sicherung der Verlässlichkeit an den Gesamtkosten der Systeme zu.

Im Grunde Funktionen zur Sicherung der Verlässlichkeit:

- ESP, ABS, Ölstandüberwachung, Reifendruck,
- Diagnosebus, Fehlerspeicher, ...

Der Preis fehlender Verlässlichkeit

- Datenverlust, Hintertüren für den Datenmissbrauch[2],
- Unfälle, Selbstzerstörung, Produktionsausfälle, ...

Am 3. Juni 1980 meldete ein Rechner des nordamerikanischen Luftverteidigungszentrums den Anflug sowjetischer Nuklearraketen. Sofort wurden Vergeltungsmaßnahmen vorbereitet. Eine Überprüfung der Daten von Radarstationen und Satelliten konnte den Angriff nicht bestätigen, so dass der Start amerikanischer Raketen mit Nuklearsprengköpfen in letzter Minute gestoppt wurde[3].

Urheber der nahen Katastrophe war ein defekter Schaltkreis.

---

[2] https://www.faz.net/aktuell/wirtschaft/diginomics/43-milliarden-euro-schaden-durch-hackerangriffe-15786660.html

[3] Hartmann, J., Analyse und Verbesserung der probabilistischen Testbar- keit kombinatorischer Schaltungen, Diss. Universität des Saarlandes, 1992

Lernziele der Vorlesung

- Überblick über die Gefährdungen, Gegenmaßnahmen und Kenngrößen.
- Themenspezifische Einführung in die Stochastik.
- Funktionen + Techniken zur Überwachung und Fehlertoleranz.
- HW: Fehlermodellierung, prüfgrechter Entwurf, Test, Selbsttest.
- SW: Fehlervermeidung, Test, Testauswahl, Fehlerbehandlung.

Test und Überwachung sind Schlüssel für Verlässlichkeit

Nur auf erkannte Gefährdungen sind schadensmindernde Reaktionen (Fehlervermeidung, Fehlerbeseitigung und Fehlerbehandlung) möglich. Wie im weiteren gezeigt, sind es maßgeblich die Filtergüten der Kontrollen, die die erzielbare Verlässlichkeit begrenzen.

Foliensätze

- F1: Gefährdungen, Gegenmaßnahmen und Kenngrößen.
- F2: Wahrscheinlichkeit, insbesondere für Fehlernachweis und Fehlerbeseitigung.
- F3: Verteilungen, insbesondere für Zählwerte, Schadenskosten, Fehlernachweislängen und Überlebensdauern.
- F4: Abschätzung Fehleranzahl, Fehlfunktionsrate, Schadenskosten, ...; Modellierung Ausfallverhalten.
- F5: Funktionen + Techniken zur Überwachung und Fehlertoleranz.
- F6: HW: Fehlermodellierung, prüfgrechter Entwurf, Test, Selbsttest, ...
- F7: SW: Fehlervermeidung, Test, Testauswahl, Fehlerbehandlung, ...

Verlässlichkeit

Kenngrößen zur Beschreibung der Verlässlichkeit

Zur Bewertung der Verlässlichkeit wird ein betrachtetes System als Service-Leister modelliert, der eine abzählbare Anzahl von Service-Leistungen erbringt. Die SL werden unterteilt in »erbracht, korrekt«, »erbracht, fehlerhaft« (Fehlfunktion, FF) und »nicht erbringbar / Service nicht verfügbar«. Kenngrößen:

- Verfügbarkeit: Zeitanteil, in dem der Service verfügbar ist.
- Zuverlässigkeit: mittlere Anzahl der SL je FF.
- Sicherheit(en): Anzahl der SL je sicherheitsgefährdende FF, ...

Weitere Kenngrößen:

- Fehlerentstehungsrate: Anzahl der entstehenden Fehler je Entstehungs-SL,
- Fehlfunktionsrate: Anzahl der FF je SL.
- Fehlerüberdeckung: Anteil der nachweisbaren Fehler, ...

Service und FF

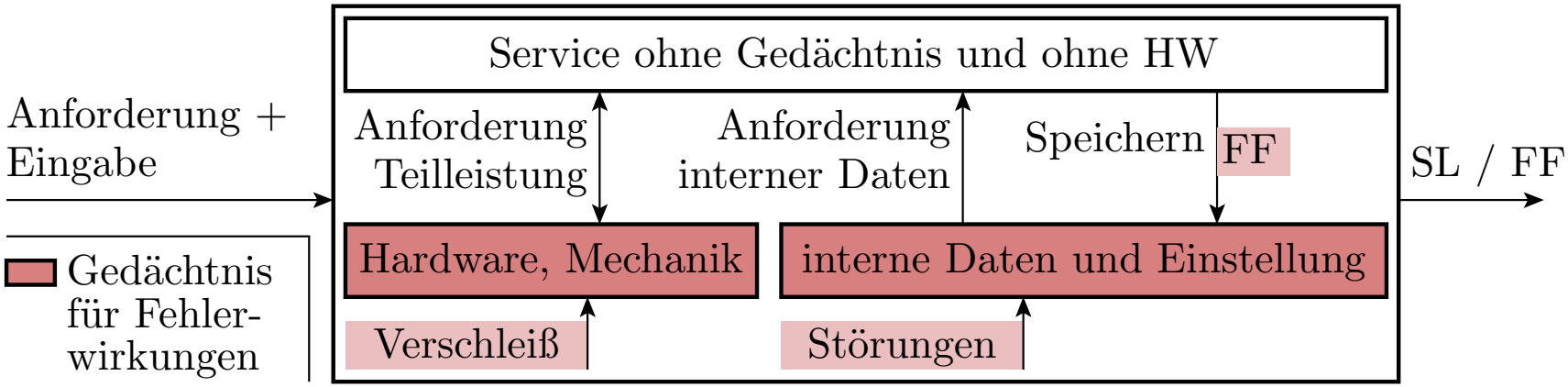
Service-Leistungen und Fehlfunktionen

Ein IT-System, aber auch ein Entwurfs- oder Fertigungsprozess für ein IT-System ist ein Service, der im fehlerfreien Fall auf Anforderung aus Eingaben korrekte Ausgaben erzeugt. Fehler, Störungen und Ausfälle können bewirken:

1. Einzel-FF: Einzel-SL mit falschem oder ohne Ergebnis,
2. FF-Burst: Folge / Häufung fehlerhafter SL bis zur Reparatur und/oder Neuinitialisierung,
3. Versagen: keine Ergebnisse / Service nicht verfügbar bis zur Reparatur und/oder Neuinitialisierung.

Der Beginn einer FF-Burst oder eines Versagens zählt im weiteren als eine FF. Nach erster FF bis Wiederherstellung gilt das System als nicht verfügbar.

Ursachen für Burst-FF und Versagen



Systeme mit Gedächtnis für Fehlerwirkungen:

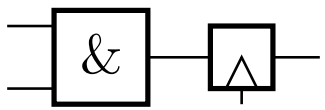
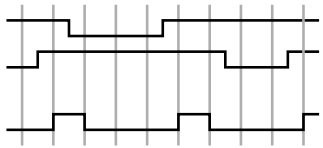
- Hardware und Mechanik, in der durch Verschleiß in der Einsatzphase neue Fehler entstehen können.
- Verfälschung service-interner Daten und Einstellungen durch FF.

Beseitigung der Fehlerwirkung:

- Reparatur: Austausch ausgefallener Hardware-Komponenten,
- Neuinitialisierung: Wiederherstellung eines korrekten Zustands für die service-interner Daten und Einstellungen.

Service-Leister

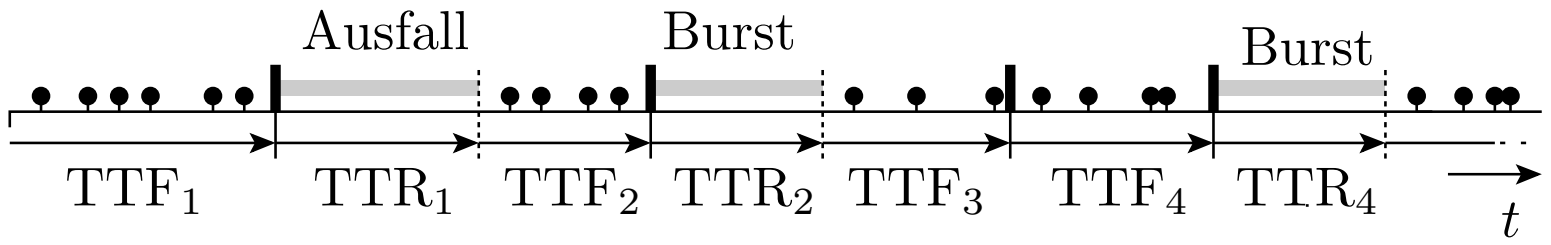
Das Service-Modell ist auf informationsverarbeitende Systeme, technische Steuerungen, Fertigungsabläufe, Entwurfsprozesse, ... anwendbar.

getaktete Digitalschaltung		E:  A:
Programm mit EVA-Struktur	`uint8_t up(uint8_t a){ return 23 * a; }`	E: 10 101 ... A: 320 19 ...
Server	E: z.B. eine Datenbankanfrage A: Ergebnisdatensatz	
Fertigungsprozess	E: Fertigungsauftrag, Material, ... A: gefertigtes Produkt	
Entwurfsprozess	E: Entwurfsauftrag A: Entwurf	

Verfügbarkeit

$MTTF$, $MTTR$, Verfügbarkeit und PFD

Verfügbarkeit ist der Anteil der Nutzungsdauer, in dem das System funktioniert (kein Ausfall, keine FF-Burst):



TTF_i Zeit bis zur nächsten FF (Time to Fail)

TTR_i Zeit bis Reparatur / Neuinitialisierung (Time to Repair)

- Mittlere Zeit bis zum Versagen (Mean Time to Fail):

$$\hat{MTTF} = \frac{1}{\#FF} \cdot \sum_{i=1}^{\#FF} TTF_i$$

- Mittlere Reparaturzeit bis Austausch ausgefallene Hardware und/oder Neuinitialisierung (Mean Time to Repair):

$$\hat{MTTR} = \frac{1}{\#FF} \cdot \sum_{i=1}^{\#FF} TTR_i$$

(.$\hat{}$. – Schätzwert; #FF – Anzahl Nichtverfügbarkeitsintervalle).

- Verfügbarkeit:

$$V = \frac{MTTF}{MTTF + MTTR} \tag{1}$$

$$\hat{V} = \frac{\sum_{i=1}^{\#FF} TTF_i}{\sum_{i=1}^{\#FF} TTF_i + \sum_{i=1}^{\#FF} TTR_i}$$

- PFD (Probability of Failure on Demand): Wahrscheinlichkeit, dass das System zu einem zufälligen Anforderungszeitpunkt nicht verfügbar ist:

$$PFD = \frac{MTTR}{MTTF + MTTR} = 1 - V$$

Reparaturzeiten für hochverfügbare Systeme

V	PFD	$\sum_{i=1}^{\#FF} TTR_i$	
		pro Monat	pro Jahr
99%	1%	7,2 h	87,6 h
99,9%	0,1%	43 min	8,8 h
99,99%	0,01%	4,3 min	53 min

99% ist normal. Hohe Verfügbarkeiten ab 99,9% verlangen spezielle Maßnahmen:

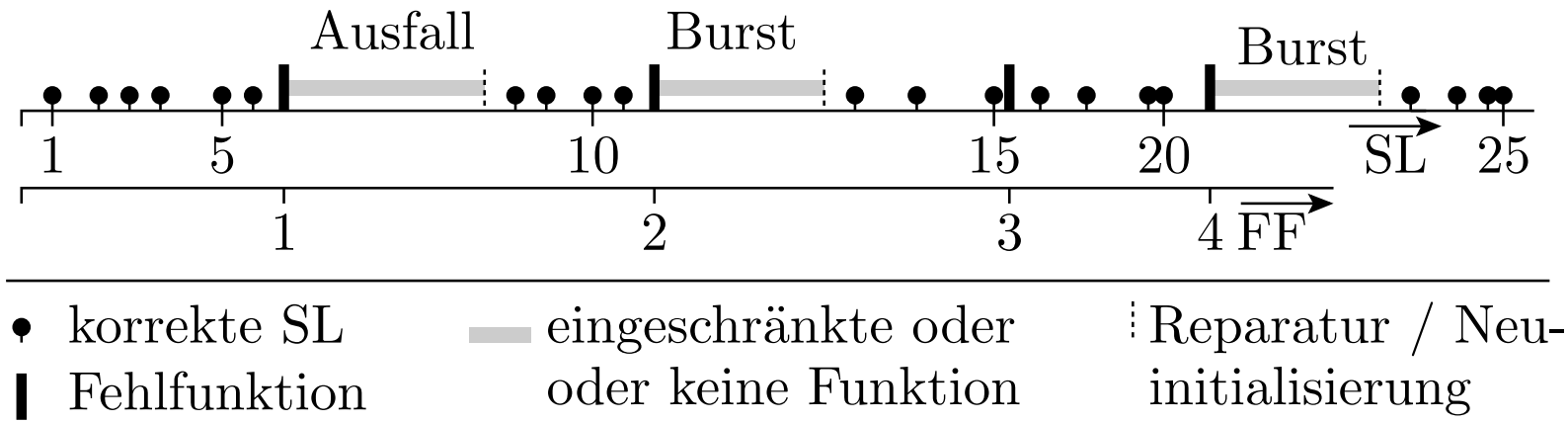
- unterbrechungsfreie Stromversorgung,
- Raid-Speicher,
- gespiegelte Server, vorbeugende Wartung, ...

(siehe später Foliensatz 4, Abschn. Ausfälle und Foliensatz 5, Abschn. Fehlertoleranz).

Zuverlässigkeit

Zuverlässigkeit

Zuverlässigkeit sei im weiteren der Anteil der vom funktionierenden System ausgeführten SL je FF:



$$\hat{Z} = \frac{\#SL}{\#FF} \tag{2}$$

($\#SL$ – Anzahl der Service-Leistungen; $\#FF$ – Anzahl der Fehlfunktionen). Service-Anforderungen und FF, während das System nicht oder nur eingeschränkt funktioniert, werden nicht mit gezählt.

Fehlfunktionsrate und $MTTF$

Die Fehlfunktionsrate ist der Kehrwert der Zuverlässigkeit:

$$\zeta = \frac{1}{Z}; \quad \hat{\zeta} = \frac{\#FF}{\#SL} \tag{3}$$

Beschreibung als Verhältnis zwischen $MTTF$ und MTS:

$$Z = MTTF/MTS \tag{4}$$

$$\zeta = MTS/MTTF \tag{5}$$

($MTTF$ – Mean Time to failture, mittlere Zeit bis zur Fehlfunktionen; MTS – Mean Time to Service, mittlere Service-Dauer).

Beispiel 1

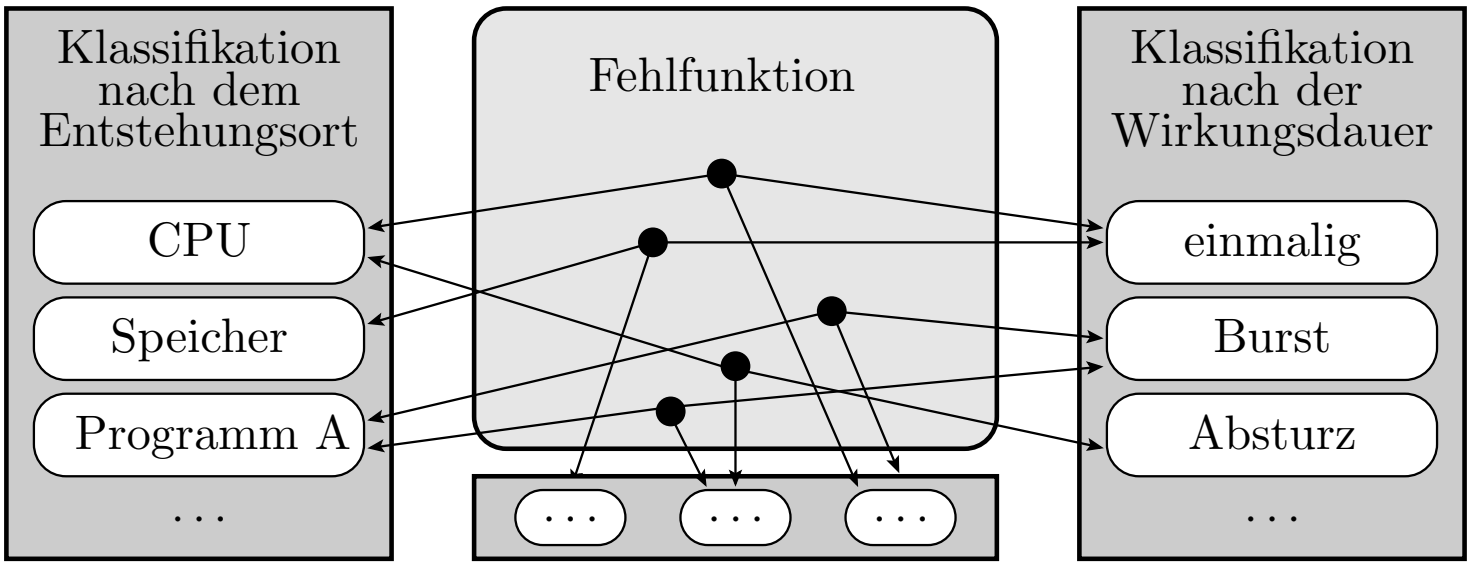
Innerhalb von 30 h Programmnutzung 3 FF, $MTS = 0{,}1\,\mathrm{h/SL}$:

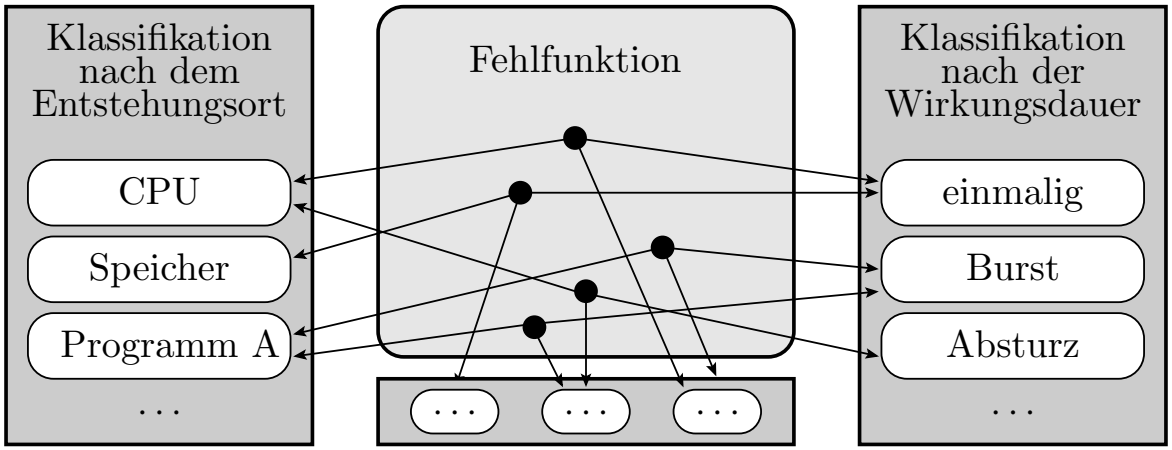
$$MTTF = 10\,\mathrm{h/FF}; \quad Z = 100\,\mathrm{SL/FF}; \quad \zeta = Z^{-1} = 10^{-2}\,\mathrm{FF/SL}$$

Teilzuverlässigkeiten

Die Fehlfunktionen (FF) eines Systems lassen sich nach Ort, Ursache, Schaden, ... unterschiedlichen Klassen zuordnen:

- nur FF eines bestimmten Teilsystems,
- nur durch HW, nur durch SW verursachte FFs,
- nur FF, die für die Betriebs- / Daten- / Zugangssicherheit relevant sind, ...:





Bei einer eindeutigen Zuordnung jeder Fehlfunktion zu genau einer Klasse i ist die Gesamtanzahl der Fehlfunktionen $\#FF$ die Summe der Anzahl der Fehlfunktionen $\#FF_i$ aller Klassen i:

$$\#FF = \sum_{i=1}^{\#FFK} \#FF_i$$

($\#FFK$ – Anzahl der Fehlfunktionsklassen). Die Fehlfunktionsrate ist die Summe der Fehlfunktionsraten aller Fehlfunktionsklassen:

$$\frac{\#FF}{\#SL} = \sum_{i=1}^{\#FFK} \frac{\#FF_i}{\#SL}; \quad \zeta = \sum_{i=1}^{\#FFK} \zeta_i$$

Der Kehrwert der Gesamtzuverlässigkeit ist die Summe der Kehrwerte der Teilzuverlässigkeiten:

$$\frac{1}{Z} = \sum_{i=1}^{\#FFK} \frac{1}{Z_i}$$

Beispiel 2

Die Fehlfunktionen seien entweder vom Speicher, vom Prozessor, von der Software oder vom Rest verursacht. Es liegen folgende $MTTF$-Werte für Teilsysteme vor:

Teilsystem i	Speicher	Prozessor	Software	Rest
$MTTF_i$	500 $^{\mathrm{h}}/_{\mathrm{FF}}$	3.000 $^{\mathrm{h}}/_{\mathrm{FF}}$	1000 $^{\mathrm{h}}/_{\mathrm{FF}}$	2.000 $^{\mathrm{h}}/_{\mathrm{FF}}$

Mittlere Service-Dauer $MTS = 1\ ^{\mathrm{min}}/_{\mathrm{SL}}$.

1. Wie groß sind die vier aus den $MTTF$-Werten ableitbaren FF-Raten ζ_i und Teilzuverlässigkeiten Z_i?
2. Wie groß ist sind die FF-Rate ζ und die Zuverlässigkeit Z des Gesamtsystems?

