Test und Verlässlichkeit F1: Gefährdungen und Gegenmaßnahmen

Prof. G. Kemnitz

Institut für Informatik, TU Clausthal (TV_F1)
November 6, 2022

Organisation

Web-Seite Vorlesung: http://techwww.in.tu-clausthal.de/TestVerl_2022

- Foliensätze, Handouts, Hausübungen, Videoaufzeichnungen
- Abgabe der Hausübungen per Mail an ha-tv@in.tu-clausthal.de als pdf. Abgabetermine siehe Web-Seite.
- Hausübungen werden bewertet und zurückgegeben. Zusätzliche Veröffentlichung der Punkteanzahl auf der Webseite.
- Prüfungszulassung 50% der erzielbaren Hausübungspunkte. Für größere Punkteanzahl bis zu 2 Bonuspunkten für die Prüfung.
- Fragen und Kommentare an: gkemnitz@in.tu-clausthal.de

Prüfung

- Prüfung ab 10 Teilnehmer schriftlich.
- 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 siehe Web-Seite.

Inhalt Foliensatz TV_F1

Vorlesung	1	2	3	4
Folien	6 bis 41	42 bis 91	93 bis 145	147 bis 181

Einführung

Vorlesung 1: Geplante Themen

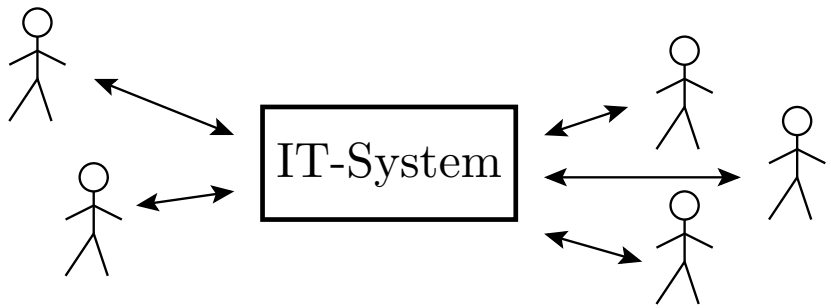
Abschn. 1: Einführung:

- Vertrauen in und Verlässlichkeit von IT-Systemen.
- Gefährdungen und Gegenmassnahmen.
- Was kostet Verlässlichkeit und was kostet fehlende Verlässichkeit?
- Fehlerkultur und die Rolle der Tests für die Sicherung der Verlässlichkeit.

Abschn. 2: Verlässlichkeit:

- Service-Modell, um erbrachte Leistungen und Fehlfunktionen zählbar zu machen.
- Verfügbarkeit: Zeitanteil in dem das System verfügbar ist,
- Zuverlässigkeit: zur erwartende Anzahl der erbarcter Service-Leistungen je Fehlfunktion.
- Sicherheit: Teilzuverlässigkeit in Bezug auf sicherheitsgefährdende Fehlfunktionen.

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 voraus.

Verlässlichkeit

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

Subjektive Einflussfaktoren auf die Wahrnehmung der Verlässlichkeit:

- Lebenserfahrungen insbesondere aus der Kindheit,
- Katastrophen oder langsam Veränderungen,
- Persönlichkeitstyp (Optimist, Pessimist, konservativ, Spieler), ...

Objektivierung durch Zählen positiver und negativer Erfahrungen und deskriptive Attribute:

- wofür verlässlich:
  - Dozent verlässlich, dass pünktlich,
  - Student verlässlich, dass HA abgegeben werden, ...
- warum verlässlich:
  - Arzt verlässlich, weil abgeschlossenes Medizinstudium,
  - Auto verlässlich, weil technische Zulassung und gültiger TÜV.

Offenbar sind Überwachung und bestandene Kontrollen wichtig für die Verlässlichkeit, aber auch die Fehlerkultur ...

Fehlerkultur

Art und Weise, wie Gesellschaften, Kulturen und soziale Systeme mit Fehlern und deren Folgen umgehen.