Lösung

1. FF-Raten und Teilzuverlässigkeiten ($MTS = 1\,^{\mathrm{min}}/_{\mathrm{SL}}$):

Teilsystem	Speicher	Prozessor	Software	Rest
$MTTF_i$	$500\,^{\mathrm{h}}/_{\mathrm{FF}}$	$3.000\,^{\mathrm{h}}/_{\mathrm{FF}}$	$1000\,^{\mathrm{h}}/_{\mathrm{FF}}$	$2.000\,^{\mathrm{h}}/_{\mathrm{FF}}$
ζ_i	$3{,}33 \cdot 10^{-5}\ \frac{\mathrm{FF}}{\mathrm{SL}}$	$5{,}56 \cdot 10^{-6}\ \frac{\mathrm{FF}}{\mathrm{SL}}$	$1{,}67 \cdot 10^{-5}\ \frac{\mathrm{FF}}{\mathrm{SL}}$	$8{,}33 \cdot 10^{-6}\ \frac{\mathrm{FF}}{\mathrm{SL}}$
Z_i	$3 \cdot 10^{4}\ \frac{\mathrm{SL}}{\mathrm{FF}}$	$1{,}8 \cdot 10^{5}\ \frac{\mathrm{SL}}{\mathrm{FF}}$	$6 \cdot 10^{4}\ \frac{\mathrm{SL}}{\mathrm{FF}}$	$1{,}2 \cdot 10^{5}\ \frac{\mathrm{SL}}{\mathrm{FF}}$

($\frac{\mathrm{SL}}{\mathrm{FF}}$ – Service-Leistungen je Fehlfunktion).

2. FF-Rate und Zuverlässigkeit des Gesamtsystems:

$$\begin{aligned}
\zeta &= \frac{1}{3 \cdot 10^{4}\ \frac{\mathrm{SL}}{\mathrm{FF}}} + \frac{1}{1{,}8 \cdot 10^{5}\ \frac{\mathrm{SL}}{\mathrm{FF}}} + \frac{1}{6 \cdot 10^{4}\ \frac{\mathrm{SL}}{\mathrm{FF}}} + \frac{1}{1{,}2 \cdot 10^{5}\ \frac{\mathrm{SL}}{\mathrm{FF}}} \\
&= 6{,}39 \cdot 10^{-5}\ \frac{\mathrm{FF}}{\mathrm{SL}} \\
Z &= \frac{1}{\zeta} = 1{,}57 \cdot 10^{4}\ \frac{\mathrm{SL}}{\mathrm{FF}}
\end{aligned}$$

Sicherheit

Schaden durch Fehlfunktionen

Der potentielle Schaden durch Fehlfunktionen reicht von unerheblich bis sehr groß. Für Industriegeräte werden nach IEC 61508 folgende Sicherheitsstufen (SIL – **S**afety **I**ntegrity **L**evel) unterschieden:

- SIL1, AK 2 & 3: Kleine Schäden an Anlagen und Eigentum.
- SIL2, AK 4: Große Schäden an Anlagen, Personenverletzung.
- SIL3, AK 5 & 6: Verletzung von Personen, einige Tote.
- SIL4, AK 7: Katastrophen, viele Tote und gravierende Umweltverschmutzung.

Mit der Sicherheitsstufe sind u.a. Obergrenzen der $PFH = \zeta \cdot \frac{MTS}{1\,\mathrm{h}}$ (Wahrscheinlichkeit einer FF in einer Stunde) und der PFD (Probability of Failure on Demand, Wahrscheinlichkeit der Nichtverfügbarkeit) und Mindestanforderungen an die Fehlerbehandlung verbunden:

SIL	1	2	3	4
$PFH_{\max}$	10^{-5}	10^{-6}	10^{-7}	10^{-8}
$PFD_{\max}$	10^{-1}	10^{-2}	10^{-3}	10^{-4}

Sicherheit

Sicherheiten sind Teilzuverlässigkeiten, bei denen nur die FF ausgewählter Gefährdungen mitgezählt werden, abschätzbar durch zählen der gefährdenden FF (GFF) einer FF-Stichprobe:

$$\hat{S} = \frac{\#SL}{\#GFF} \tag{6}$$

Rate der gefährdenden FF als Kehrwert der Sicherheit:

$$\zeta_S = \frac{1}{S}$$

($\#GFF$ – Anzahl der gefährdenden FF).

Art der Sicherheit	zu zählende Gefährdungen
Betriebssicherheit (safty)	Personen- und Umweltschäden
Datensicherheit (security)	Datendiebstahl
Sicherheit Datenerhalt	Datenverlust
...	...

Sicherheit und Zuverlässigkeit

Die Rate der gefährdenden FF verhält sich proportional zur Anzahl aller FF:

$$\zeta_{\mathrm{S}} = \eta_{\mathrm{g}} \cdot \zeta$$

η_{g} – Anteil der FF, die als gefährdend zählen:

$$\hat{\eta}_{\mathrm{g}} = \frac{\#GFF}{\#FF} \tag{7}$$

Die Sicherheit S und die mittleren Zeit bis zu nächsten sicherheitsgefährdenden FF $MTTF_{\mathrm{S}}$ erhöht sich um den Kehrwert von η_{g}:

$$S = \frac{Z}{\eta_{\mathrm{g}}}; \quad MTTF_{\mathrm{S}} = \frac{MTTF}{\eta_{\mathrm{g}}}$$

($MTTF$ – mittleren Zeit bis zu nächsten FF).

Die Sicherheit eines System lässt sich erhöhen durch

- Erhöhung der Zuverlässigkeit und/oder
- Verringerung des Anteils der gefährdenden FF durch Fehlerbehandlung.

Beispiel: Sicherheit durch Zusatzsteuergerät

Eine Fahrzeug habe eine $MTTF = 1000\,\mathrm{h}$ bis zu einer Fehlfunktionen. Der Anteil der betriebssicherheitsgefährdenden FF sei $\eta_\mathrm{G} = 1\%$ und die mittlere Service-Dauer (mittlere Fahrtdauer) betrage $MTS = 1\,\mathrm{h}$.

1. Wie hoch sind Zuverlässigkeit Z des Systems, wie hoch ist die mittlere Zeit bis zu einer gefährdenden FF ($MTTF_\mathrm{S}$) und wie hoch ist die Sicherheit S?
2. Ein zusätzliches elektronisches Steuergerät verringert den Anteil der gefährdenden FF auf $\eta_\mathrm{G.SG} = 10^{-3}$ GFF je FF, hat aber selbst nur eine begrenzte Zuverlässigkeit Z_SG. Wie hoch muss die Zuverlässigkeit des Steuergeräts Z_SG mindestens sein, damit sich die Sicherheit des Gesamtsystems mit Steuergerät S_MSG mindestens verfünffacht?

Lösung Aufgabenteil 1

Zuverlässigkeit nach Gl. 4:

$$Z = \frac{MTTF}{MTS} = \frac{10^3\mathrm{h}}{1\mathrm{h}} \cdot \frac{\mathrm{SL}}{\mathrm{FF}} = 10^3 \frac{\mathrm{SL}}{\mathrm{FF}}$$

$MTTF_\mathrm{S}$ bis zu einer für die Betriebssicherheit gefährlichen FFs:

$$MTTF_\mathrm{S} = \frac{MTTF}{\eta_\mathrm{G}} = \frac{1000}{1\%}\mathrm{h} = 10^5\mathrm{h}$$

Betriebssicherheit:

$$S = \frac{MTTF_\mathrm{S}}{MTS} \approx \frac{10^5\mathrm{h}}{1\mathrm{h}} = 10^5 \frac{\mathrm{SL}}{\mathrm{GFF}}$$

Lösung Aufgabenteil 2

Ein zusätzliches elektronisches Steuergerät verringert den Anteil sicherheitsgefährdenden FF auf $\eta_{\mathrm{G.SG}} = 0{,}1 \cdot \eta_{\mathrm{G}}$.. Welche Zuverlässigkeit Z_{SG} muss das Steuergerät mindesten haben, damit sich die Gesamtsicherheit verfünffacht:

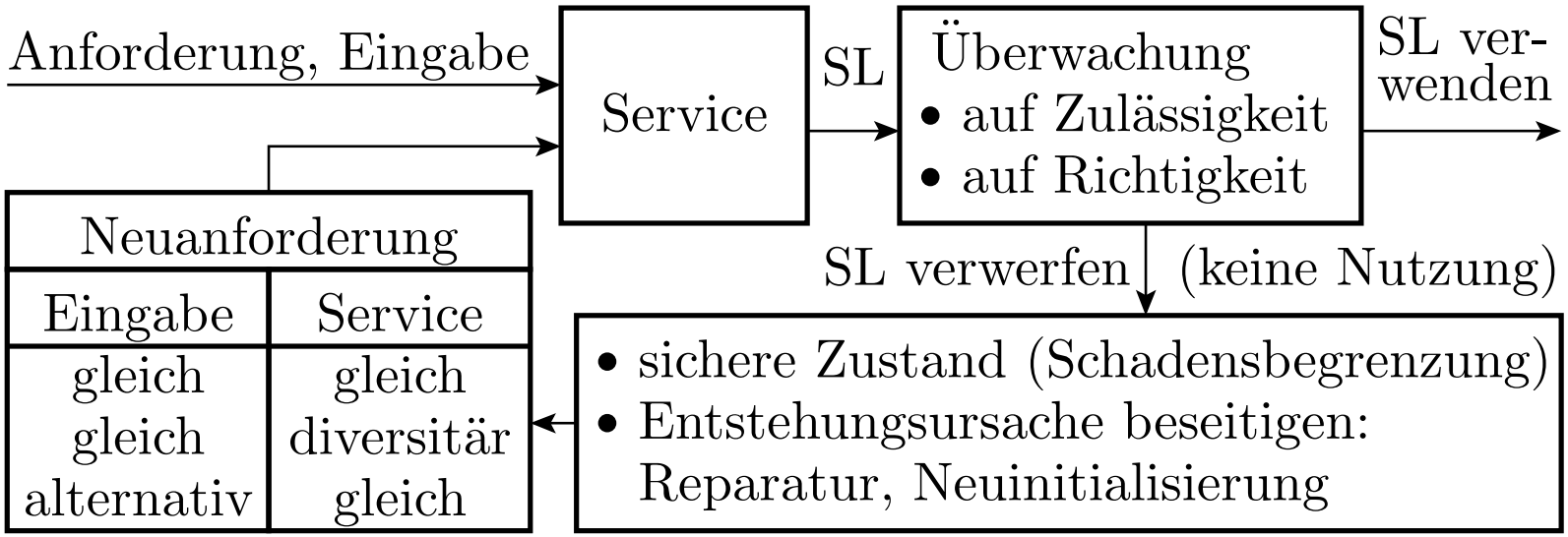
$$S_{\mathrm{MSG}} \geq 5 \cdot S$$

$$\begin{aligned} \frac{5 \cdot Z}{\eta_{\mathrm{G}}} &\leq S_{\mathrm{MSG}} = \frac{Z_{\mathrm{MSG}}}{\eta_{\mathrm{G.SG}}} = \frac{10}{\eta_{\mathrm{G}}} \cdot \frac{1}{\frac{1}{Z} + \frac{1}{Z_{\mathrm{SG}}}} \\ \frac{Z}{2} &\leq \frac{1}{\frac{1}{Z} + \frac{1}{Z_{\mathrm{SG}}}}; \quad Z_{\mathrm{SG}} \geq Z = 10^3 \, \frac{\mathrm{SL}}{\mathrm{FF}} \end{aligned}$$

Das zusätzliche Steuergerät muss mindestens genauso so zuverlässig wie das Fahrzeug sein.

Fehlerbehandlung

Fehlerbehandlung



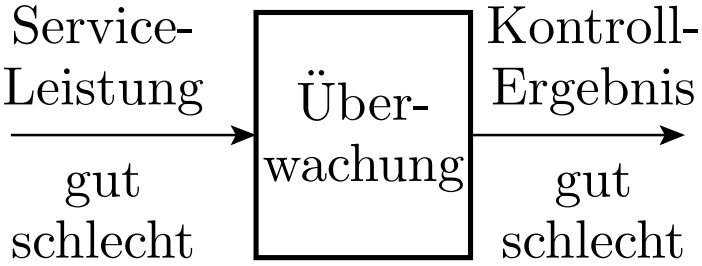
Zur Vermeidung von unvorhersehbarem Systemverhalten bei FF:

- Überwachung Eingaben, innerer Zustände und SL auf Zulässigkeit oder Richtigkeit,
- fehlerhafte SL verwerfen,
- definierten (sicheren) Zustand herstellen (Robustheit).
- Korrektur (Fehlertoleranz). Entstehungsursache der FF vorher beseitigen oder umgehen.

Kenngrößen

Kenngrößen der Überwachung

Eine Überwachung ist ein Service mit gut/schlecht-Ergebnis:



Service-Ergebnis	Kontroll-ergebnis	Fehlfunktion der Kontrolle
schlecht	schlecht	—
gut	gut	—
schlecht	gut	Maskierung
gut	schlecht	Phantom-FF

Mögliche FF der Service-Leistung »Überwachung«:

1. Maskierung, Nichterkennen von FF. Kenngröße FF-Überdeckung:

$$F\hat{F}C = \frac{\#EFF}{\#FF} \tag{8}$$

   ($\#EFF$ – Anzahl der erkannten FF, $\#FF$ – Anzahl aller FF).

2. Phantom-FF. Vermeindliches Erkennen nicht vorhandener FF. ...

Mögliche FF der Service-Leistung »Überwachung«:

1. Maskierung, Nichterkennen von FF. ...
2. Phantom-FF. Vermeindliches Erkennen nicht vorhandener FF. Kenngröße Phantom-FF-Rate:

$$\hat{\zeta}_{\mathrm{Phan}} = \frac{\#PFF}{\#SL} \tag{8}$$

($\#PFF$ – Anzahl der Phantom-FF, $\#SL$ – Anzahl der SL.

Beispiel 3

System: FF-Rate ohne Überwachung $\zeta_{\mathrm{O\ddot{U}}} = 1\,\%\,^{\mathrm{FF}}/_{\mathrm{SL}}$,
Kenngrößen der Überwachung: $FCC = 80\%$, $\zeta_{\mathrm{Phan}} = 2\,\%\,^{\mathrm{PFF}}/_{\mathrm{SL}}$.

- FF-Rate nach Korrektur der FF, ohne dass Phantom-FF bei der Korrektur zu richtigen FF werden:

$$\zeta_{\mathrm{M\ddot{U}.\,min}} = \zeta_{\mathrm{O\ddot{U}}} \cdot (1 - FFC) = 0{,}2\,\%\,^{\mathrm{FF}}/_{\mathrm{SL}}$$

- ... mit Phantom-FF-Umwandlung in richtige FF:

$$\zeta_{\mathrm{M\ddot{U}.\,max}} = \zeta_{\mathrm{M\ddot{U}}} + \zeta_{\mathrm{Phan}} = 2{,}2\,\%\,^{(\mathrm{P})\mathrm{FF}}/_{\mathrm{SL}}$$

Robustheit und Fehlertoleranz

Robustheit: Vermeidung unvorhersehbares Systemverhalten. Kenngröße Anteil der FF, auf die das System robust reagiert. Schätzwert:

$$\hat{ROB} = \frac{\#FFR}{\#FF} \tag{9}$$

($\#FFR$ – Anzahl der internen FF ohne unvorhersehbares Systemverhalten).

Fehlertoleranz (von lateinisch tolerare »erleiden«, »erdulden«): Aufrechterhalten der Funktion bei unvorhergesehenen Eingaben oder oder internen FF. Kenngröße Anteil der FF, die das System selbst korrigiert. Schätzwert:

$$\hat{FT} = \frac{\#FFT}{\#FF} \tag{10}$$

($\#FFT$ – Anzahl der internen FF, bei denen die Funktion aufrecht erhalten bleibt).

Fehlerbehandlung verbessert die Sicherheit (SL je gefährdende FF) ... und die Zuverlässigkeit (SL je FF) ...

Fehlerbehandlung verbessert die Sicherheit (SL je gefährdende FF) um den Kehrwert der Gegenwahrscheinlichkeit der Robustheit

$$S_{\mathrm{FB}} = \frac{S}{1 - ROB}$$

und die Zuverlässigkeit (SL je FF) um den Kehrwert der Gegenwahrscheinlichkeit der Fehlertoleranz:

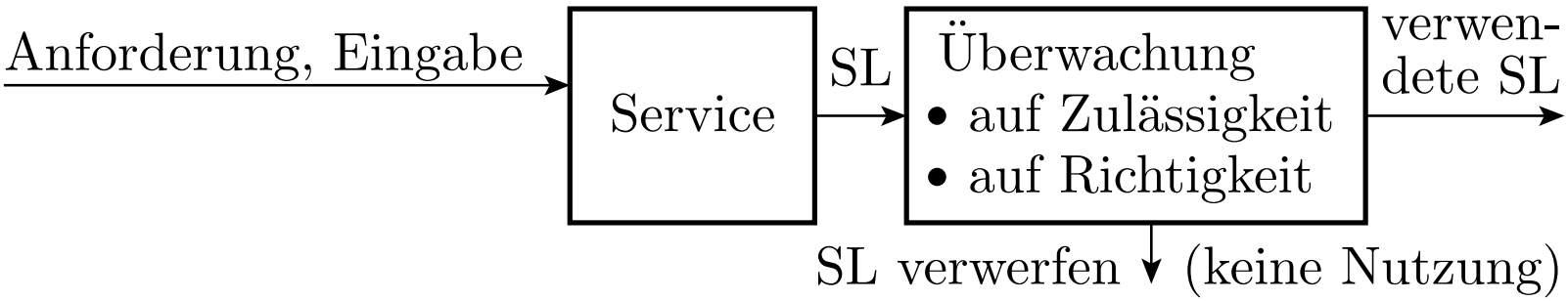
$$Z_{\mathrm{FB}} = \frac{Z}{1 - FT}$$

Fehlertoleranz setzt Robustheit und Robustheit Nachweisbarkeit der FF voraus:

$$FT \leq ROB \leq FFC$$

Überwachungsverfahren

Überwachungsverfahren und ihre Güte



Service-Eingaben und Service-Leistungen bestehen aus:

- Format: Konstante, immer erfüllte Merkmale (Zeitschranken, Wertebereiche, ...).
- Daten: Variable Merkmale (Werte von Datenobjekten, ...).

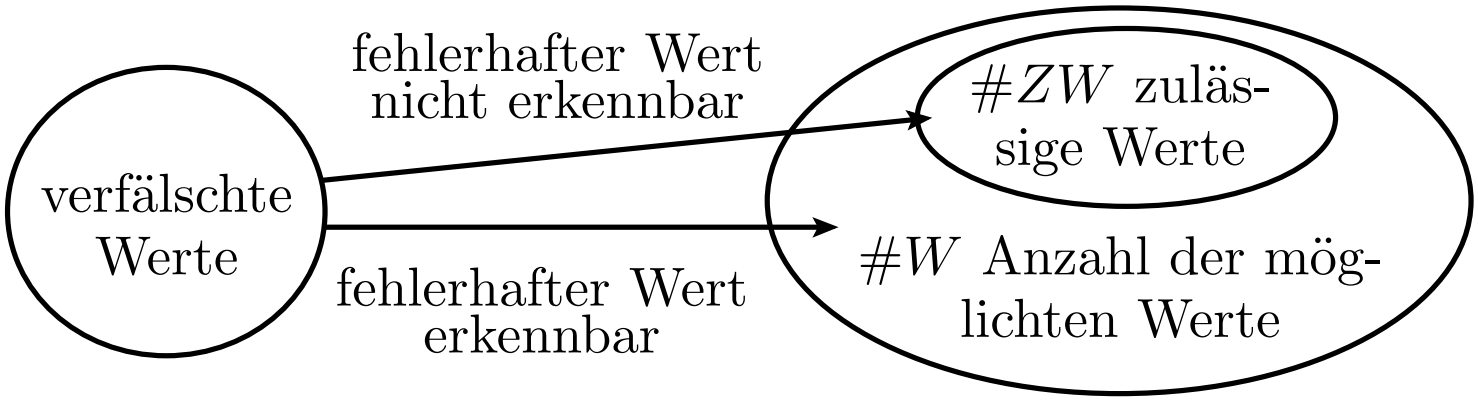
Überwachungsarten:

1. Formatkontrollen: Überwachung der SL auf Zulässigkeit,
2. Datenkontrollen: Überwachung der SL auf Richtigkeit.

Richtige SL sind auch zulässig, zulässige SL können, aber müssen nicht richtig sein. Erstaunlicherweise lassen sich mit Formatkontrollen bei vergleichbarem Aufwand höhere FF-Überdeckungen erzielen.

Kontrolle auf Zulässigkeit und Datenredundanz

Fehlererkennende Codes, Prüfkennzeichen, Wertebereichskontrolle, ...



Fehlfunktionsüberdeckung ist der Anteil der auf unzulässige Werte abgebildeten fehlerhaften Werte. Wenn Verfälschungen gleich häufig auf alle möglichen Werte abgebildet werden

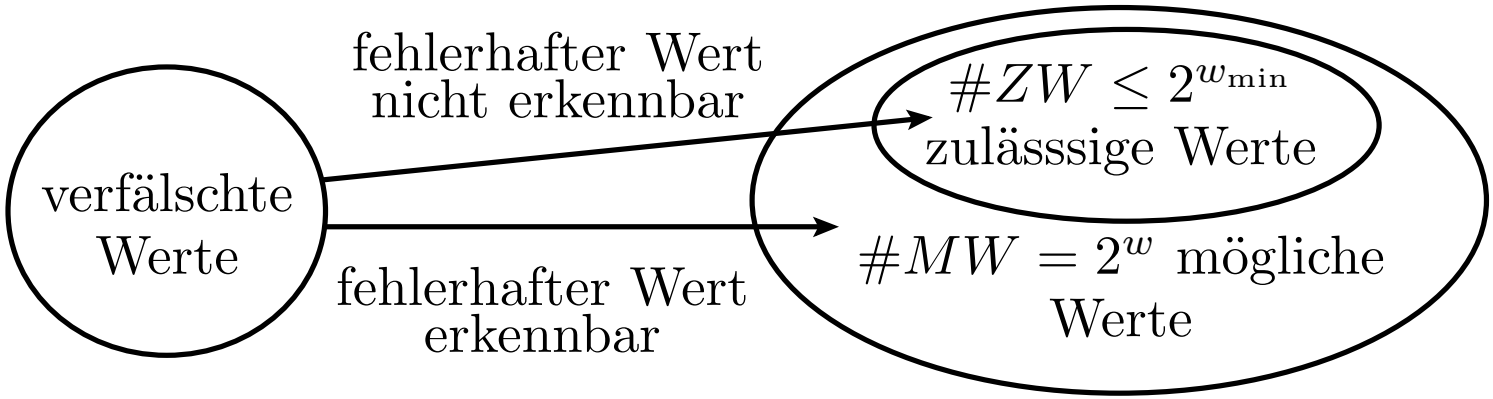
$$FFC \approx 1 - \frac{\#ZW}{\#W}$$

($\#W$ – Anzahl der überprüften Werte; $\#ZW$ – Anzahl der davon zulässigen Werte). Phantom-FF entstehen bei diesem Überwachungsprinzip nicht.

Redundante Bits

Es genügen $w_{\min}$ Bits für die Unterscheidung aller zulässigen Werte.
Bei Darstellung mit r zusätzlichen (redundanten) Bits:

$$w = r + w_{\min}$$



$$1 - FFC \approx \frac{\#ZW}{\#W} < \frac{2^{w_{\min}}}{2^{r+w_{\min}}} = 2^{-r}$$

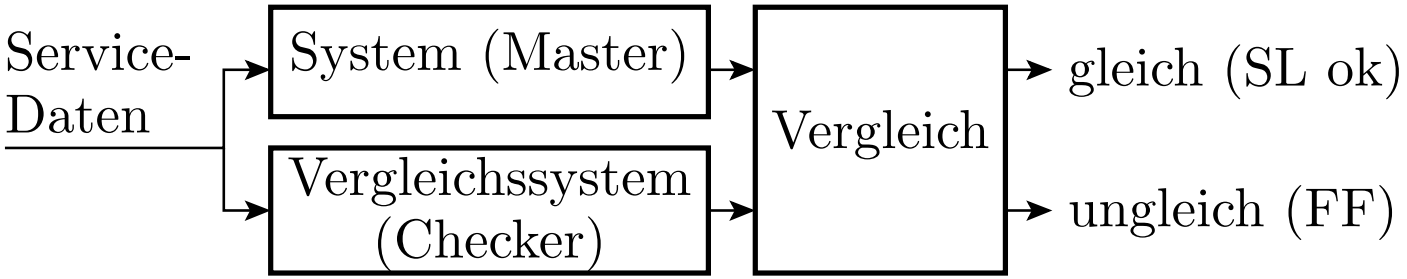
$$FFC \gtrapprox 1 - 2^{-r}$$

r	10	20	30
FFC	$\approx$ 99,9%	$\approx 1 - 10^{-6}$	$\approx 1 - 10^{-9}$

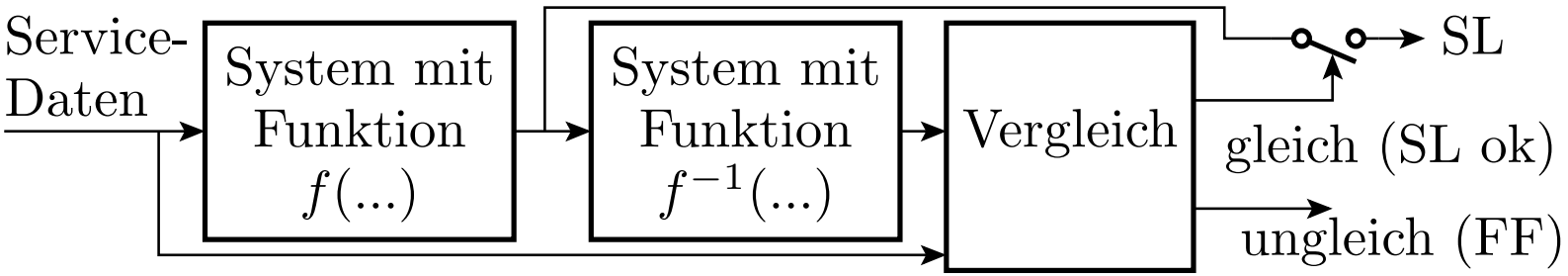
Bei angenommen $w_{\min} = 10^3$ kein nennenswerte Zusatzaufwand.

Verfahren zur Kontrolle auf Richtigkeit

1. Verdopplung und Vergleich:

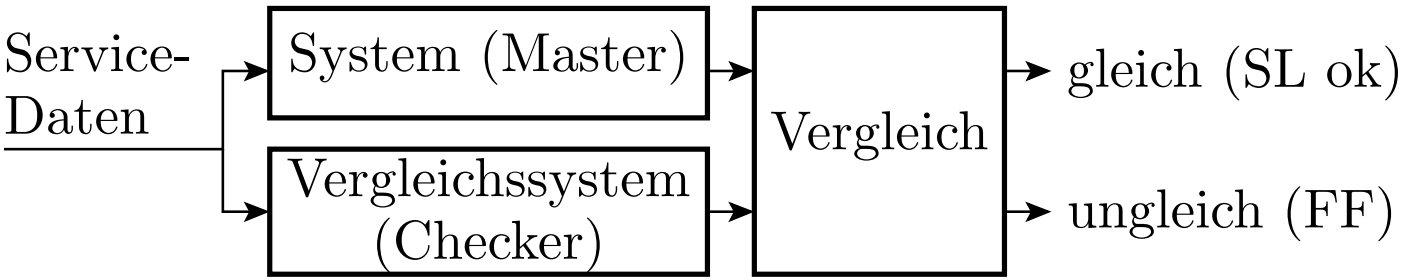


2. Eingaberückgewinnung aus der SL und Vergleich, z.B. Überwachung Versenden durch Empfang und Vergleich der empfangenen mit den Sendedaten:



3. Für SL vom Typ »Suche eine Lösung, die ein Korrektheitskriterium erfüllt« (z.B. einen Test der ein Fehler nachweist), Überwachung des Kriteriums.

Eigenschaften »Verdopplung und Vergleich«



Übereinstimmende Fehler als FF-Ursache verursachen gleiche FF. Erkennen setzt Verschiedenartigkeit voraus. Kenngröße Diversität:

$$\hat{Div} = \frac{\#DFF}{\#FF} \tag{9}$$

($\hat{.}$ – Schätzwert; $\#DFF$ – Anzahl der FF, bei denen eine Wiederholung. zu einem korrekten Ergebnis oder zu einer geänderten FF führt). Fehlfunktionsüberdeckung:

$$FFC \approx Div;$$

Phantom-FF-Rate gleich Rate der diversitären Checker-FF:

$$\zeta_{\text{Phan}} \approx \zeta_{\text{O}\ddot{\text{U}}} \cdot (1 - Div)$$

Diversität von Software-Versionen

Software-Fehler als Hauptquelle für FF verlangen Verschiedenartigkeit in den Entstehungsprozessen der beiden Versionen und ihrer Fehler:

- Komplette Entwicklung mindestens zweimal.
- durch getrennte Teams, keine Kommunikation,
- aus einer nicht diversitären Spezifikation, ...

Ursprüngliche euphorische Meinung, dass so Diversität gegenüber allen Fehlern, außer denen in der Spezifikation erzielbar sei, nicht bestätigt. Die direkte oder indirekte Kommunikation der Entwicklungsteams über die Interpretation der Spezifikation, während des Test etc. trägt Gemeinsamkeiten in die Entwürfe. Neigung von Menschen, gewisse Fehler zu wiederholen, ... Erzielbare Divisität laut[4]

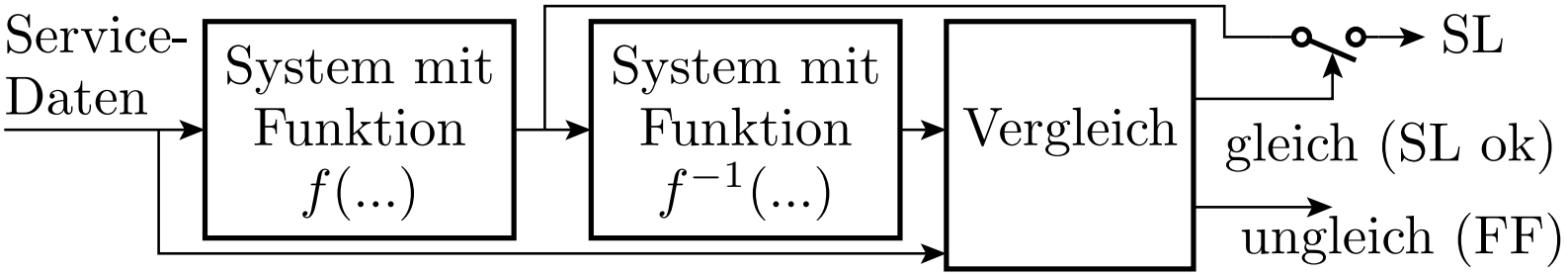
$$Div \approx FFC \leq 90\%$$

Eine Kontrolle mit $r = 10$ Bit Informationsredundanz erreicht bis zu $FFC \leq 99{,}9\%$ fast ohne Zusatzaufwand und ohne Phantom-FF.

[4] U. Voges, Software-Diversität und ihre Modellierung - Software-Fehlertoleranz und ihre Bewertung durch Fehler- und Kostenmodelle, Springer (1989)

Eigenschaften »Eingaberückgewinnung«

- Eingaberückgewinnung und Vergleich:

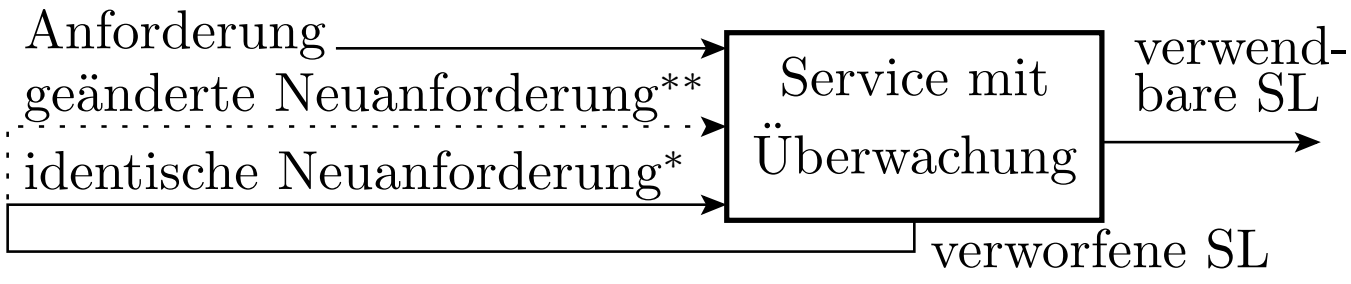


  - Höhere natürliche Diversität als Mehrfachberechung und Vergleich, weil $f(\ldots)$ und $f^{-1}(\ldots)$ in der Regel unterschiedliche, getrennt zu entwerfende Algorithmen sind.
  - Nur einsetzbar, wenn, $f(\ldots)$ eine umkehrbar eindeutige Abbildung ist. Besonders geeignet, wenn $f^{-1}(\ldots)$ viel einfacher als $f(\ldots)$ ist, z.B Wurzel $\Leftrightarrow$ Quadrat.
- Überwachung Korrektheitskriterium:
  - FF-Überdeckung und Phantom-FF-Rate ergeben sich aus der Erfolgsrate der Suche sowie der FF-Überdeckung und Phantom-FF-Rate der Kontrollfunktion.

Wenn einsetzbar, gute Überwachungsverfahren, aber ...

Korrekturverfahren

Wiederholung nach erkannter FF



* Erfolgswahrscheinlichkeit begrenzt durch Diversität
** Fehlerumgehung, in der Regel durch Benutzer

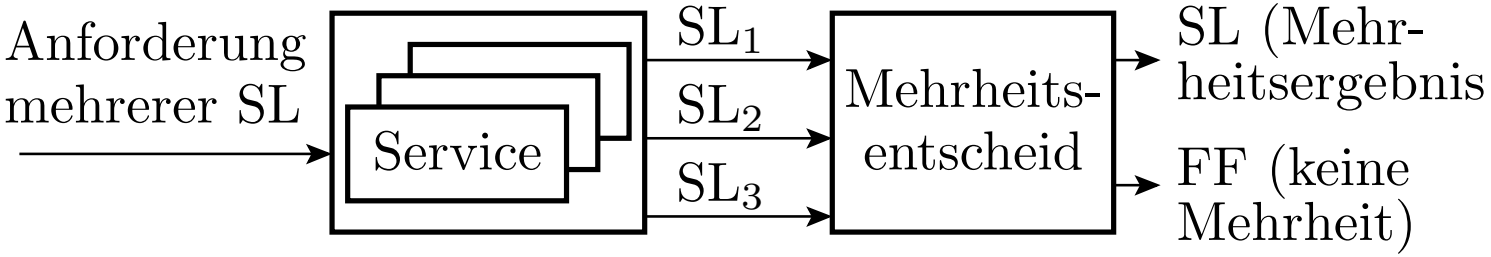
Fehlertoleranz (Erfolgshäufigkeit der Korrektur) bei identischer Neuanforderung:

$$FT \approx Div$$

(Div – Diversität), bei Fehlern als Hauptursache für FF gering. Geänderte Service-Anforderung (Fehlerumgehung[5]) erhöht die Diversität, ist aber schwer zu automatisieren.

[5]Fehlerumgehung: Aus der Vielzahl möglicher Arten, eine Aufgabe zu lösen, lernt ein Benutzer mit der Zeit, was funktioniert, und nutzt das System entsprechend. Eine alternative Service-Anforderung verlangt andere Eingaben, liefert kein direkt vergleichbares Ergebnis und erfordert in der Regel Bedienerinteraktionen.

Mehrfachberechung und Mehrheitsentscheid



- Voraussetzung für ein Mehrheitsergebnis sind identische SL.
- Fehlertoleranz FT (Erfolgshäufigkeit der Korrektur) auch nur in der Größenordnung der Diversität, hier aber für 3 Systeme.

Drei getrennte Rechner[6]:

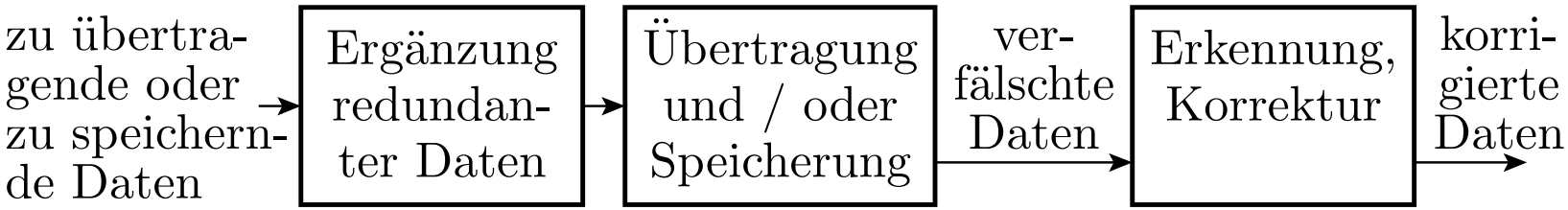
- hohe Diversität für FF durch Störungen und Ausfälle,
- fast keine gegenüber Entwurfsfehlern.

Unterschiedliche Rechnertypen und Betriebssystemen:

- auch Diversität gegenüber Hardware-Entwurfsfehlern und Fehlern im Betriebssystem.

[6] bereits 1956 von »von Neumann« vorgeschlagen.

Fehlerkorrigierende Codes



- Höhere Datenredundanz als fehlererkennde Codes. Einsatz für die Korrektur von Einzelbit- und Burst-Fehlern nach Datenübertragung und Speicherung.
- Gute Lösung für die Korrektur gespeicherter oder empfangener Daten. Für andere SL ungeeignet.
- Der Anteil FT der korrigierbaren FF hängt vom Code, der Anzahl der redundaten Datenbits und den zu erwartenden Verfälschungen ab.

Fehlerbeseitigung

Ursachen von FF

Ursachen von Fehlfunktionen

Fehler:

- Entstehen mit dem System oder bei Fehlerbeseitigungsversuchen. Sind permanent im System.
- Beseitungungserfolg kontrollierbar durch Testwiederholung.
- Fehlerbehandlung schwierig, da FF in der Regel nicht durch Wiederholung derselben SL mit demselben System korrigierbar.

Störungen:

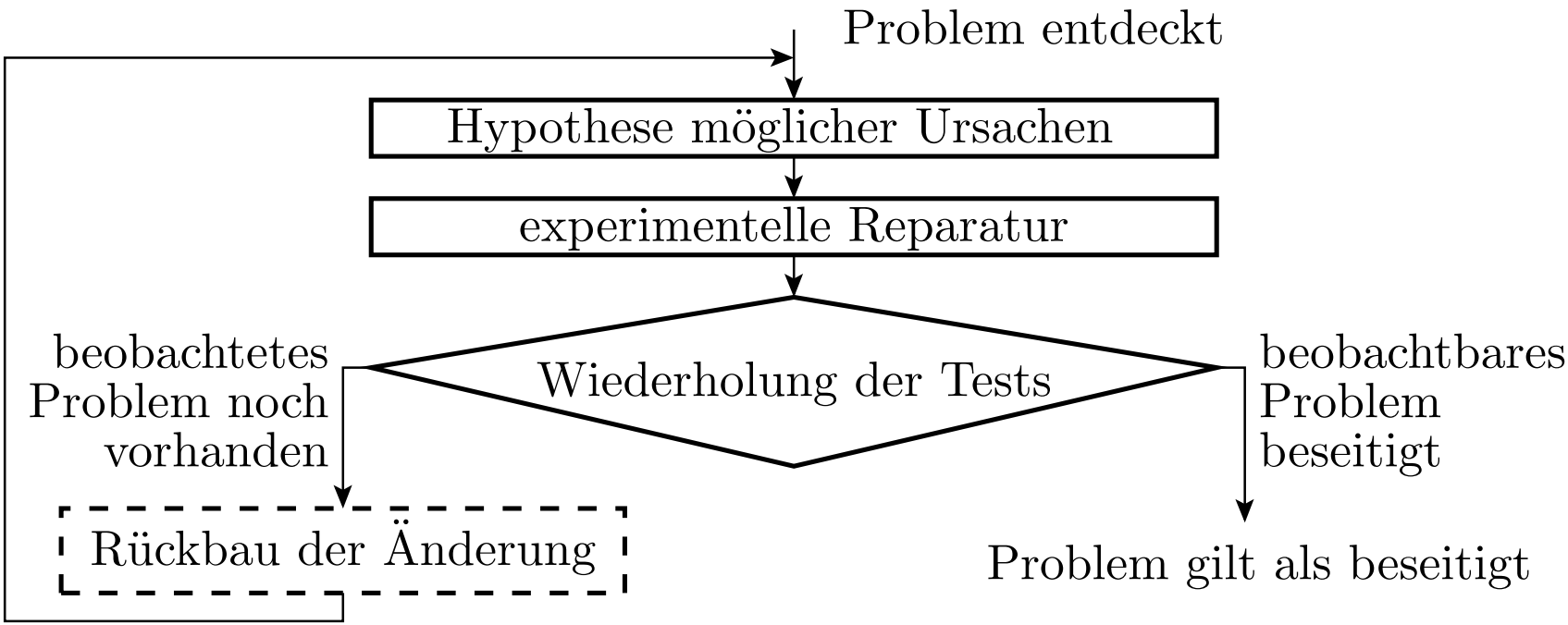
- Zufällige, nicht reproduzierbare Ursache-Wirkungs-Beziehungen.
- Fehlerbehandlung einfach, da FF durch Wiederholung derselben SL mit demselben System korrigierbar.
- Störquellen sind schwerer als Fehler zu beseitigen, allein weil sich der Beseitigungserfolg nicht durch Testwiederholung kontrollieren lässt.

Ausfälle:

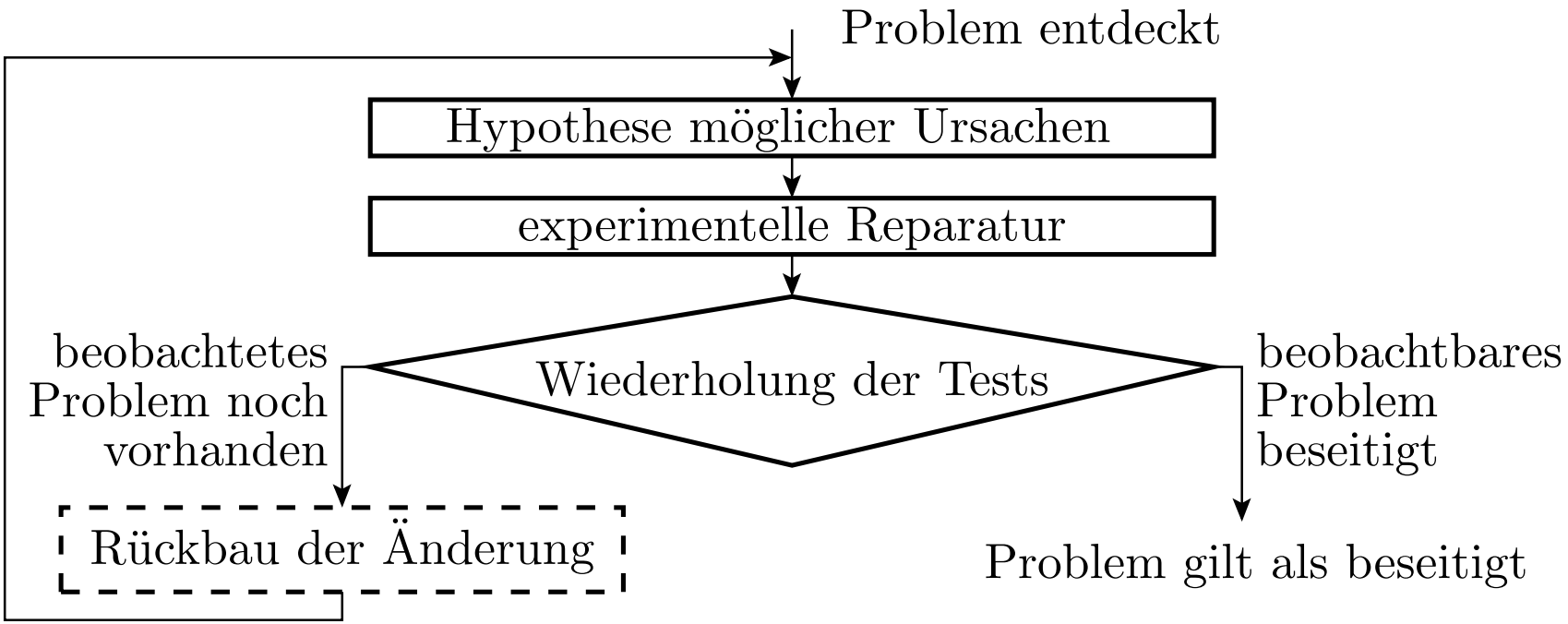
- während des Betriebs entstehende Fehler.