Negative Sichtweise: Fehler verstecken, wegreden, ...

Positive Sichtweisen: Aus Fehlern lernen, Fehler beseitigen. ...

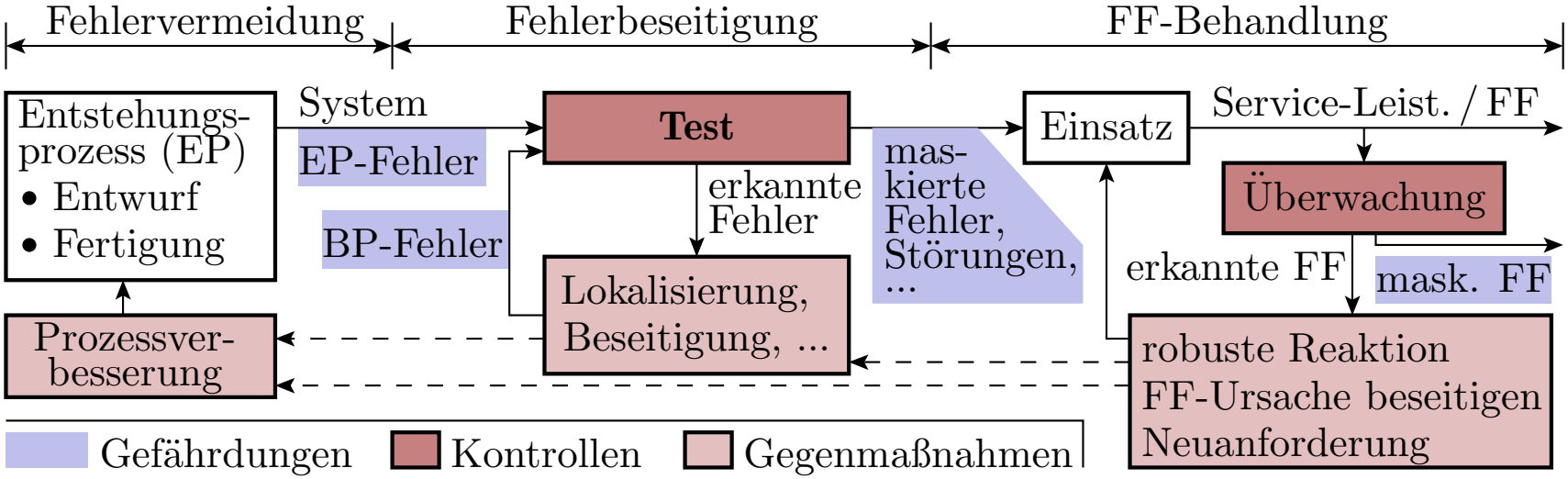
- Pädagogik: positives Klima für Lernen aus Fehlern.
- Qualitätsmanagement: Minimierung der Fehlerkosten.
- Innovationsmanager: Streben nach Neuerungen. Fehler als Chance / produktives Potential.

Die Vorlesung unterstellt folgende idealisierte Fehlerkultur:

- Alle erkannten Probleme werden beseitigt.
- Beseitigungserfolg wird durch Testwiederholung kontrollieren.

Ungeeignet für die Kostenoptimierung im IT-Bereich. Auch für den Umgang mit Freunden, Vorgesetzten und Partnern, ist eine weniger radikale Fehlerkultur empfehlenswert.

Gefährdungen und Gegenmaßnahmen



Gefährdungen für die Verlässlichkeit von IT-Systemen:

- Fehlfunktionen (FF) und deren Ursachen und Folgen.
- Ursachen für FF sind Fehler, Störungen, Ausfälle.
- Ursachen für Fehlerentstehung, ...

Gegenmaßnahmen zur Gefährdungsabwendung:

1. Überwachung und geeignete Reaktion auf erkannte FF.
2. Test und Fehlerbeseitigung,
3. Fehlervermeidung durch Verbesserung der Entstehungsprozesse.