Experimentelle Reparatur

Experimentelle Reparatur



- Iteration aus Beseitigungsversuchen für hypothetische Fehler und Erfolgskontrolle durch Testwiederholung.
- Beseitigt alle vom Test nachweisbaren Fehler. Bei Reparaturversuchen können aber neue Fehler entstehen. Deshalb nach jedem erfolglosem Reparaturversuch Rückbau.



- Im Idealfall, wenn der Test keine Phantomfehler[7] ausweist und bei der Reparatur keine neuen Fehler entstehen, ist die Fehlerbeseitigungswahrscheinlichkeit gleich der Nachweiswahrscheinlichkeit des Tests.
- Abschätzungen unter Einbeziehung der bei Reparaturversuchen entstehenden Fehlern und der Wirkung von Phantomfehlern später nach der themenspezifischen Einführung in die Stochastik.

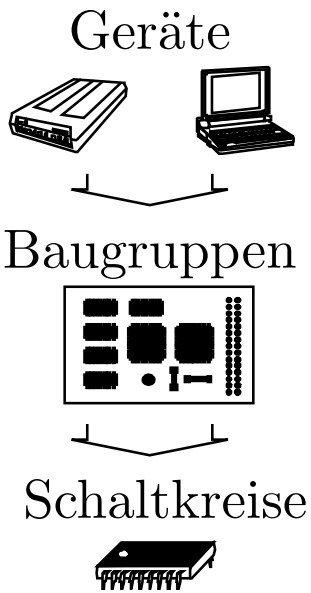
[7] Phantomfehler: Vermeindlicher Fehler, die kein Fehler ist.

Reparatur bei wenig tauschbaren Komponenten

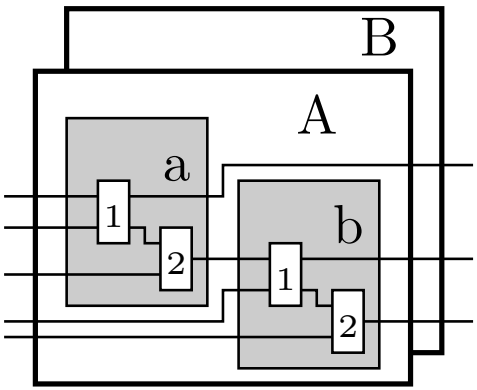
Ein reparaturgerechtes System hat eine hierarchische Struktur aus tauschbaren Komponenten, z.B.

1. Ebene: Austauschbare Geräte.
2. Ebene: Austauschbare Baugruppen.
3. Ebene: Austauschbare Schaltkreise.

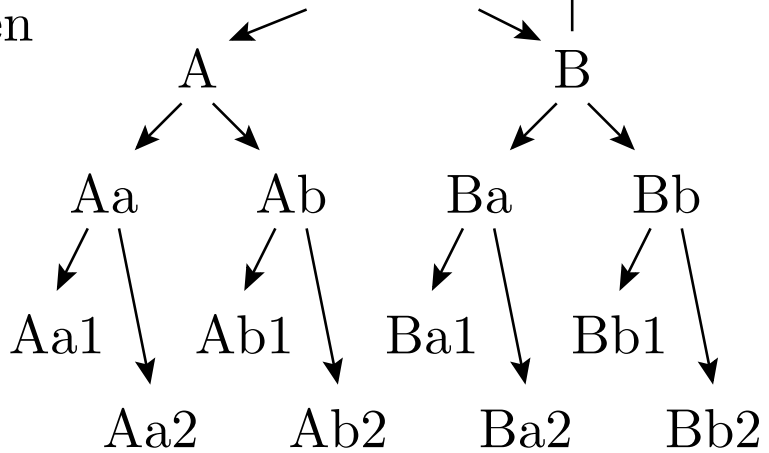
Fehlerlokalisierung durch systematisches Tauschen:



hierarchisches System mit tauschbaren Komponenten



Tauschbaum



Typisches Mechanikervorgehen:

- Grobabschätzung, welches Rechnerteil defekt sein könnte aus den Fehlersymptomen.
- Kontrolle der Steckverbinder auf Kontaktprobleme durch Abziehen, Reinigen, Zusammenstecken, Ausprobieren.
- Tausch möglicherweise defekter Baugruppen gegen Ersatzbaugruppen, Ausprobieren, ...

Voraussetzungen:

- Wiederholbare Tests, die den Fehler nachweisen.
- Ausreichend Ersatzteile. Allgemeine Mechnikerkenntnisse[8].

Wichtig ist der Rückbau nach jedem erfolglosen Reparaturversuch. Warum?

Günstig ist der Tausch der Hälfte, von der fehlerhaften Hälfte wieder der Hälfte, ... Warum?

[8] Verständnis der Funktion des zu reparierenden Systems nicht zwingend.

Fehlerdiagnose

Fehlerdiagnose

Bestimmung von Ort- und Ursache eines Fehlers.

- Erfassen der beobachtete Symptomem (Fehlfunktionen),
- Suche von Tests, die die FF reproduzierbar anregen.
- Abschätzen wahrscheinlicher Ursachen und/oder
- Rückverfolgung von Verfälschungen bis zur Quelle.
- experimentelle Reparatur.

Da jede Diagnose durch die Testwiederholung nach dem Reparaturversuch kontrolliert wird, ist es nicht so schwerwiegend, wenn im Mittel mehrere Beseitigungsversuch je Fehler erforderlich sind.

Die wichtigsten Fehlerlokalisierungstechniken:

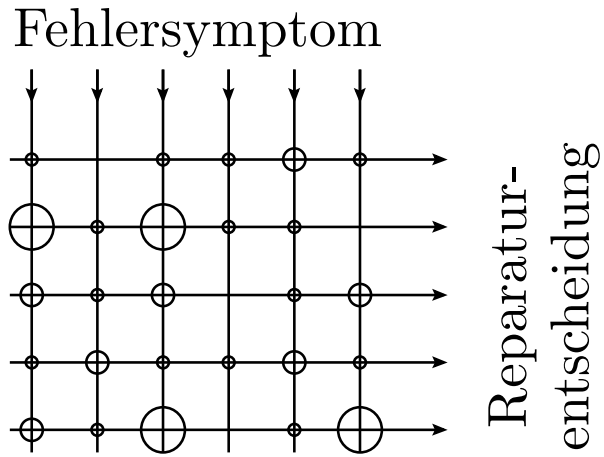
- Ausnutzung des Parato-Prinzips.
- Rückverfolgung.

Pareto-Prinzip

Produkte haben Schwachstellen. Richtwert: 80% der Probleme geht auf 20% der Ursachen zurück.

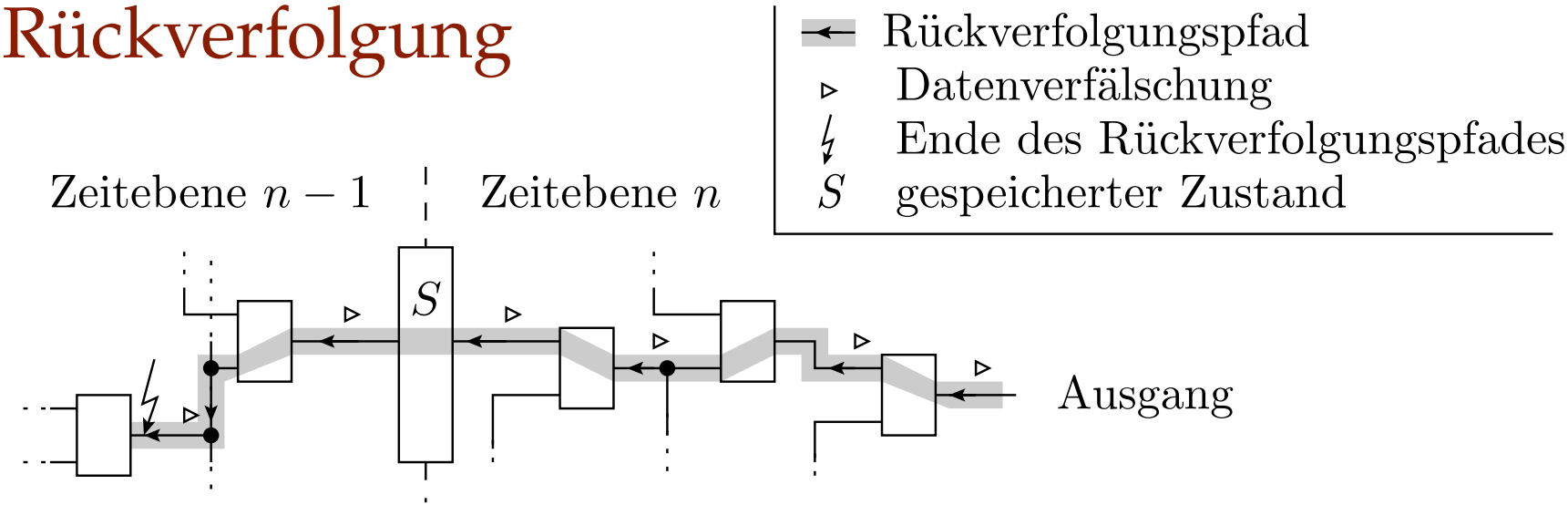
- Zählen der erfolgreichen und erfolglosen Reparaturversuche.
- Bei Alternativen, Beginn mit der erfolgsversprechendsten Reparaturmöglichkeit.

◎ bisherige Häufigkeit, mit der die Reparaturentscheidung für das Symptom richtig war



Nach erfolglosen Reparaturversuchen Rückbau der Änderung, um Entstehung neuer Fehler zu mindern.

Rückverfolgung



- Ausgehend von einer erkannten falschen Ausgabe Rückwärtsuche nach dem Entstehungsort, gegebenenfalls über Zeitebenen.
- Suche endet am Teil-Service, der aus richtigen Eingaben falsche Ergebnisse erzeugt.
- Tausch oder weiter hierarchisch absteigende Suche.
- Verfälschungsursache kann außer dem erzeugenden Service auch ein anderer, z.B. mit fehlgeleitetem Schreibzugriff, sein.

Test

Testen

Verfahren zum Aufspühren von Fehlern.

- Statische Tests: direkte Kontrolle von Merkmalen.
- Dynamischer Tests: Ausprobieren der Systemfunktion mit einer Stichprobe von Beispieleingaben.

Statisch kontrollierbare Merkmale

- Dokumentationen: Verständlichkeit, Vollständigkeit, ...
- Software: Review, Syntax, Entwurfsregeln, ...
- Baugruppen: Bestückung, Verdrahtung, ...

Statische Tests sind bereits nach Teilschritten des Entwurfs möglich, dynamische Tests erst am funktionsfähigen Produkt.

Vor dem Einsatz werden Systeme in der Regel einer Vielzahl von unterschiedlichen statischen und dynamischen Tests unterzogen.

Kenngrößen von Tests

Wie jede Kontrolle mit einem gut/schlecht-Ergebnis gibt es bei Tests zwei Arten von Fehlklassifikationen:

- Nichterkennen von Fehlern. Kenngröße Fehlerüberdeckung (Fault Coverage, Anteil der nachweisbaren Fehler):

$$\hat{FC} = \frac{\#EF}{\#F} \tag{10}$$

  ($\#EF$ – Anzahl der nachweisbaren Fehler; $\#F$ – Anzahl der vorhandenen Fehler).
- Phantomfehler. Klassifizierung fehlerfreier Testergebnisse als fehlerhaft. Kenngröße Phantomfehlerrate:

$$\hat{\varphi}_{\mathrm{Phan}} = \frac{\#PF}{\#T} \tag{11}$$

  ($\#PF$ – Anzahl der Phantomfehler, $\#T$ – Anzahl der durchgeführten Tests).

Beispiel 4

Programmgröße 10.000 NLOC. 10 ... 100 Fehler je 1000 NLOC. Fehlerüberdeckung der Tests $FC = 70\%$. Zu erwartende Fehleranzahl nach Beseitigung aller erkennbaren Fehler:

$$10.000\,\mathrm{NLOC} \cdot \frac{10\,\mathrm{F} \ldots 100\,\mathrm{F}}{1000\,\mathrm{NLOC}} \cdot (1 - 70\%) = 30\,\mathrm{F} \ldots 300\,\mathrm{F}$$

Ob ein System mit 30 bis 300 Fehlern zuverlässig oder unzuverlässig ist, hängt von den FF-Raten der nicht erkannten Fehler ab.

- Der Anteil der (nicht) nachweisbaren Fehler hängt bei statischen Tests vom Kontrollverfahren und bei dynamischen Tests von der Anzahl und Auswahl der Testbeispiele ab.
- Die FF-Rate nicht erkannter / beseitigter Fehler hängt bei dynamischen Tests gleichfalls von Anzahl und Auswahl der Tests ab.
- Phantomfehler verursachen (überflüssige) Reparaturversuche, bei denen neue Fehler entstehen können (»kaputt reparieren«).

Fehlerorientierte oder zufällige Testauswahl

Dynamischen Tests legen Eingaben an und überwachen Ausgaben. Ausgabeüberwachung meist Soll/Ist-Vergleich, der alle FF während des Tests erkennt. Für die Auswahl der Testeingaben sind folgende Strategien zu unterscheiden:

- fehlerorientiert: geziele Suche von Eingaben für den Fehlernach-. weis. Fehlerüberdeckung abhängig vom Sucherfolg und, wie gut die Fehlerannahmen tatsächliche Fehler widerspiegeln
- zufällige: Auswahl unabhängig von Fehlerannahmen, willkürliche/typische/zufällige Werte. Fehlerüberdeckung hängt hauptsächlich von der Testsatzlänge ab.
- Mischformenen, bei den fehlernachweisende Eingaben bevorzugt werden.

Die Fehler in einem System sind immer erst nach ihrer erfolgreichen Beseitigung genau bekannt. Fehlerannahmen für die Testauswahl:

- Stichprobe möglicher Fehler.
- Modellfehler.

Haftfehler

Das Haftfehlermodell

Ein Fehlermodell definierte abzählbare Mengen von simulierbaren Fehlerannahmen, die ähnlich wie potentielle Fehler nachweisbar sind.

Das am weitesten verbreitete Fehlermodell ist das Haftfehlermodell, entwickelt für Hardware. Für jeden Eingang eines Logikgatters Annahme von zwei Modellfehlern:

- Wert ständig null (sa0, stuck-at-0) und
- Wert ständig eins (sa1, stuck-at-1).

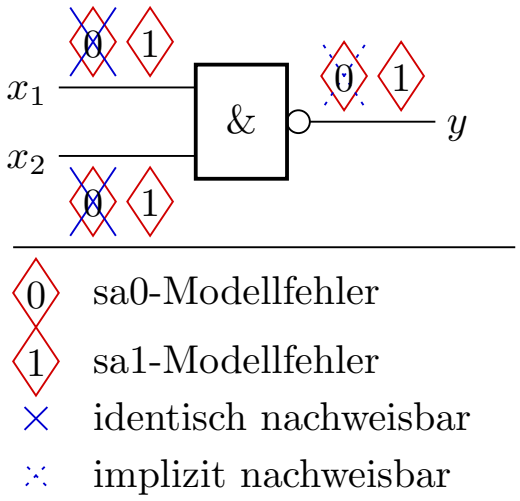
Erprobt seit 4 bis 5 Jahrzehnten für die Fehlersimulation und Testsatzberechung für große digitale Schaltungen.

Neuere systematische Techniken für die Testauswahl und Bewertung für Software auf das Haftfehlermodell zurückführbar, so dass die lange bewährten Techniken übernehmbar sind.

Haftfehler für Logikgatter

Für jeden Gatteranschluss wird unterstellt:

- ein sa0 (stuck-at-0) Fehler
- ein sa1 (stuck-at-1) Fehler



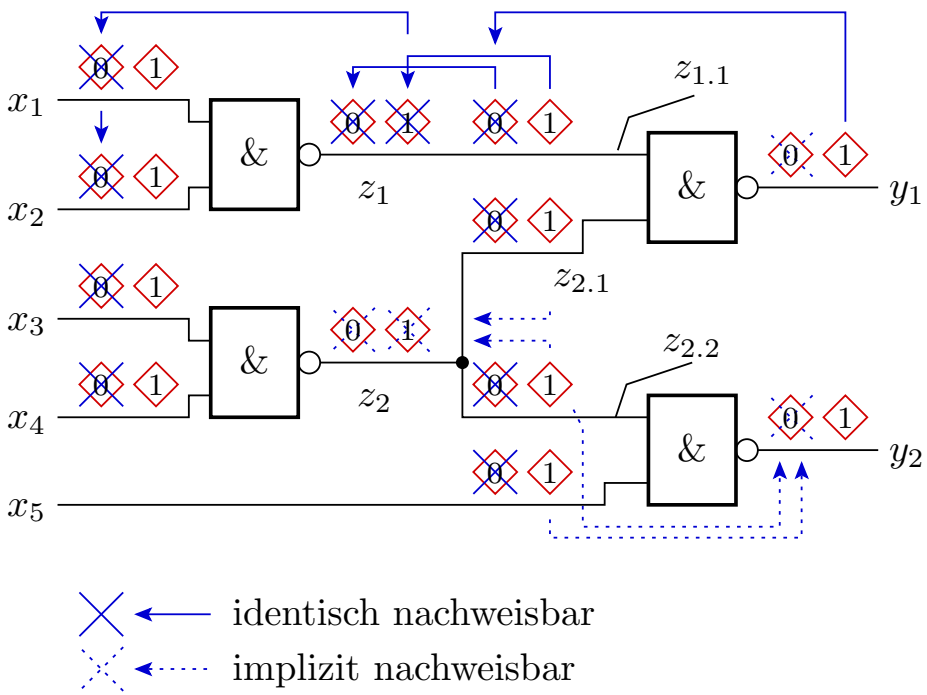
x_2	x_1	$\overline{x_2 \wedge x_1}$	sa0(x_1)	sa1(x_1)	sa0(x_2)	sa1(x_2)	sa0(y)	sa1(y)
0	0	1	1	1	1	1	0	1
0	1	1	1	1	1	0	0	1
1	0	1	1	0	1	1	0	1
1	1	0	1	0	1	0	0	1
				Nachweisidentität (gleiche Nachweismenge)				

Nachweisimplikation

zugehörige Eingabe ist Element der Nachweismenge

Zusammenfassung identisch nachweisbarer Fehler. Optionale Streichung redundanter und implizit nachweisbarer Modellfehler. Modellierte Fehler sind ähnlich wie Transistorfehler in Gattern nachweisbar.

Streichen identischer und implizit nachweisbarer Fehler



Größe der Anfangsfehlermenge:	24
Anzahl der nicht identisch nachweisbaren Fehler:	14
ohne implizit nachgewiesene Fehler:	10

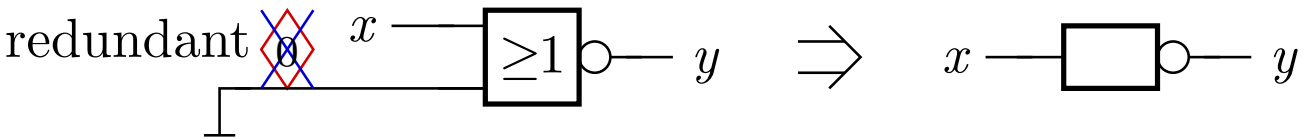
	Mengen von identisch nachweisbaren Fehlern	Nachweis impliziert durch
1	sa0(x_1), sa0(x_2), sa1(z_1), sa1($z_{1.1}$)	
2	sa1(x_1)	
3	sa1(x_2)	
4	sa0(x_3), sa0(x_4), sa1(z_2)	9, 12
5	sa1(x_3)	
6	sa1(x_4)	
7	sa0(z_2)	5, 6, 8, 11
8	sa0(z_1), sa0($z_{1.1}$), sa0($z_{2.1}$), sa1(y_1)	2, 3
9	sa1($z_{2.1}$)	
10	sa0(y_1)	1, 9
11	sa0($z_{2.2}$), sa0(x_5), sa1(y_2)	
12	sa1($z_{2.2}$)	
13	sa1(x_5)	
14	sa0(y_2)	12, 13

Redundante Fehler

Definition redundanter (Modell-) Fehler

Fehler in einem Teilsystem, der die Funktion des Gesamtsystems nicht beeinträchtigt.

- Der Gatteranschluss kann mit »0« (sa0-Fehler nicht nachweisbar) bzw. »1« (sa1-Fehler nicht nachweisbar) verbunden sein, ohne dass sich die Funktion ändert.
- Umformungen zur Beseitigung redundanter Modellfehler dienen auch zur Systemoptimierung.



Bestimmung der Modellfehlerüberdeckung

- Erzeugen einer Modellfehlermenge. Jeder Modellfehler beschreibt eine Schaltung, bei der ein Gatteranschluss statt mit seiner Signalquelle fest mit 0 oder 1 verbunden ist.
- (zufällige) Auswahl eines Testsatzes (Menge von Tupeln aus Eingabe und Sollausgabe).
- Wiederhole für jeden Modellfehler:
  - Wiederhole für jedes Testeingabe-Sollausgabe-Tupel:
    - Simulation des Testobjekts mit Ergebniskontrolle
    - Wenn Ausgabe verfälscht, Modellfehler als nachweisbar abhakten.

Simulationsaufwand $\sim$ Testsatzlänge $\cdot$ Modellfehleranzahl

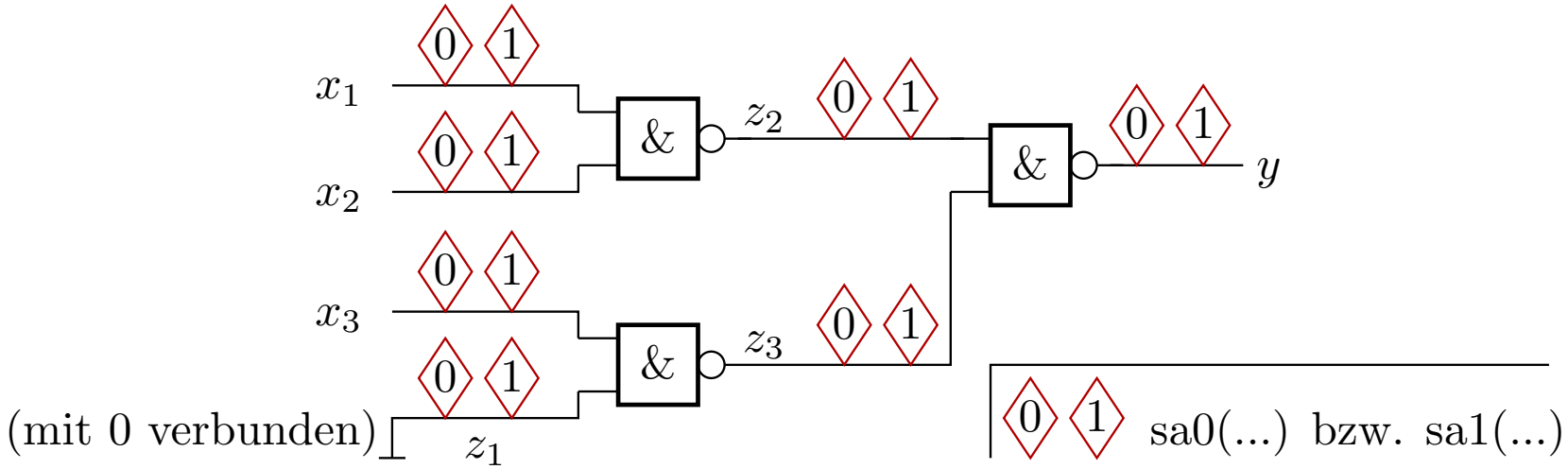
---

Fehlerorientierte Testsuche

- Wiederhole für jede Testobjektversion mit Modellfehler
  - Suche Eingaben, bei den eine Fehlfunktion auftritt.

Beispielaufgabe

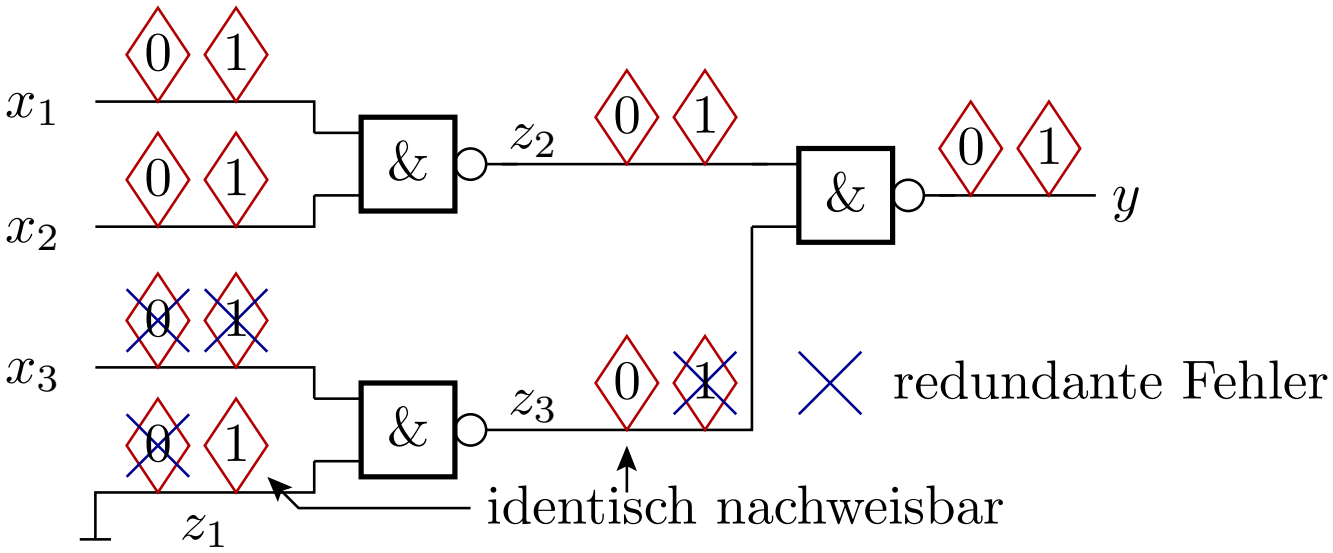
Gegeben ist die nachfolgende Schaltung mit 12 eingezeichneten Haftfehlern.



Welche der Haftfehler sind

1. redundant, d.h. mit keiner Eingabebelegung nachweisbar,
2. nach einer Konstanteneliminierung identisch nachweisbar,
3. implizit durch die Tests anderer Haftfehler nachweisbar?

Lösung Aufgabenteil 1

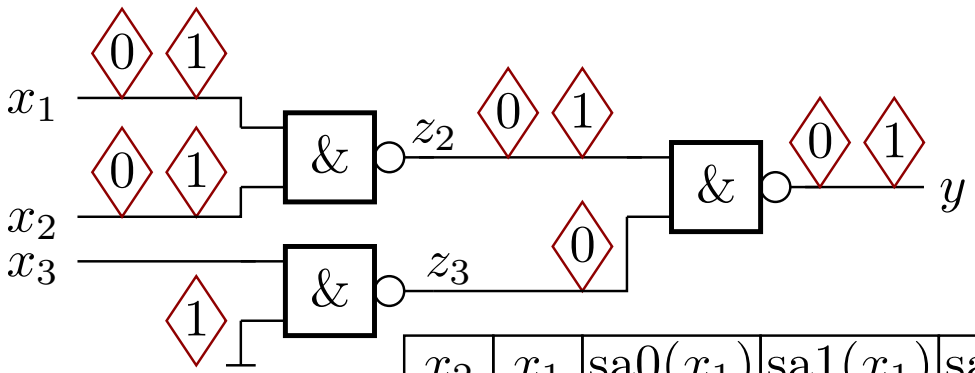


$z_1 = 0$ impliziert, dass

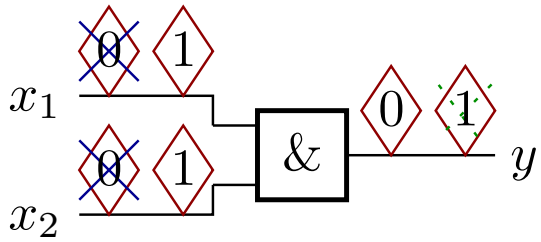
- sa0(z_1) nicht anregbar ist,
- $z_3 = 1$, so dass sa1(z_3) nicht anregbar ist und
- dass x_3 nicht beobachtbar ist, so dass sa0(x_3) und sa1(x_3) auch redundant sind.

Lösung Aufgabenteil 2 und 3

Schaltung ohne redundante Fehler



nach Konstanteneliminierung



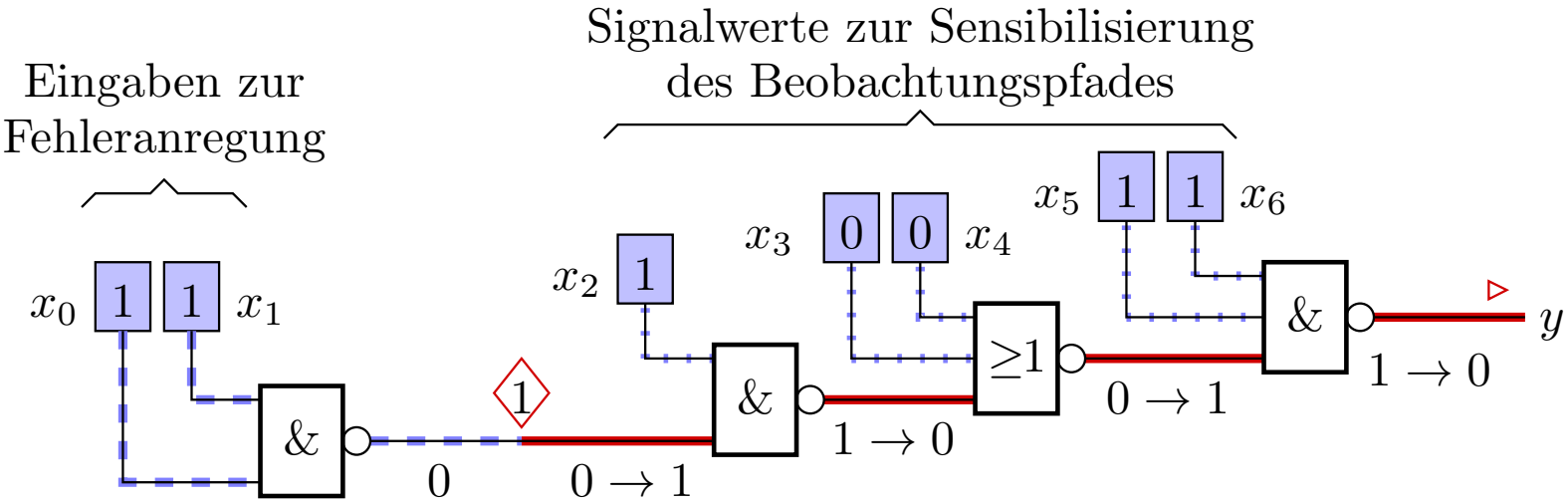
— identischer Nachweis

— impliziter Nachweis

x_2	x_1	sa0(x_1)	sa1(x_1)	sa0(x_2)	sa1(x_2)	sa0(y)	sa1(y)
0	0	–	–	–	–	–	+
0	1	–	–	–	+	–	+
1	0	–	+	–	–	–	+
1	1	+	–	+	–	+	–

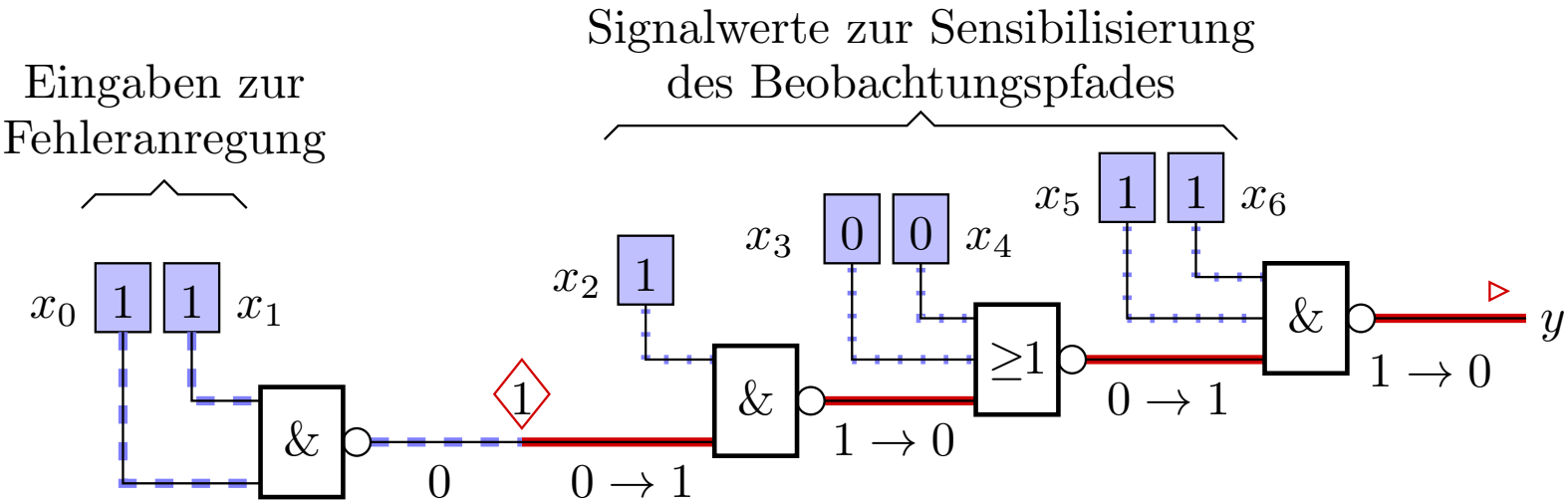
Die Fehlermenge ohne redundante, identisch und implizit nachweisbare Haftfehler umfasst sa1(x_1), sa1(x_2) und sa0(y).

Nachweisbedingungen für einen Haftfehler



- Signalwerte für den Fehlernachweis
- Fehler (ständig 1, stuck-at 1)
- Fehlfunktion (Ausgabeinvertierung)
- Eingaben zur Fehleranregung
- Einstellung der Beobachtbarkeit
- Beobachtungspfad

Eingabemenge Fehleranregung: $M_1 = \{\text{-----11}\}$
Eingabemenge Beobachtbarkeit: $M_2 = \{\text{11001--}\}$
Fehlernachweismenge: $M_1 \cap M_2 = \{1100111\}$



- Signalwerte für den Fehlernachweis
- Fehler (ständig 1, stuck-at 1)
- Fehlfunktion (Ausgabeinvertierung)
- Eingaben zur Fehleranregung
- Einstellung der Beobachtbarkeit
- Beobachtungspfad

- Geziele Suche durch »Pfadsensibilisierung« (siehe später Foliensatz 6)
- Zufallstest: Nachweis mit einer von 2^7 möglichen Eingaben
  - FF-Rate $\approx 2^{-7}\,\frac{\mathrm{FF}}{\mathrm{SL}}$
  - mittlere Testsatzlänge für den Fehlernachweis 2^7.

Test und Zuverlässlichkeit

Fehlfunktionsrate durch Fehler

Ziel: Herleitung Zuverlässigkeitswachstumsgesetz

$$(1 - FC) \sim n^{-k} \Rightarrow Z \sim S \sim n^{k+1} \quad \text{bzw.} \quad Z \sim S \sim \frac{n}{1 - FC}$$

Jeder Fehler i verursacht mit der FF-Rate ζ_i (in FF je SL) Fehlfunktionen. Die Summe der FF-Raten aller Fehler

$$\zeta_{\sum} = \sum_{i=1}^{\#F} \zeta_i$$

($\#F$ – Anzahl der Fehler) ist eine Obergrenze für die gesamte FF-Rate durch Fehler $\zeta \leq \zeta_{\sum}$ und für $\zeta_{\sum} \ll 1$ (dieselbe FF hat fast immer nur einen Fehler als Ursache) eine gute Abschätzung für die gesamte FF-Rate durch Fehler:

$$\zeta_{\mathrm{F}} = \sum_{i=1}^{\#F} \zeta_i \quad \text{für} \quad \zeta \ll 1$$

Dichte und Verteilung der FF-Rate

Die Dichte der FF-Rate

$$h(\zeta) \text{ mit } \int_0^1 h(\zeta) \cdot d\zeta = 1$$

ordnet jedem Wert der FF-Rate $0 < \zeta \leq 1$ die relative Häufigkeit zu, mit der Fehler diese FF-Rate besitzen. Die Verteilungsfunktion der FF-Rate ist die relative Häufigkeit, dass ein Fehler mindestes die FF-Rate ζ besitzt:

$$F(\zeta) = \int_0^\zeta h(x) \cdot dx$$

Die Summe alle FF-Raten und für $\zeta \ll 1$ die Gesamt-FF-Rate durch Fehler:

$$\zeta_\mathrm{F} = \sum_{i=1}^{\#F} \zeta_i = \#F \cdot \underbrace{\int_0^1 \zeta \cdot h(\zeta) \cdot d\zeta}_{\text{mittlere FF}-\text{Rate je Fehler}}$$

($\#F$ – Fehleranzahl).

Typische Verteilung der FF-Rate

Bei einem Zufallstest erfordert eine Verringerung des Anteil der nicht nachweisbaren Fehler um eine Dekade eine Erhöhung der Testsatzlänge um mehr als eine Dekade. Nachbildbar durch die Potenzfunktion:

$$1 - FC\,(n) = \left(\frac{n}{n_0}\right)^{-k} \quad \text{mit } 0 < k < 1$$

$\frac{n}{n_0}$ für $1 - FC\,(n) = 0{,}5$	2	4	8	16
k	1	0,5	0,25	1/8

Unter der vereinfachten Annahme, dass ein Zufallstest der Länge n alle Fehler mit einer FF-Rate

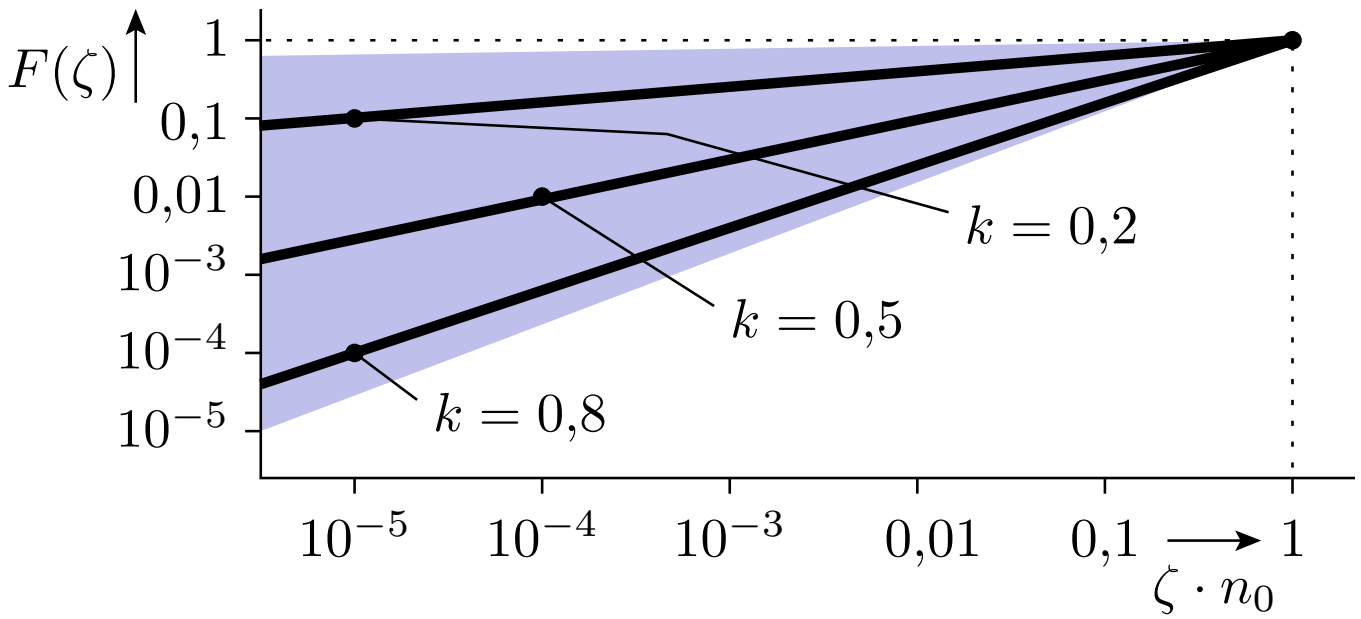
$$\zeta \geq \frac{1}{n} \cdot {}^{\mathrm{FF}}/_{\mathrm{SL}}$$

erkennt, folgt daraus für die Verteilungsfunktion der FF-Rate ζ :

$$F\,(\zeta) = 1 - FC\left(n = \frac{1}{\zeta} \cdot {}^{\mathrm{FF}}/_{\mathrm{SL}}\right) = \left(\zeta \cdot n_0 \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}}\right)^k \quad \text{für } 0 < \zeta \leq \frac{1}{n_0 \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}}}$$

Verteilungsfunktion:

$$F(\zeta) = (\zeta \cdot n_0 \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}})^k \quad \text{für } 0 < \zeta \leq \frac{1}{n_0 \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}}}$$



Dichte:

$$h(\zeta) = \frac{dF(\zeta)}{d\zeta} = k \cdot n_0 \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}} \cdot (\zeta \cdot n_0 \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}})^{k-1} \quad \text{für } 0 < \zeta \leq \frac{1}{n_0 \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}}}$$

$$h(\zeta) = k \cdot n_0 \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}} \cdot \left(\zeta \cdot n_0 \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}}\right)^{k-1} \quad \text{für } 0 < \zeta \leq \frac{1}{n_0 \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}}} \tag{12}$$

Die Summe alle FF-Raten und für $\zeta \ll 1$ die Gesamt-FF-Rate nach Beseitigung aller mit $n \geq n_0$ nachweisbaren Fehler:

$$\zeta_{\mathrm{F}} = \sum_{i=1}^{\#F} \zeta_i = \#F \cdot \underbrace{\int_0^1 \zeta \cdot h(\zeta) \cdot d\zeta}_{\text{mittlere FF}-\text{Rate je Fehler}}$$

$$= \#F \cdot \int_0^{\frac{1}{n \cdot \mathrm{SL}/\mathrm{FF}}} \zeta \cdot k \cdot n_0 \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}} \cdot \left(\zeta \cdot n_0 \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}}\right)^{k-1} \cdot d\zeta$$

$$= \#F \cdot k \cdot \int_0^{\frac{1}{n \cdot \mathrm{SL}/\mathrm{FF}}} \left(\zeta \cdot n_0 \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}}\right)^{k} \cdot d\zeta$$

$$\zeta_{\mathrm{F}} = \frac{\#F \cdot k}{n_0 \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}} \cdot (k+1)} \cdot \left(\frac{n_0}{n}\right)^{k+1}$$

FF-Rate durch Fehler Beseitigung aller Fehler mit $\zeta \geq \frac{1}{n}$:

$$\zeta_{\mathrm{F}} = \frac{\#F \cdot k}{n_0 \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}} \cdot (k+1)} \cdot \left(\frac{n_0}{n}\right)^{k+1}$$

mit $= \left(\frac{n}{n_0}\right)^{-k} = 1 - FC\,(n)$

$$\zeta_{\mathrm{F}} = \#F \cdot \frac{k}{k+1} \cdot \frac{1 - FC}{n \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}}}$$

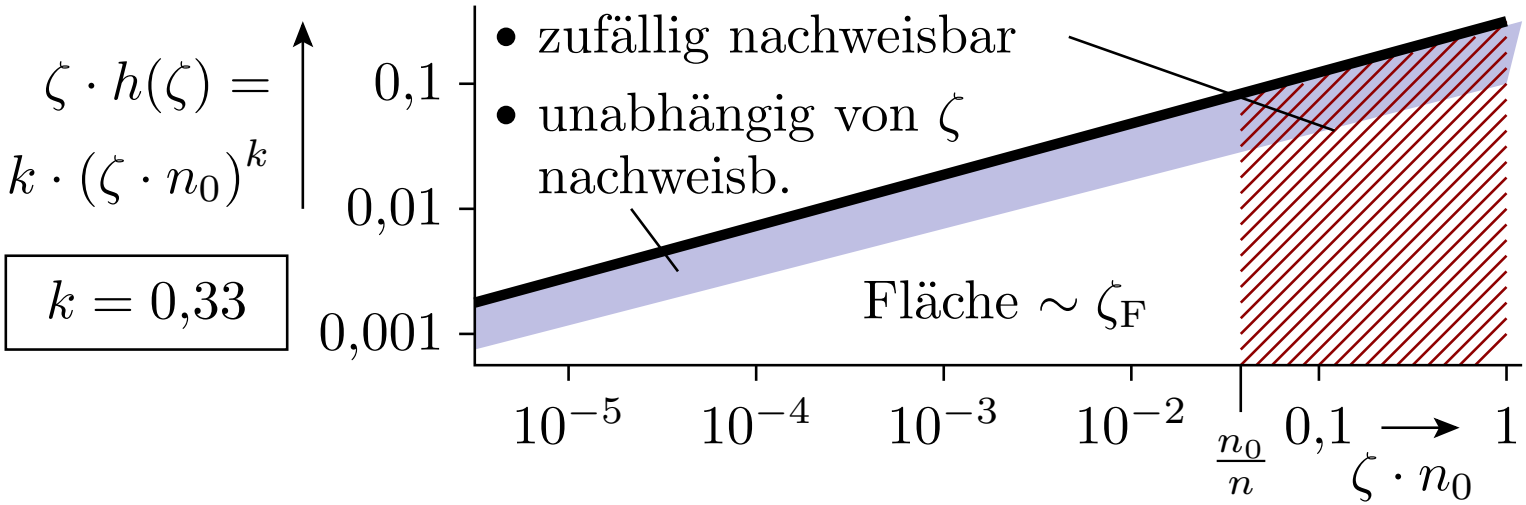
$$Z_{\mathrm{F}} = \frac{1}{\zeta_{\mathrm{F}}} = \frac{k+1}{\#F \cdot k} \cdot \frac{n \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}}}{1 - FC}$$

($\#F$ – Fehleranzahl vor dem Test; FC – Fehlerüberdeckung; n – Anzahl der Tests; k – Exponent, mit dem die Anzahl der nichtnachweisbaren Fehler bei einem Zufallstest mit der Testlänge abnimmt; Z – fehlerbezogene Teilzuverlässigkeit).
Hohe Zuverlässigkeit verlangt:

- geringe Fehleranzahl $\#F$ vor dem Test,
- hohe Fehlerüberdeckung FC aller Tests zusammen,
- große Testanzahl n.