Was kostet Verlässlichkeit?

Der Preis für Verlässlichkeit sind die Gesamtkosten aller Maßnahmen für die Gefährdungsabwendung auf allen drei Ebenen:

1. Kontrollen und geeignete Reaktion auf erkannte FF: Kann mehr als 50% der Gesamtfunktionalität erfordern, plus Kosten für Reparatur, Schadensbegrenzung, ...
2. Test, Fehlersuche und Fehlerbeseitigung: Für HW und SW typisch mehr als 50% des Gesamtentwurfsaufwand.
3. Fehlervermeidung durch Verbesserung der Entstehungsprozesse: Kosten für die Qualitätssicherung und die Weiterentwicklung und Verbesserung der Entstehungsprozesse.

Verlässlichkeit ist selbst für IT-Systemen ohne erhöhte Anforderungen die teuerste Produkteigenschaft. Bei erhöhten Anforderungen betragen die anteiligen Gesamtkosten für die Sicherung der Verlässlichkeit weit über 50%.

Der Preis fehlender Verlässlichkeit

Wenn Verlässlichkeit teuer, warum kein Verzicht? – Schadenskosten:

- Datenverlust, Hintertüren für den Datenmissbrauch[1],
- 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*[2] *...*

Ursache beinahe atomarer Schlagabtausch: defekter Schaltkreis.

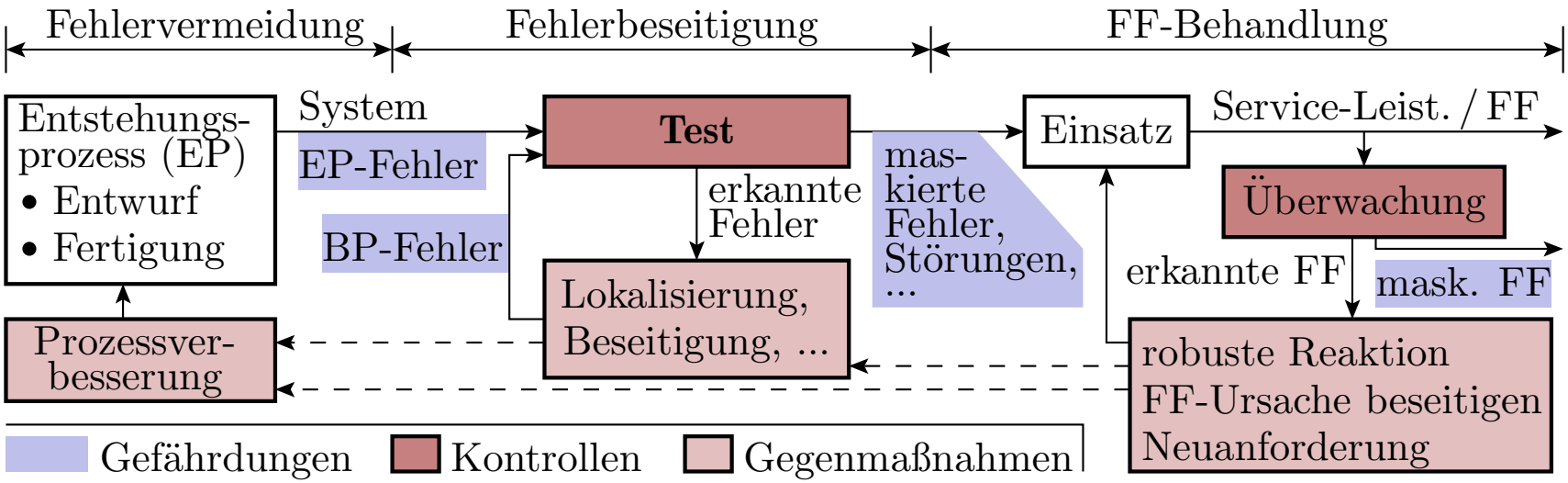
Unzuverlässige IT-Systeme können nicht eingesetzt werden.

---

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

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

Warum heißt Vorlesung »Test & Verlässlichkeit«



Verlässlichkeit wird durch Iterationen aus Kontrollen, der Beseitigung erkannter Gefährdungen und Erfolgskontrollen gesichert. Mit der unterstellten Fehlerkultur »Beseitigung alle erkannten Gefährdungen (FF, Fehler, ...)« hängt die Verlässlichkeit der Systeme im Einsatz hauptsächlich von den durchgeführten Tests und Kontrollen auf den drei Ebenen ab.

Lernziel und Inhalt

Lernziel

Kennenlernen der Gefährdungen und der Maßnahmen zu Gefährdungsabwendung mit Schwerpunkt auf Tests und Kontrollen, und wie sich deren Güte auf die Verlässlichkeit eingesetzter IT-Systeme auswirkt.

Entstehung und Abwendung von Gefährdungen sowie Einfluss nicht abgewendeter Gefährdungen auf die Verlässlichkeit sind stochastischer Natur. Hierzu themenspezifische Einführung in die Stochastik.

Foliensätze:

1. Gefährdungen, Gegenmaßnahmen, Kenngrößen.
2. Wahrscheinlichkeit: Grundl., Markov-Ketten, Fehlernachweis, ...
3. Verteilungen: Fehleranzahl, Nachweislänge, ...
4. Kontrollen: mixed Signal, Inspektion, digital (automatisierbar), ...
5. HW: Fehlermodellierung, Testsuche, Selbsttest.
6. SW: Fehlervermeidung, Testauswahl.
7. Ausfälle und Fehlertoleranz.

Verlässlichkeit

Beschreibung der Verlässlichkeit

Die Verlässlichkeit von IT-Systemen wird durch die Entstehung und Abwendung von Gefährdungen auf 3 Ebenen beschrieben:

	entstehende Gefährdungen	Gefährdungsabwendung
Fehlervermeidung	Fehler	Minderung Entstehungsrate
Test und Fehlerbeseitigung	Fehler durch Reparatur	Beseitigung vorhandener Fehler
Betrieb + Kontrolle + FF-Behandlung	FF und Schaden durch FF	Schadensvermeidung und Korrektur von FF

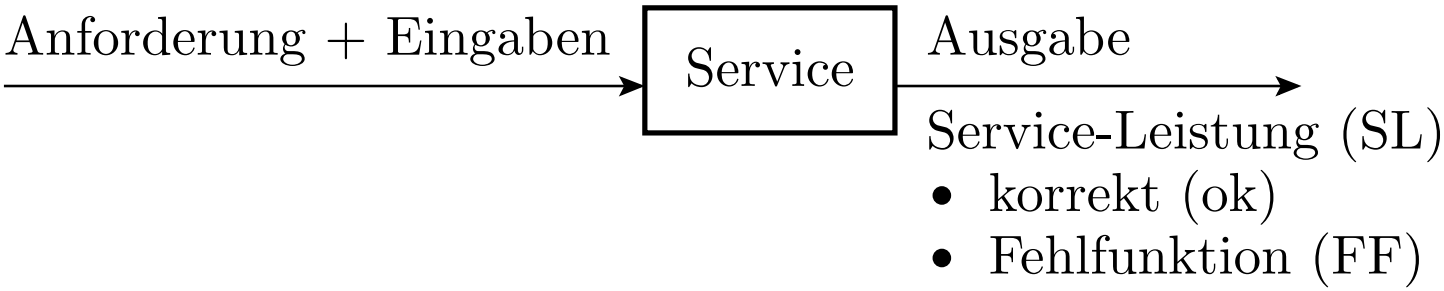
Eine quantitative Schätzung der Verlässlichkeit benötigt

- Zählwerte für entstandene, vermiedene, ..., nicht erkannte FF,
- dasselbe für Fehler und deren Entstehungsursachen,

Deshalb müssen wir IT-Systeme so modellieren, dass erbrachte Leistungen, Fehlfunktionen, Fehler, ... zählbar sind.