Der Exponent k für die Testobjekteigenschaften hat kaum Einfluss.

Test und fehlerbezogene Teilzuverlässlichkeit



- statische Tests: Großer Beitrag zu FC, kein Beitrag zu n.
- fehlerorientiert gesuchte Tests: Großer Beitrag zu FC und kleiner Beitrag zur Testsatzlänge n für den zufälligen Nachweis unberücksichtigter Fehler.
- Zufallstest: Kleiner Beitrag zur Fehlerüberdeckung und ein großer Beitrag zur Testsatzlänge n für den zufälligen Nachweis in

$$Z_F = \frac{k+1}{\#F \cdot k} \cdot \frac{n \cdot {}^{SL}/_{FF}}{1 - FC}$$

Reifeprozesse

FF-Rate incl. Störungen und Fehlertoleranz

In einem komplexen IT-System kommt zur FF-Rate durch Fehler ζ_{F} eine durch Störungen verursachte FF-Rate ζ_{S} hinzu. Die fehlerbedingte FF-Rate lässt sich durch Testen und die störungsbedingte FF-Rate durch Fehlertoleranz (z.B. Überwachung + Wiederholung) mindern:

$$\zeta = \frac{\zeta_{\mathrm{S}}}{1 - FT} + \#F \cdot \frac{k}{k+1} \cdot \frac{1 - FC}{n \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}}}$$

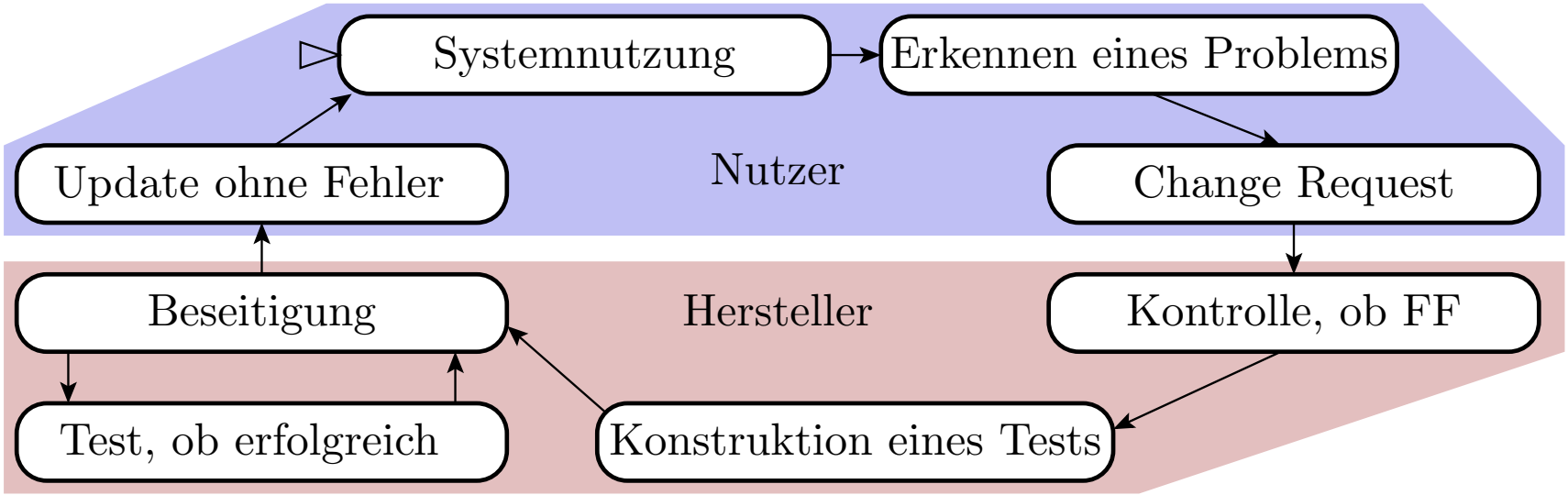
Die Anzahl der entstehenden Fehler $\#F$ nimmt mindestens proportional mit der Systemgröße $\#$ zu (siehe später Absch. Fehlervermeidung). Richtwert für den Sofware-Entwurf sind 10 bis 100 entstehende Fehler je 1000 NLOC (Netto Lines of Code). Für große Software-Systeme:

$$\#F = 10^3 \ \ldots \ 10^6$$

Die Fehlermaskierung $1 - FC$ lässt sich nicht um Zehnerpotenzen senken. Folglich muss zur Kompensation die Anzahl der dynamischen Tests n mindesten proportional zur Systemgröße erhöht werden.

Reifeprozesse

Fortsetzung der Iteration aus Zufallstest und Fehlerbeseitigung in der Anwendungsphase mit den Service-Anforderungen der Anwender.



Fehlerbeseitigungsiteration für von Anwendern beobachtete FF:

- Erfassen der FF mit allen Daten, um die FF nachzustellen,
- Übermittlung an den Hersteller,
- Priorisierung, Fehlersuche und Beseitigung,
- Herausgabe und Einspielung von Updates.

- Bei einer vermuteten Fehlfunktion stellt der Nutzer einen Änderungsanforderung (Change Request).
- Der Hersteller prüft diese, selektiert daraus FFs und versucht, für jede FF reproduzierbare Testbeispiele zu finden.
- Die Testbeispiele dienen zur Fehlerlokalisierung und zur Erfolgskontrolle nach jedem Beseitigungsversuch.
- Fehlerbeseitigung beim Nutzer erfolgt über Einspielen von Updates, in seltenen Ausnahmen über eine Rückrufaktion für Hardware oder komplette Geräte.

Kenngrößen Reifeprozess:

- Beseitigungswahrscheinlichkeit, dass bei einer erkennbaren FF der zugrunde liegende Fehler beseitigt wird:

$$p_{\mathrm{BE}} = \frac{\#FF_{\mathrm{BR}}}{\#FF_{\mathrm{N}}} \ll 1$$

  ($\#FF_{\mathrm{N}}$ – Anzahl der aufgetretenen FF; $\#FF_{\mathrm{BR}}$ – Anzahl der FF davon, für die der verursachende Fehler beseitigt wurde).
- n_{R} – Anzahl der von allen Anwendern insgesamt genutzten Service-Leistungen.

FF-Rate und Zuverlässigkeit durch Reifeprozess

$$\zeta = \zeta_{\mathrm{S}} \cdot (1 - FT) + \#F \cdot \frac{k}{k+1} \cdot \frac{(1 - FC)^{*1}}{\left(n + (p_{\mathrm{BE}} \cdot n_{\mathrm{R}})^{*2}\right) \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}}}$$

*1 – Abnahme der Fehlermaskierung $1 - FC \sim \left(p_{\mathrm{BE}} \cdot n_{\mathrm{R}} \cdot {}^{\mathrm{SL}}/_{\mathrm{FF}}\right)^{-k}$ mit $1 < k < 1$; *2 – effektive Erhöhung der Testanzahl. Für ein bei vielen Nutzern über Jahre eingesetztes System ist $p_{\mathrm{BE}} \cdot n_{\mathrm{R}}$ um Zehnerpotenzen größer als die Anzahl der dynamischen Tests n beim Hersteller. Die fehlerbezogene Teilzuverlässigkeit nimmt überproportional mit der Reifedauer t_{R} und der Nutzeranzahl $\#N$ zu:

$$Z_{\mathrm{F}} \sim \left(\#N \cdot t_{\mathrm{R}}\right)^{k+1} \tag{13}$$

- Systeme, die viele Jahre gereift sind, haben hohe, auf anderem Wege unerreichbare Zuverlässigkeiten. Schwer ersetzbar durch neue Systeme. (siehe Jahr2000-Problem).
- Neue / alternative Systeme sind in den ersten Nutzungsjahren vielfach viel unzuverlässer als die Systeme, die sie ersetzen. Wenn das die Akzeptanz beeinträchtigt, reifen sie auch nicht ...

Lernprozesse der Benutzer

Bei der Einarbeitung in ein neues IT-System ist es typisch, dass zu Beginn häufig FF und mit zunehmender Nutzung immer seltener FF auftreten, weil der Nutzer lernt, die Fehler und Schwachstellen im System zu umgehen. Auch hier Zuverlässigkeitswachtum nach Gl. 13:

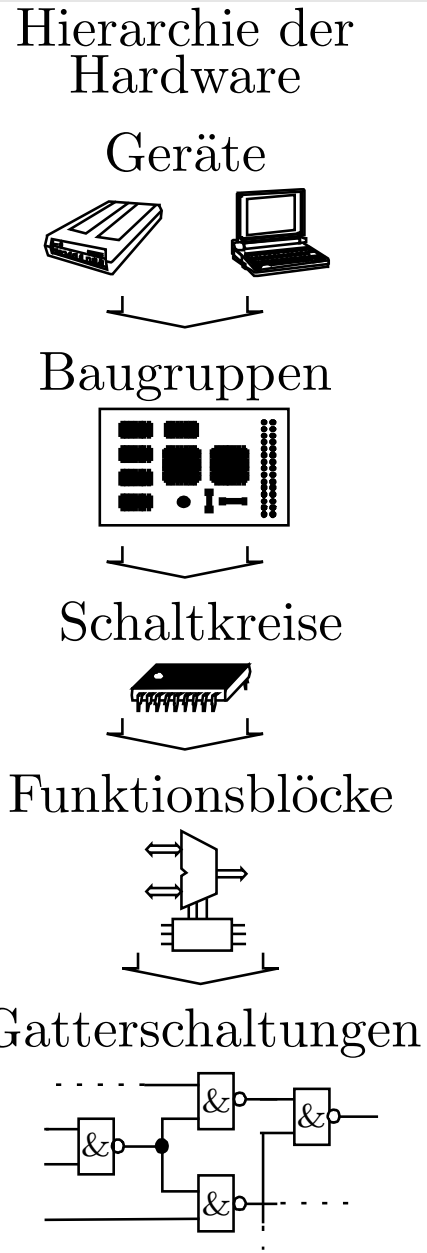
$$Z_{\mathrm{F}} \sim {t_{\mathrm{R}}}^{k+1}$$

Wenn Wissen über Fehlerumgehungsmöglichkeiten weitergegeben wird, z.B. über Foren, FAQ-Seiten, lernt die gesamte Nutzergemeinschaft. Summierung der $\#N$ vieler Nutzer.

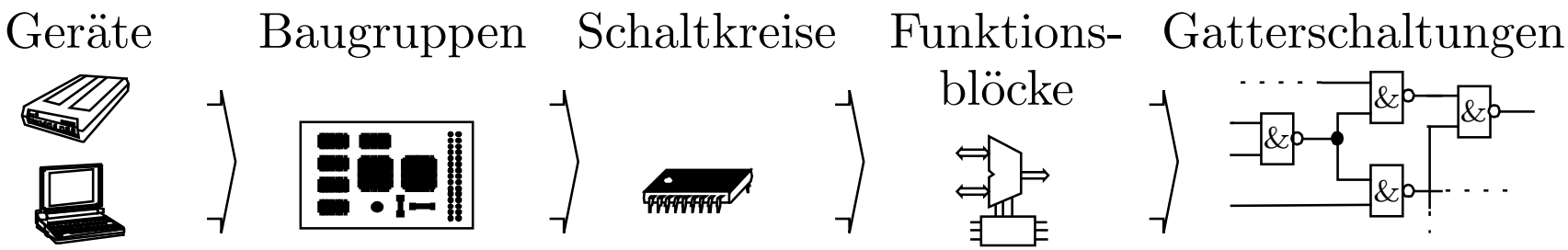
Modularer Test

IT-Systeme sind modular aufgebaut

- Rechner-Systeme bestehen aus Rechnern und Netzwerkkomponenten.
- Rechner, Netzwerkkomponenten, ... bestehen aus Hard- und Software.
- Software besteht aus Programm-bausteinen, diese sind aus Anweisungen zusammengesetzt, die ihrerseits mit Maschinenbefehlen nachgebildet werden.
- Maschinenbefehle sind Service-Leistungen der Hardware. Die Hardware besteht aus Funktionsbausteinen, diese meist aus Gattern und diese wiederum aus Transistoren.



Modularität ist wichtig für ...



- Entwurf: Aufspaltung in Teilaufgaben, Nachnutzung von Teilentwürfen, ...
- Test: Test der Komponenten vor Einfügung in das übergeordnete System.
- Reparatur: Austauschbarkeit von Komponenten.
- effektive Testsatzlänge von Zufallsttests ...

Effektive Testsatzlänge von Zufallstests

Für die Testauswahl interessieren nur die schlecht testbaren Teilbausteine, weil die Fehler in den gut testbaren Bausteinen auch ohne explizite Berücksichtigung bemerkt und beseitigt werden. Für schlecht testbare Teilbausteine gilt:

- nur ein kleiner Teil der Gesamt-SL nutzt sie als Teil-SL..
- Nur ein kleiner Teil der lokalen FF bildet sich auf eine Gesamt-FF ab.
- Die FF-Rate bei einem seperaten Test ist $c \gg 1$ mal größer als die Teil-FF-Rate des Bausteine für das Gesamtsystem.

Zur Erzielung derselben fehlerbezogenen Teilzuverlässigkeit des betrachteten Moduls sind bei einem ganzheiltichen Test c-mal so viele Tests wie für den seperaten Modultest erforderlich.

Fehlervermeidung

Fehler als FF des Entstehungsprozesses

Ein Entstehungsprozess ist auch ein Service

- mit Entwurfsvorgaben bzw. Material (-Eigenschaften) als Eingabe
- und Entwurfsergebnissen bzw. Produkten (oder ihren Eigenschaften als Ausgabe.

Er erbt damit auch die Kenngrößen zur Beschreibung der Verlässlichkeit:

- Verfügbarkeit, FF-Rate,
- Zuverlässigkeit und Sicherheit, ...

und die Maßnahmen zur Sicherung der Verlässlichkeit.

Im weiteren werden wir davon nur betrachten:

- FF-Rate als Entstehungsrate der Fehler und
- den Reifeprozesser zur Verringerung der FF-Rate und damit der Anzahl der entstehenden Fehler.

Fehlerentstehungsraten und -metriken

Ein Entstehungsprozess hat wie jeder Service eine FF-Rate, hier die Anzahl der entstehenden Fehler je SL, auch umrechenbar in entstehende Fehler je Zeit oder Produkt.

Für grobe Abschätzungen gibt es »entstehungsprozessunabhängige« Metriken für »entstehende Fehler je Systemgröße«, »entstehende Fehler je Reparaturschritt«, ...:

- Dokumentationen: mittlere Anzahl der Fehler pro Seite,
- Programmcode: mittlere Anzahl der Fehler pro 1000 NLOC (Netto Lines of Code) oder
- Schaltkreise: mittlere Fehleranzahl pro 10^6 Transistoren, ...

$$\text{Fehleranzahl} \approx \text{Systemgröße} \cdot \text{Kennwert}$$

Beispiel 5

30 Fehler / 1000 NLOC, Programm mit 2000 NLOC. Zu erwartende Anzahl der entstehenden Programmfehler: 60

Beispiel 6

1 Fehler je 10^6 Transistoren. Schaltkreis mit 10^5 Transistoren.
Zu erwartende Anzahl der entstehenden Fehler je Schaltkreis: 0,1.

Es gibt auch empirische Modelle, die eine überproportionale Zunahme der Fehleranzahl mit der Systemgröße postulieren. Für Software-Module wird z.B. unterstellt, dass die Fehleranzahl je NLOC ab 3 Quellcode-Seiten für einen Funktionsbaustein überproportional zunimmt, weil die Entwerfer die Übersicht verlieren.

Fehleranteil, Ausbeute

Fehleranteil und Ausbeute

Bei nicht reparierbaren Systemen und tauschbaren Komponenten interessiert nicht die Fehleranzahl, sondern nur, ob sie Fehler enthalten.

- Fehleranteil. Anteil der fehlerhaften Produkte $\#FP$ in einer Menge gleichartiger Produkte $\#P$:

  $$DL = \frac{\#FP}{\#P}$$

  Maßeinheiten dpu (defects per unit), dpm (defects per million):

  $$1\,\mathrm{dpu} = 10^6\,\mathrm{dpm}$$

  Für Fehleranzahl $\varphi \ll 1$ (fast nie mehr als ein Fehler je Produkt):

  $$DL = \varphi$$

- Ausbeute (Yield). Anteil der als gut befundenen gefertigten gleichartigen Objekte:

  $$Y = 1 - DL \cdot FC_{\mathrm{Obj}}$$

$$Y = 1 - DL \cdot FC_{\mathrm{Obj}}$$

Die Ausbeute hängt vom Anteil der erkennbaren fehlerhaften Objekte FC_{Obj} der Tests zur Abschätzung der Ausbeute ab. Ohne Test ist $FC_{\mathrm{Obj}} = 0$ und der Anteil der als gut befundenen Objekte $Y = 1$.

Beispiel 7

Ausbeute $Y = 95\%$, abgeschätzt mit einem Test, der $FC_{\mathrm{Obj}} = 50\%$ der fehlerhaften Objekte erkennt. Fehleranteil der Objekte:

$$DL = \frac{1 - Y}{FC_{\mathrm{Obj}}} = 10\%$$

Beim Aussortieren der erkannten fehlerhaften Objekte verringern sich die Anzahl der fehlerhaften Objekte in Zähler und die Anzahl aller Objekte im Nenner jeweils um die Anzahl der erkannten fehlerhaften Objekte $\#Obj \cdot DL \cdot FC_{\mathrm{Obj}}$:

$$DL_{\mathrm{T}} = \frac{\#Obj \cdot DL - \#Obj \cdot DL \cdot FC_{\mathrm{Obj}}}{\#Obj - \#Obj \cdot DL \cdot FC_{\mathrm{Obj}}} = \frac{DL \cdot (1 - FC_{\mathrm{Obj}})}{1 - DL \cdot FC_{\mathrm{Obj}}}$$

$$DL_{\mathrm{T}} = \frac{\#Obj \cdot DL - \#Obj \cdot DL \cdot FC_{\mathrm{Obj}}}{\#Obj - \#Obj \cdot DL \cdot FC_{\mathrm{Obj}}} = \frac{DL \cdot (1 - FC_{\mathrm{Obj}})}{1 - DL \cdot FC_{\mathrm{Obj}}}$$

($\#Obj$ – Anzahl aller Objekte; FC_{Obj} – Anteil der erkannten fehlerhaften Objekte).

Beispiel 8

Schaltkreisausbeute $Y = 80\%$, Fehleranteil nach Test und Fehlerbeseitigung $DL_{\mathrm{T}} = 1000\,\mathrm{dpm}$. Gesucht FC_{Obj}.
Eine Verringerung von DL von ≈ 1 auf 10^{-3} verlangt $FC_{\mathrm{Obj}} \approx 1$:

$$DL = \frac{1 - Y}{FC_{\mathrm{Obj}}} = 20\%$$

$$DL_{\mathrm{T}} = \frac{DL \cdot (1 - FC_{\mathrm{Obj}})}{1 - DL \cdot FC_{\mathrm{Obj}}} \approx \frac{DL \cdot (1 - FC_{\mathrm{Obj}})}{1 - DL}$$

$$FC_{\mathrm{Obj}} \approx 1 - \frac{DL_{\mathrm{T}} \cdot (1 - DL)}{DL} = 1 - \frac{10^{-3} \cdot (1 - 20\%)}{20\%} = 99{,}6\%$$

Das ist auch die typische Größenordnung der Fehlerüberdeckung von Schaltkreistests.