Das Service-Modell

Das Service-Modell



Ein »Service« oder »Service-Leister« ist für uns im weiteren ein System, dass auf Anforderung aus Eingaben Ausgaben erzeugt. Erbrachte Service-Leistungen (SL) werden unterteilt in

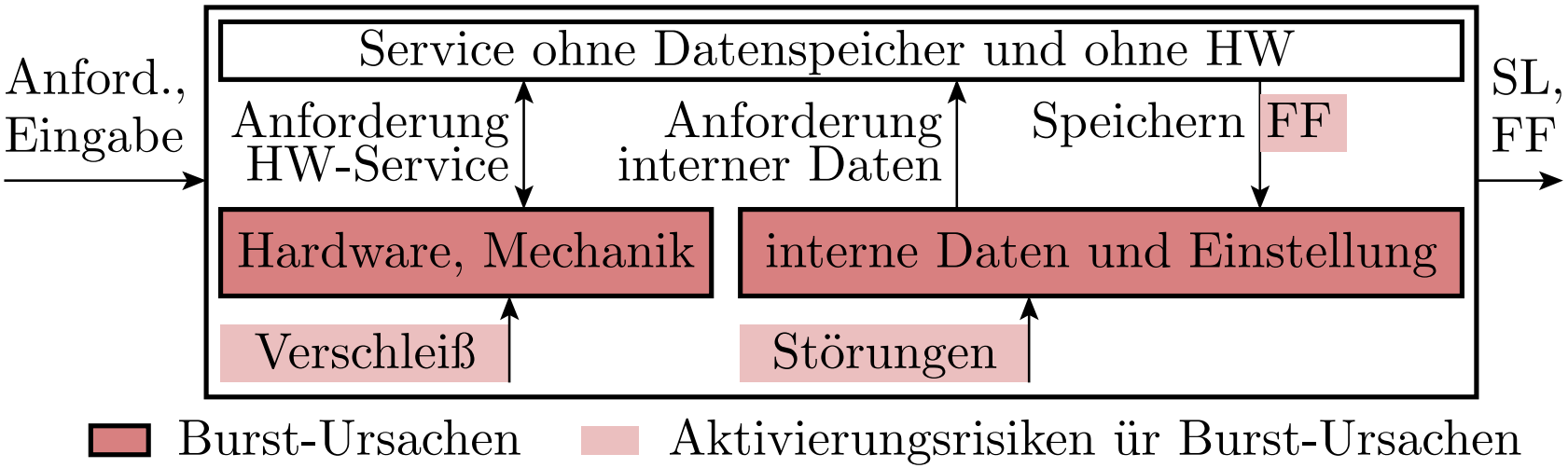
- korrekt und
- fehlerhaft, Fehlfunktion, FF.

Schätzbare Kennwerte zu Beschreibung Verlässlichkeit:

- Verfügbarkeit: Zeitanteil, in dem Service auf Anford. SL liefert,
- Zuverlässigkeit*: zu erwartende Anzahl der SL je FF und
- Sicherheit(en)*: zu erwartende Anzahl der SL je sicherheitsgefährdende FF.

* Zweckmäßige, in der Fachwelt jedoch noch unübliche Definitionen.

Fehlfunktions-Burst



HW-Ausfällen und / oder Verfälschung gespeicherter Daten beeinträchtigen nicht nur eine, sondern alle folgende SL:

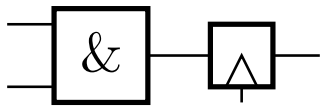
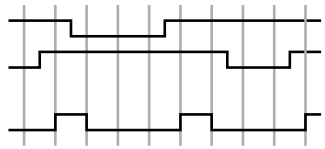
- keine weitere Service-Ausführung,
- erkennbar erhöhte FF-Rate oder
- nicht erkennbar erhöhte FF-Rate

bis defekte HW repariert und Datenverfälschungen behoben sind.