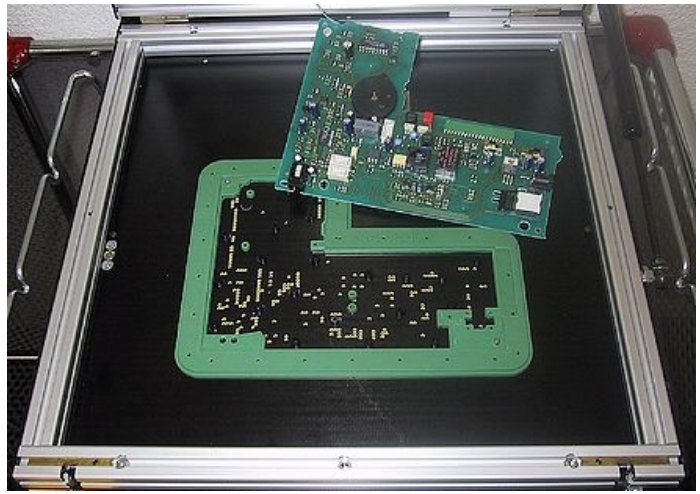
Fehleranzahl komplexer Systeme

Komplexe Systeme werden oft aus vielen getesteten Teilsystemen mit je einem kleinen Fehleranteil $DL_{\mathrm{TS}.i} \ll 1$ zusammengesetzt. Der übergeordnete Test kontrolliert nur noch auf Verbindungsfehler, die beim Zusammensetzen entstehen, aber fast nicht mehr auf Fehler innerhalb der Teilsysteme. Zu erwartende Fehleranzahl des getesteten Gesamtsystems:

$$\varphi_{\mathrm{Sys.T}} \approx \varphi_{\mathrm{Verb}} \cdot (1 - FC_{\mathrm{Verb}}) + \sum_{i=1}^{\#TS} DL_{\mathrm{TS}.i}$$

(φ_{Verb} – Anzahl der Verbindungsfehler; $\#TS$ – Anzahl der Teilsysteme; $DL_{\mathrm{TS}.i}$ Fehleranteil der getesteten Teilsysteme).

Beispiel Baugruppentest



Baugruppen, besteht aus getesteten Komponenten, werden in der Regel nach der Fertigung auf ein Nadelbett gespannt und auf Verbingungs- und Bestückungsfehler getestet.
Fehlerüberdeckung für Verbindungsfehler (Kurzschüsse und Unterbrechungen) und Bestückungsfehler praktisch 100%.
Fehlerüberdeckung für die vom Bauteiltest nicht erkannten Bauteilfehler praktisch 0%. Fehleranteil Baugruppe:

$$\varphi_{\mathrm{BG.T}} \approx \sum_{i=1}^{\#BT} DL_{\mathrm{T}i}$$

($\#BT$ – Anzahl der Bauteile; $DL_{\mathrm{T}i}$ – Fehleranteil Bauteil i). Für $DL_{\mathrm{BG.T}} \ll 1$:

$$DL_{\mathrm{BG.T}} = \varphi_{\mathrm{BG.T}} = \sum_{i=1}^{\#BT} DL_{\mathrm{T}i}$$

Beispiel 9

Anzahl und Fehleranteil der Bauteile einer Baugruppe:

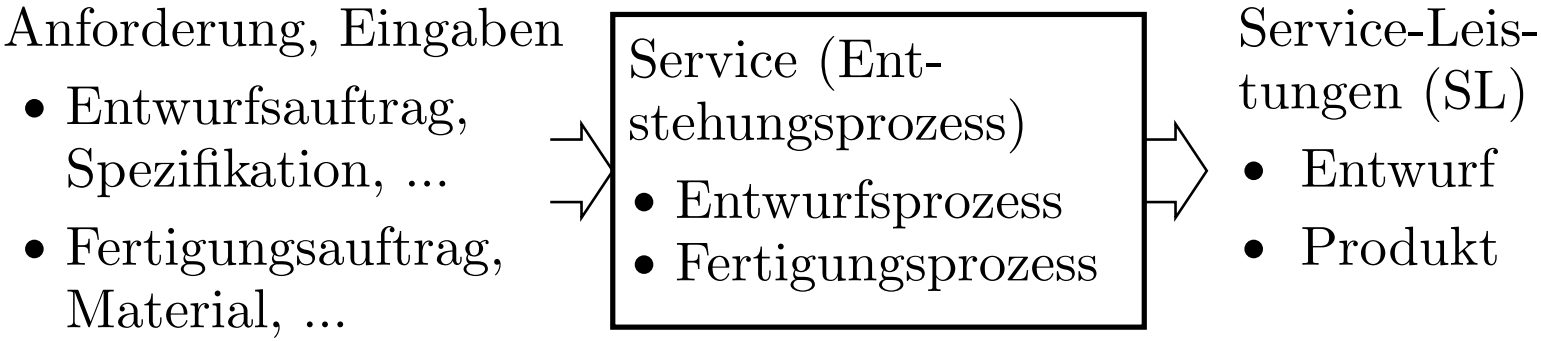
Typ	Anzahl	DL_{BT}
Leiterplatte	1	20 dpm
Schaltkreise	20	200 dpm
diskrete Bauteile	35	10 dpm
Lötstellen	560	1 dpm

$$\begin{aligned} DL_{\mathrm{BG.T}} &= 10\,\mathrm{dpm} + 20 \cdot 200\,\mathrm{dpm} + 35 \cdot 10\,\mathrm{dpm} + 560 \cdot 1\,\mathrm{dpm} \\ &= 5000\,\mathrm{dpm} = 0{,}005\,\mathrm{dpu} \end{aligned}$$

(dpm – defects per million) Etwa jedes 200ste Gerät enthält ein nicht erkanntes defektes Bauteil.

Determinismus und Zufall

Fehlerentstehung

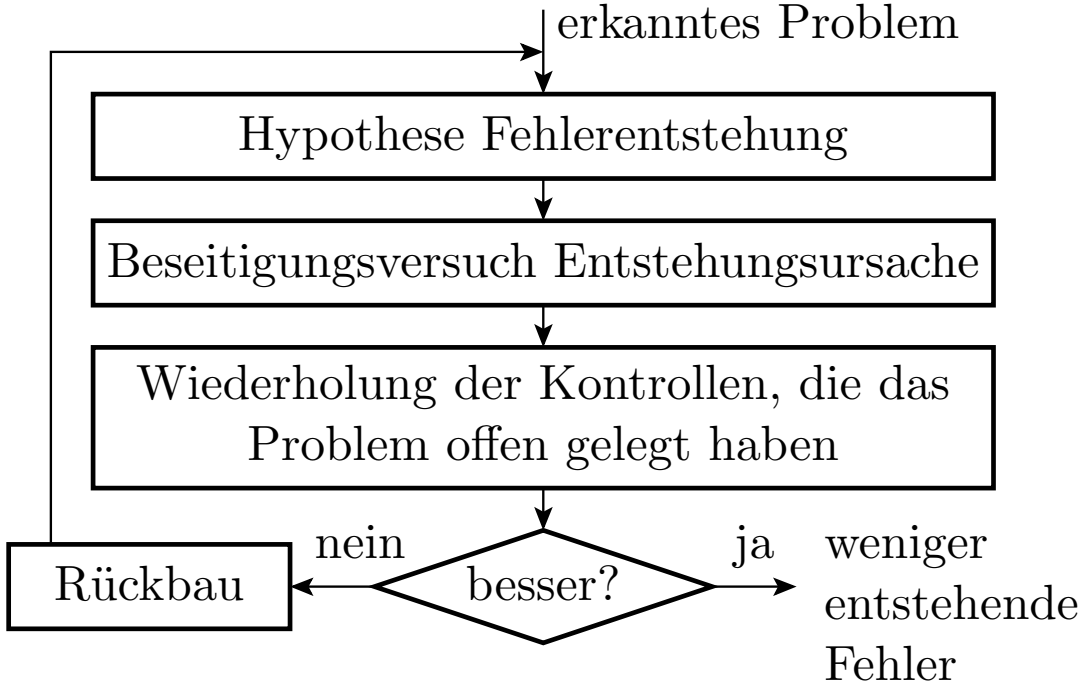


Ursachen für die Fehlerentstehung:

- Fehler: deterministische Ursache-Wirkungsbeziehung
  - beseitigbare Ursachen,
  - Erfolgskontroll durch Testwiederholung, ...
- Störungen: zufällige Ursache-Wirkungsbeziehung
  - FF durch Wiederholung beseitigbar,
  - Erfolgskontrolle Beseitigung schwierig, ...
- Ausfälle: bei Service-Nutzung entstehende Fehler, ...

Fehlervermeidung erfolgt durch Beseitigung von Fehlern in Entstehungsprozessen und durch Minderung der Störanfälligkeit.

Fehlervermeidung ist experimentelle Reparatur



Fehlervermeidung ist eine Reifeprozess für einen Entstehungsprozess mit experimenteller Reparatur zur Problembeseitigung. Iteration aus:

- Problemerkennung, Lokalisierung, versuchsweise Beseitigung,
- Erfolgkontrolle durch Wiederholung der Prozessabläufe, die das Problem erkannt haben.

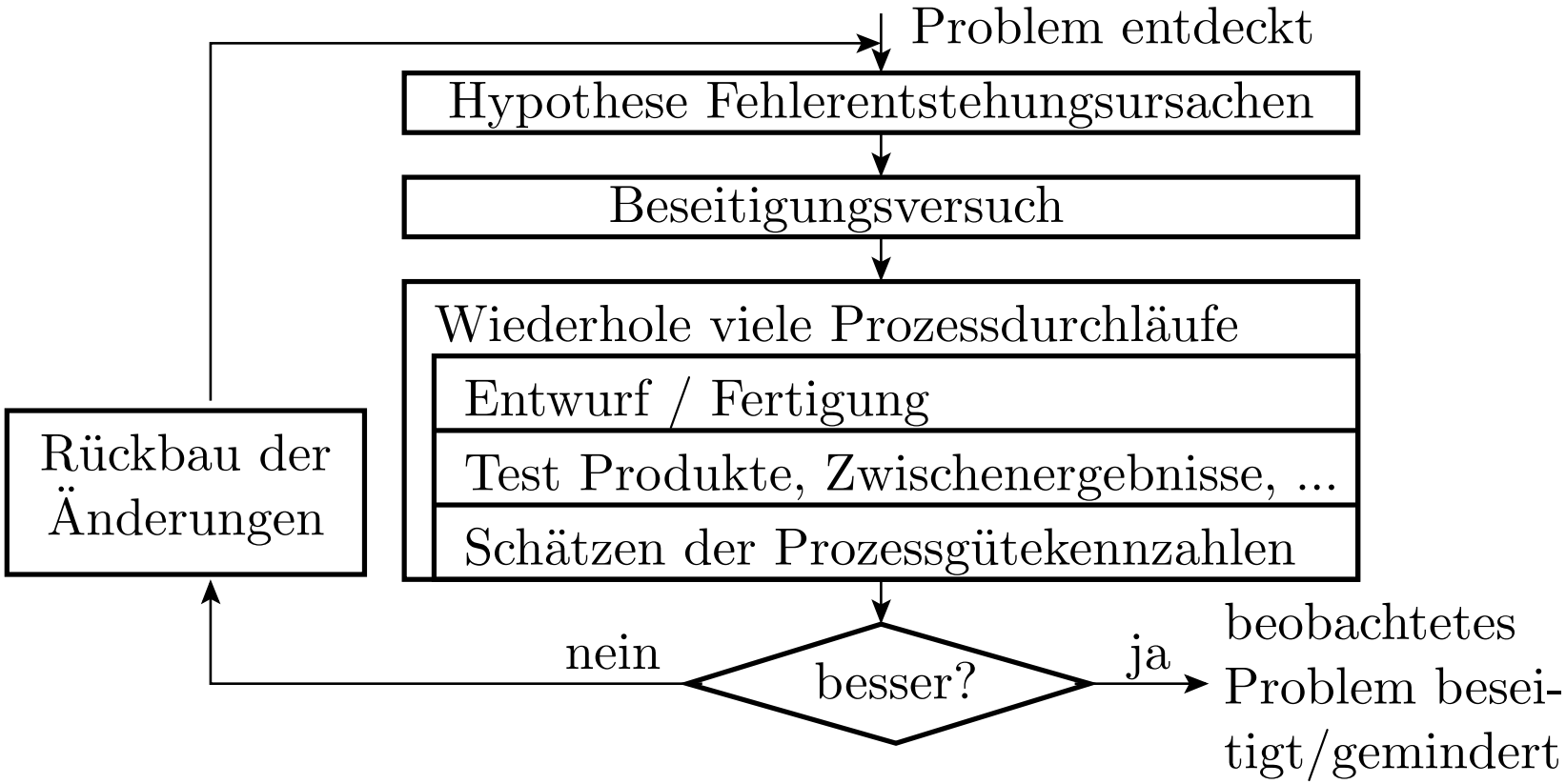
Experimentelle Reparatur und Determinismus

Determinismus bedeutet, dass das fehlerfreie System für denselben Entwurfs- oder Fertigungsauftrag (nach derselben Sezifikation, mit demselben Material, ...) immer dieselben Ausgaben (dasselbe Entwurfsergebnis, ein identisches Produkt, ...) liefert.

Für Fehler in deterministischen Prozessen lassen sich in der Regel Prozesabläufe mit Soll/Ist-Kontrollen an Zwischenergebnissen und Endprodukte finden, die eindeutige ja/nein-Aussage über das Vorhandensein/Beseitigung von Fehlern liefern.

Für nicht deterministische Prozesse, Fehler mit nicht deterministischer Wirkung und Prozessstörungen verlangt die Kontrolle des Erfolgs eines Problembeseitigungsversuchs in der Regel eine statistisch signifikannte Stichprobe von Prozessdurchläufen zur Bestimmung von Prozessgütekennzahlen und Entscheidungen mit Irrtumswahrscheinlichkeiten.

... nicht deterministische Prozesse, Fehlerwirkungen, Störungen:

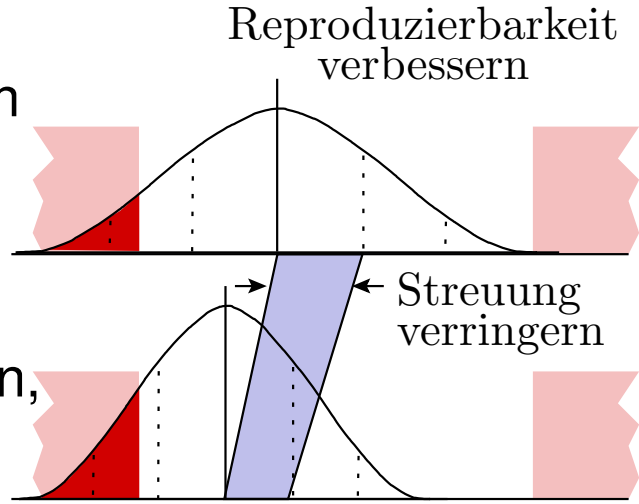


Die Beseitigung eines einzelnen Problems verlangt um Zehnerpotenzen mehr Prozessdurchläufe und Kontrollen, schlechtere Erfolgschancen, viel höheres Risko, bei Beseitigungsversuchen neue Fehler einzubauen, die nicht durch Rückbau beseitigt werden, ...

Prozesszentrierung und -verbesserung

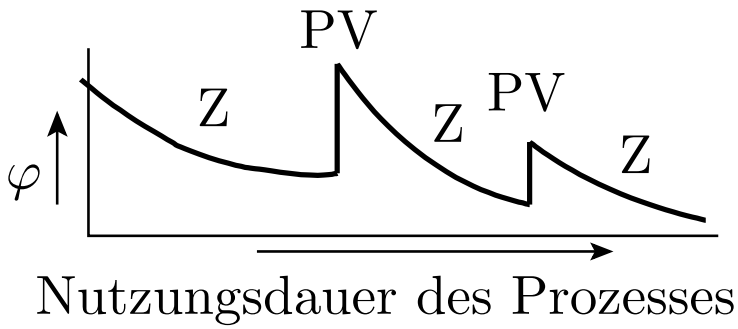
Bei der mechanischen Fertigung haben die Zielparameter, z.B. bei einer Bohrung Durchmesser und Tiefe, eine Verteilung und einen Toleranzbereich. Entstehungshäufigkeit eines Parameterfehlers ist etwa die Wahrscheinlichkeit, Parameter außerhalb Toleranzbereich:

- Prozesszentrierung: Verschiebung der Verteilung mit Hilfe von Einstellgrößen in die Mitte des Toleranzbereichs.
- Prozessverbesserung: Verringerung der Streuung durch technologische Neuerungen neue Maschinen, Verfahren, ...



Bei der Prozessverbesserungen geht die Zentrierung verloren. Sprunghafte Zunahme der Fehlerenstehungsrate.

Sägezahnverlauf der Fehleranzahl



Z	Prozesszentrierung
PV	Prozessverbesserung mit Verlust der Zentrierung
φ	Fehleranzahl in den entstehenden Produkten

Technologische Verbesserungen (neue Maschienen, Programme, Technologien, ...) erfolgen in größeren zeitlichen Schritte (Monate, Jahre) und haben das Potential, die zu erwartende Fehleranzahl zu verringern.

- Bei jeder technologischen Umstellung geht die Zentrierung verloren und die Fehleranzahl steigt sprunghaft.
- Die potentiell geringere Fehleranzahl wird erst durch erneute Zentrierung nach einer gewissen Nutzungsdauer erreicht.
- Abnehmender Sägzahnverlauf der zu erwartenden Fehleranzahl.

Auch bei anderen Fertigungsprozessen und Entwurfsprozessen

- gibt es in größeren Zeitschritten technologische Neuerungen, die die erreichbare Fehlerenstehungsrate durch geringere Störanfälligkeit, höhere Reproduzierbarkeit, ... absenken. Bei Neuerungen entstehen jedoch neue Prozessfehler, die beobachtbare Fehleranzahl bzw. den Fehleranteil der Produkte sprunghaft erhöhen.
- dazwischen eine kontinuierliche Suche und Beseitigung der hinzugekommenen Fehlerentstehungsursachen, beginnend mit denen, die die meisten Fehler verursachen. Wirkung auf den Prozess ähnlich wie Zentrierung.

Fakt 10

Am qualitativ hochwertigsten sind tendentiell die Produkte, die kurz vor technologischen Neuerungen entstehen. Maxima der Prozesszuverlässigkeit. (*Am besten versuchen, immer solche Produkte zu bekommen.*)

Eine Schattenseite von Innovationen

Technologische Reifeprozesse sind heute bei jeder Art von Produkten und Service-Leistungen zu beobachten:

- Verbesserte Wiederholgenauigkeit der Abläufe,
- verbesserte vorhersagbare Material- und Produkteigenschaften,
- weniger entstehende Fehler, Ausbeute ↑, Kosten ↓, ...

Schattenseite:

- Neuerungen führen fasts zwangsläufig zu »neuen Kinderkrankheiten«, die erst nach einer gewissen Reifezeit beseitigt sind.
- Mehr entstehende Fehler bedeutet nicht nur schlechtere Ausbeute und mehr Kosten, sondern auch auch mehr Fehler in eingesetzten Systemen, mehr Frühausfälle, ...

Linux unterscheidet z.B. in seiner Versionsverwaltung:

- »Innovative« Beta-Versionen mit vielen Kinderkrankheiten, ...
- und einsatztaugliche (zuverlässige) Stable-Versionen.

Projekte, Vorgehensmodelle

Der Technologiegedanke

Technologie: Lehre von reproduzierbaren Abläufen zur Erzeugung von Produkten[9].

Technologiegedanke

Ein technologischer Prozess ist so zu gestalten, dass, wenn er unter gleichen Bedingungen wiederholt wird, gleiche Produkte mit (nahezu) gleichen Eigenschaften entstehen.

Die technologische Entwicklung hin zur

- automatisierten menschenfreien Fertigung und
- rechnergestützten / automatisierten Entwurfsprozessen

dient nicht nur zur Kostensenkungung, sondern ist auch wesentliche Grundlage für die Fehlervermeidung.

[9] Der Begriff »Technologie« wurde erstmalig von dem Göttinger Professor Johann Beckmann (1739-1811) in seinem Lehrbuch »Grundsätze der teutschen Landwirthschaft« verwendet. Heute interdisziplinäres Gebiet.

Übertragung des Technologiegedanken auf Projekte

Technologien reifen dadurch, dass die Abläufe sehr oft durchlaufen werden, um viele Fehler zu erkennen und den Beseitigungserfolg zu kontrollieren.

Wie verhält es sich mit Projekten:

- Manuelle kreative Teile der Entwurfsprozesse[10] und
- Fertigung von Prototypen, Demonstratoren, ... ?

Ein Projekt ist ein zielgerichtetes, einmaliges Vorhaben, das aus einem Satz von abgestimmten, gelenkten Tätigkeiten besteht. ...

Projekten fehlt aus Sicht der Fehlervermeidung die Reproduzierbarkeit und die häufige Wiederholung.

Schließt das eine Fehlervermeidung aus?

[10] Hier inbesondere der Software- und Hardware-Entwurf.

Vorgehensmodelle

Vereinheitlichung des Vorgehens für große Klassen von Projekten

- zur Aufwandsminimierung, besseren Vorhersagbarkeit und
- zur Fehlervermeidung durch »Lernen aus Fehlern«.

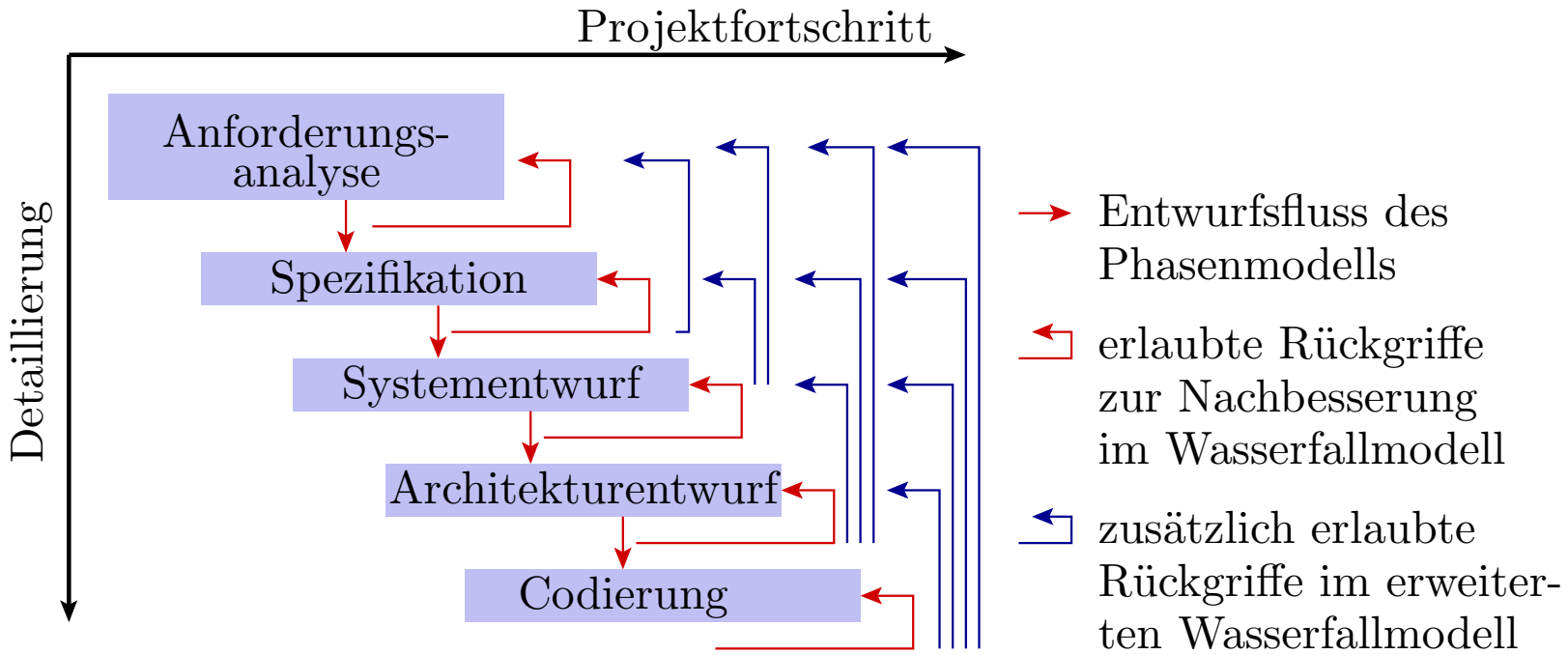
Typische Vorgehensmodelle für den Entwurf und die Fertigung von IT-Komponenten umfassen:

- Unterteilung in Schritte und Phasen,
- Referenzabläufe,
- Definition von Zwischen- und Endkontrollen, ...

Die klassischen Vorgehensmodelle für den Software-Entwurf sind Stufenmodelle. Sie unterteilen Entstehungsprozesse in Phasen:

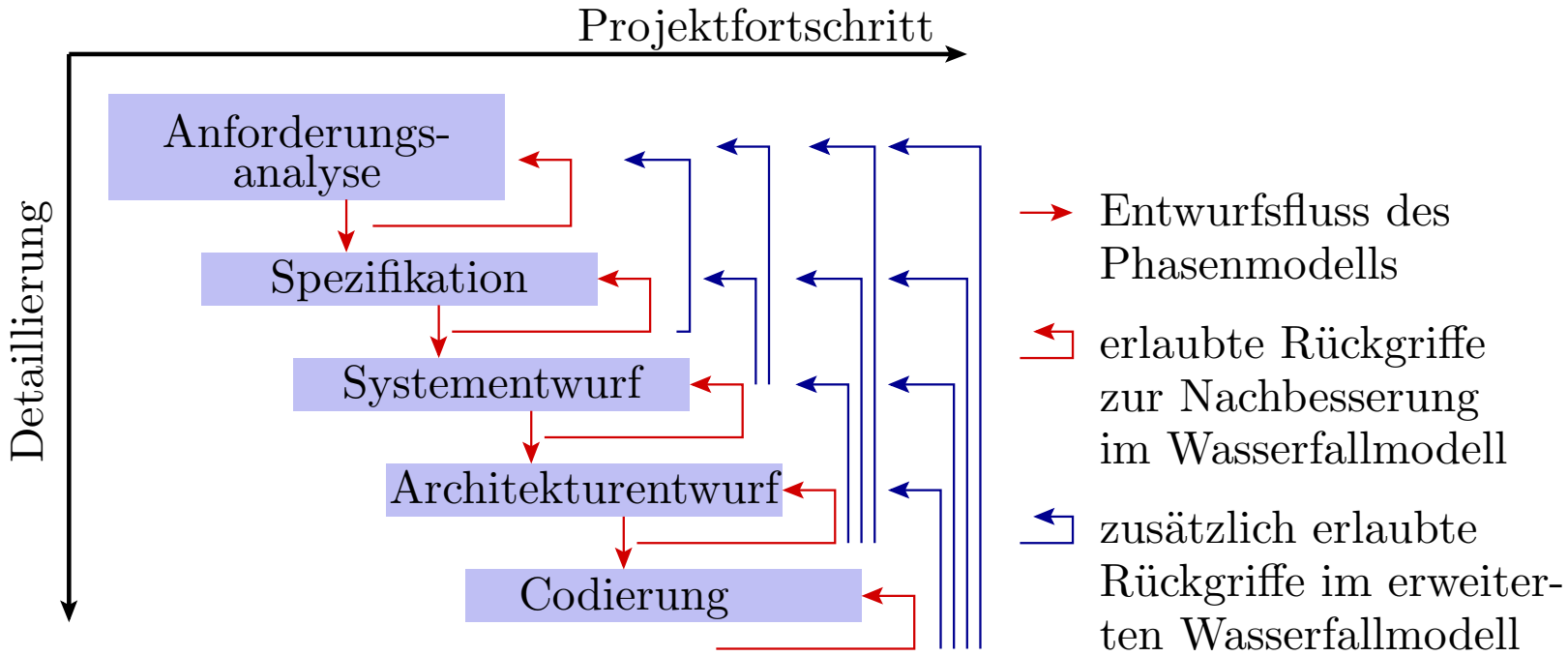
- Anforderungsanalyse,
- Spezifikation der Ziele,
- Architekturentwurf, Codierung, Test, ...

Stufenmodelle



Stufenmodelle variieren:

- in den Abgrenzungen der Entwurfsphasen,
- Dokumentation und Kontrolle bei Phasenübergängen,
- dem Vorgehen bei Rückgriffen (rückwirkende Änderungen an den Ergebnissen bereits abgeschlossener Phasen). ...



Gestaltbare Einflussfaktoren auf Qualität und Kosten:

- Arbeitsorganisation der Phasen,
- geforderte Tests, Dokumentation, ... bei Phasenübergängen,
- Regeln für Rückgriffe zur Nachbesserung, ...

Fehlervermeidung bei Projektarbeit ist die empirische Suche nach einem guten Vorgehensmodell und seine Einhaltung.

Bewertung von Vorgehensmodellen

Jede Art der Fehlervermeidung benötigt eine Erfolgskontrolle:

Daraus resultierende Frage

An welchen mess- oder abschätzbaren Parametern ist eine Verbesserung eines Vorgehensmodells erkennbar?

Diese Parameter müssen zwischen unterschiedlichen konkreten Projekten eines Vorgehensmodells vergleichbar sein:

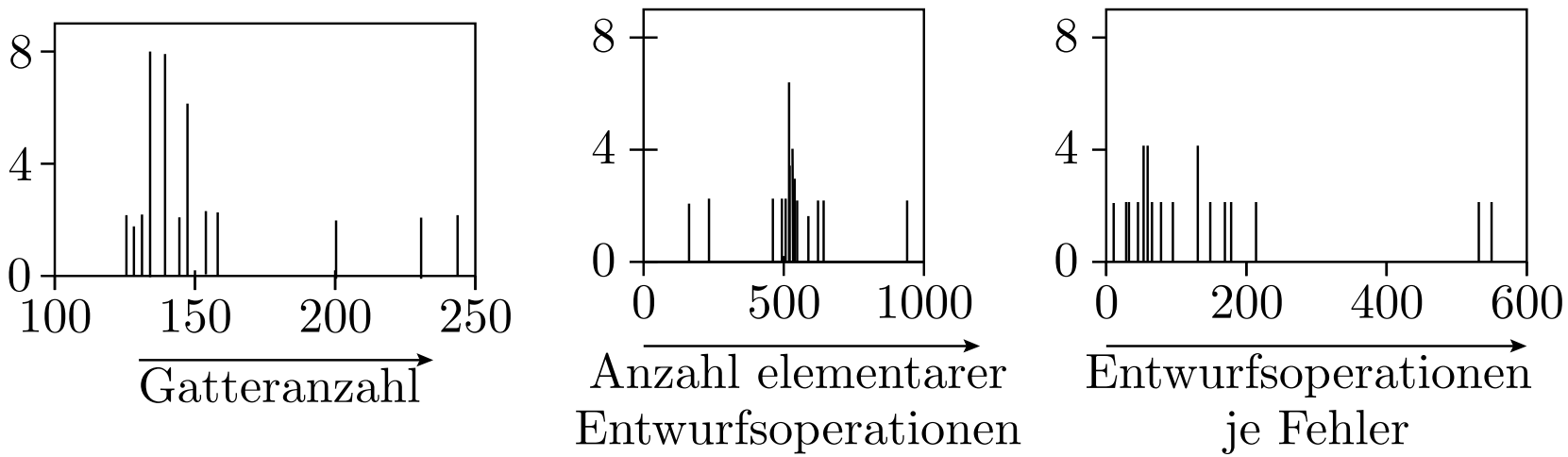
- Projektdauer, Projektkosten,
- Arbeitsschritte je entstehender Fehler, Umfrageergebnisse, ...

Erwartungswerte, Streuungen, Skalierbarkeit auf Projektgröße, Schwierigkeit, ...

Signifikante Aussagen über Vorgehensmodelle verlangen die Beobachtung tausender Projekte mit vergleichbarem Vorgehen.

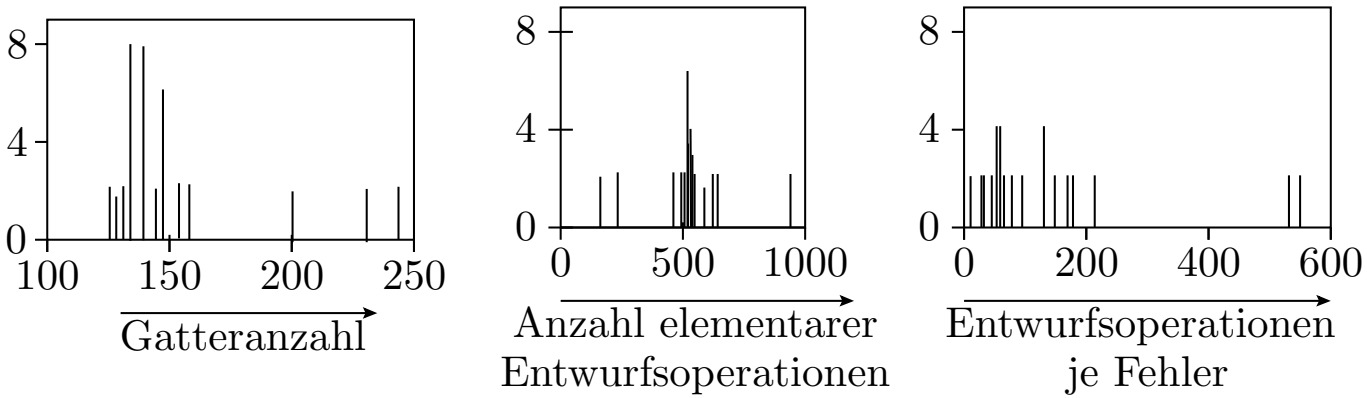
Ein Experiment [11]

Eine Gruppe von 72 Studenten sollte aus der Beschreibung eines PLAs eine Gatterschaltung zu entwickeln und diese über eine GUI in ein CAD-System einzugeben. Für jeden Entwurf wurden die elementaren Entwurfsoperationen, die Gatteranzahl und die Entwurfsfehler gezählt. Als elementare Entwurfsoperationen galten das Anordnen eines Gatters auf dem Bildschirm, das Zeichnen einer Verbindung, ...



[11] Aas, J. E., Sundsbo, I.: Harnessing the Human Factor for Design Quality, IEEE Circuits and Devices Magazine, 3/1995, S. 24-28

Welche Rückschlüsse erlaubt das Experiment?



Angenommen, der Versuch wird genauso an anderen Hochschulen wiederholt:

- Auch hier dieselben Kenngrößen je Student bestimmen.
- Verteilung, Erwartungswert und Varianz vergleichen.
- Unterschiede statistisch signifikant?

Aus den Vergleichergenissen ließe sich schlussfolgern, ob und an welcher Hochschule Studierende für diese Aufgabe besser ausgebildet werden. (So etwas hat sicher noch niemand probiert.)

Qualität und Kreativität

Qualität und Kreativität

Qualität verlangt Fehlervermeidung. Fehlervermeidung verlangt Reproduzierbarkeit:

- eine hohe Wiederholrate gleicher oder ähnlicher Tätigkeiten,
- einzuhaltende Arbeitsabläufe,
- Protokollierung aller Unregelmäßigkeiten und Probleme, ...

Kreativität verlangt »Einzigartigkeit«:

- Einbringen neuer Konzepte,
- Ausprobieren neuer Lösungswege,
- flexible Anpassung an sich ändernde Anforderungen.

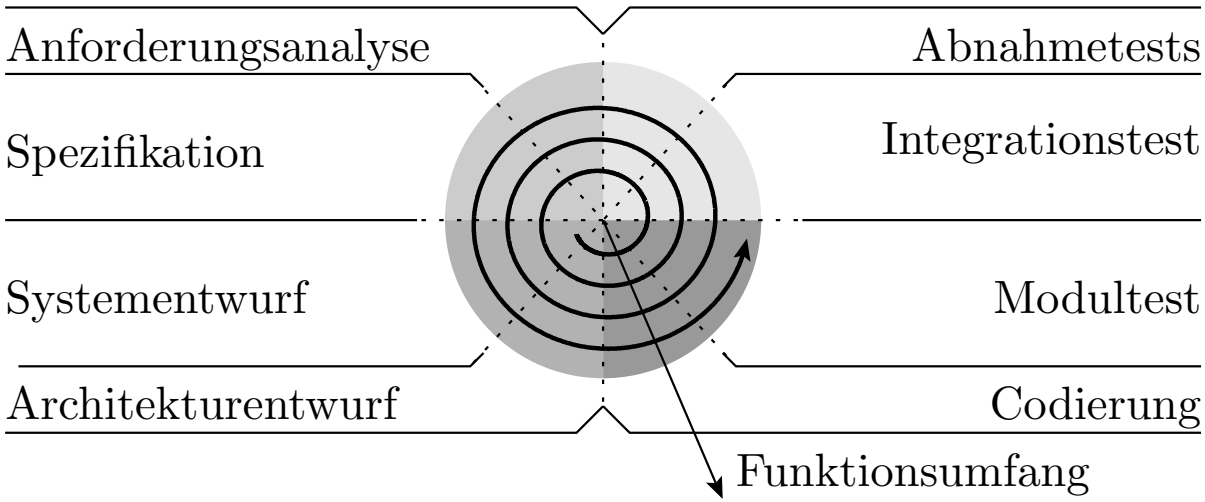
Schlussfolgerung

Qualität und Kreativität haben entgegengesetzte Anforderungen an den Gestaltungsspielraum von Arbeitsabläufen.

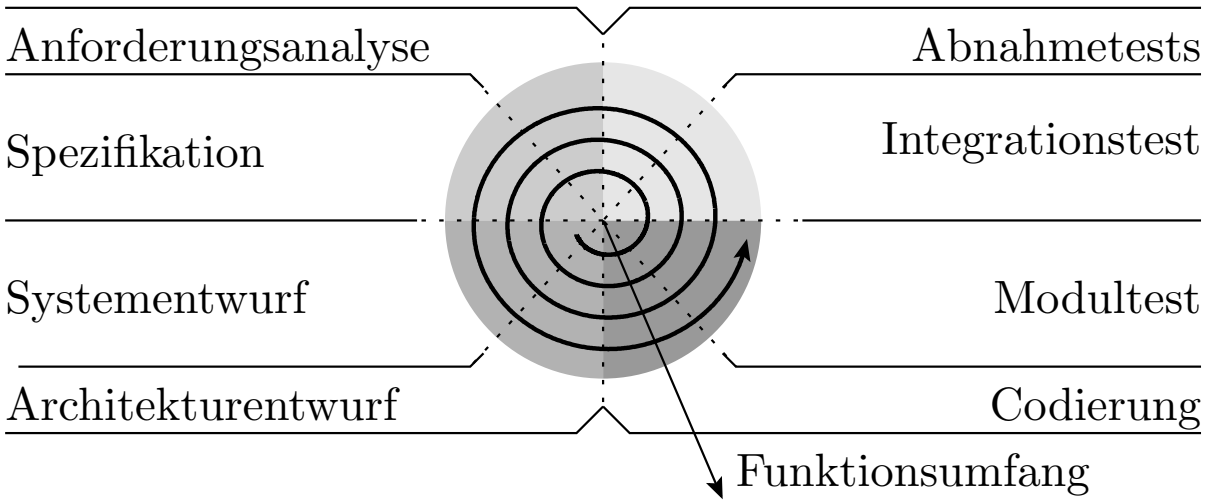
IT-Entwurf verlangt Qualität und Kreativität.

Spiralmodell als Beispiel evolutionärer Modelle

Evolutionäre Vorgehensmodelle versuchen einen Rahmen für Projekte zu bieten, bei denen sich Kundenwünsche, Ziele, Vorgehen, ... mit dem Projekt weiterentwickeln. Weniger starre Abläufe. Mehr kreativer Gestaltungsspielraum. Beispiel Spiralmodell:



- Aufteilung einer Entwicklung auf ein mehrmaliges Durchlaufen eines Stufenmodells.



Aufteilung einer Entwicklung auf ein mehrmaliges Durchlaufen eines Stufenmodells.

- Durchlauf 1: Spezifikation von Grundanforderungen, Entwurf, Codierung, Test, ..., Abnahme und Einsatz.
- Durchlauf 2 bis n: Ideensammlung und Auswahl gewünschter Zusatzanforderungen und Änderungen. Entwurf bis Einsatz.

Innerhalb der Iteration ist der Ablauf festgeschrieben. Kreativer Freiraum in Form einer Ideensammlung für die nächste Version.

Querverbindungen zum akademischen Alltag

Auch für die Gestaltung von Lernprozesses werden Vorgehensmodelle genutzt. Der Bolonia-Prozess (Bachelor-Master) strebt danach, Referenzabläufe zu etablieren.

Dahinter verbirgt sich die Hoffnung, dass sich mit dem Technologiegedanken im Bildungssystem ähnlich spektakuläre Fortschritte wie in Naturwissenschaft und Technik erzielen lassen:

- Vereinheitlichung der Abläufe.
- Verbesserung der Vorhersagbarkeit und Vergleichbarkeit der Bildungsergebnisse und Kosten.
- Übernahme der »Vorgehen« aus Bildungseinrichtungen mit besseren Ergebnissen von Bildungseinrichtungen mit schlechteren Ergebnissen.

Fehlervermeidung beschränkt die Kreativität. Sind Kreativitätsbeschränkungen für Universitäten akzeptabel?

Ein Abstecher zu Lernrozessen

In der Schule und beim Erlernen praktischer Tätigkeiten werden zum erheblichen Teil Vorgehensmodelle vermittelt und trainiert:

- Rechnen, Schreiben, Handwerkern, Programmieren, ...
- Bewertung in Arbeitsmenge pro Fehler und pro Zeit.

Lernphasen:

1. Wissenvermittlung: anlesen, erklärt bekommen, ...
2. Training, bis Ergebnisse vorhersagbar.
3. Professionalisierung: Prozessüberwachung; Beseitigung von Vorgehensfehlern und -schwachstellen.

An Universitäten:

- Phase 1: Vorlesung, Seminare, Selbststudium, ...
- Phase 2: Übung, Klausurvorbereitung[12], Praktika.
- Phase 3: Aus Zeitgründen erst in der Berufspraxis für den eigenen eingeschränkten Tätigkeitsbereich.

[12] Auch Bewertung in Arbeitsmenge pro Zeit und Fehler pro Arbeitsmenge.

Querverbindung Drittmittelprojekte

- Die Professionalisierungsphase durchlaufen erst die Absolventen in der Praxis.
- Akademiker und Studenten sind selten für »fehlerarme Arbeitsabläufe« qualifiziert.
- In industriellen Software-Projekten entstehen durch Akademiker tendenziell mehr Fehler je Aufgabengröße.
- Die Kosten für die Fehlerbeseitigung trägt der Industriepartner.
- Deshalb rechnet es sich für die Industrie nicht, Hochschulen und Studenten in ihr Tagesgeschäft einzubinden.
- Industrielle Studenten-Projekte dienen der Ausbildung.
- Drittmittelforschung ist wertvoll für den Knowhow-Transfer, Literaturstudien, Demostratoren, ... aber im IT-Bereich ungeeignet für die Einbindung in die Produktentwicklung.

Fehlerkultur

Art und Weise, wie Gesellschaften, Kulturen und soziale Systeme mit Fehlern, Fehlerrisiken und Fehlerfolgen umgehen.

Positive Sichtweisen:

- Pädagogik: positives Klima, in dem die Angst vorm Fehlermachen abgebaut wird und Lernen aus Fehlern stattfindet.
- Qualitätsmanagement: Minimierung der Fehlerkosten für Ausschuss, Nacharbeiten, Reklamationsbearbeitung, Wiedergutmachungskosten, Imageschäden, ...
- Innovationsmanager: Streben nach Neuerungen. Fehler nicht nur unvermeidbare Begleiterscheinung, sondern auch Chance / produktives Potential.

Normale Sichtweise in unserer westlich Kultur:

Ein Geschäftsmann, der nicht lügt, ist kein guter Geschäftsmann. Ein Geschäftsmann, der sich beim Lügen erwischen lässt, ist ein schlechter Geschäftsmann.