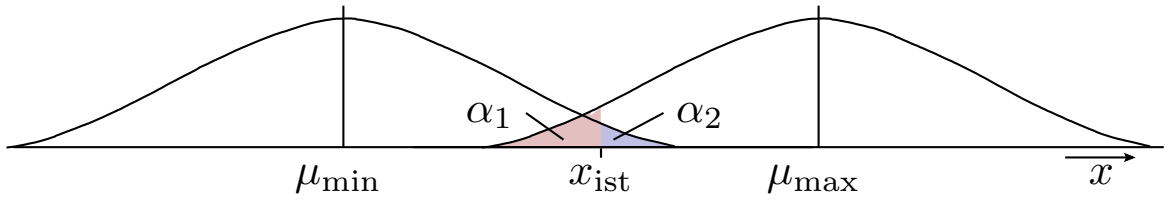
Erkannte MF-Bursts zählen nur als eine FF und SL und das System gilt als nicht verfügbar, solange die Ursache nicht beseitigt ist.

Anwendungsbereiche des Service-Modells

Das Service-Modell ist sehr universell und auf unterschiedliche Abstaktionsebenen für IT-Systeme, aber auch menschliche Dienstleistungen, 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	

Mindestzählwerte zur Kenngrößenschätzung



α_1,α_2 – Irrtumswahrscheinlichkeiten, dass der Erwartungswert größer $\mu_{\max}$ oder kleiner $\mu_{\min}$ ist.

Viele der nachfolgend eingeführten Kenngrößen werden auf Basis von Erwartungswerten für Zählwerte definiert:

- FF (aufgetretene, vermiedene, ...),
- Fehler (vorhandenen, modellierte, nachweisbare, vermiedene, ...).

Aussagekräftige Bereichsschätzungen verlangen geeignete Zählwertgrößen (ACR, **a**ppropriate **c**ounting **r**anges). Für unabhängige Zählwerte:

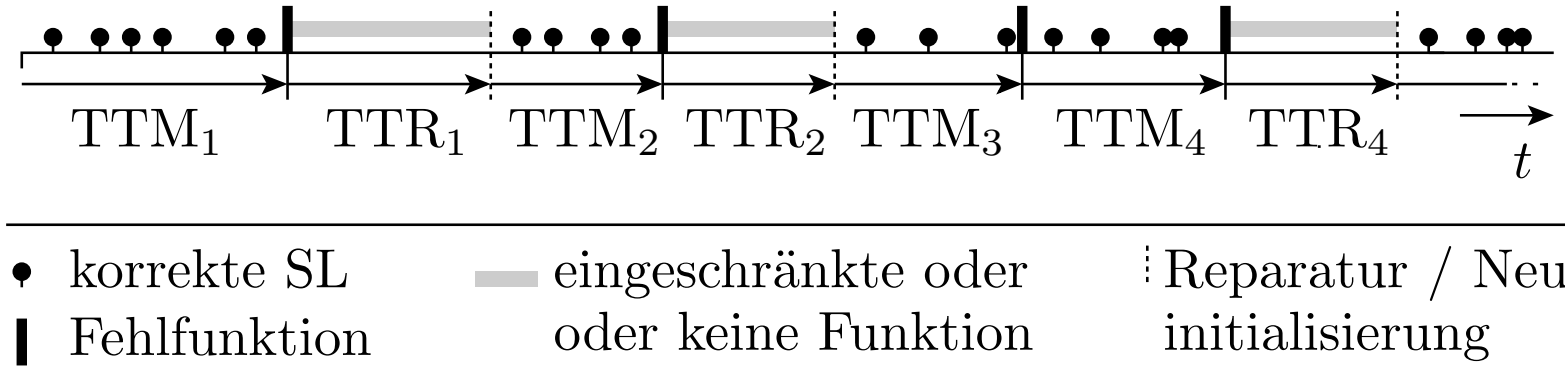
$$x_{\min} + 10 \leq x_{\text{ist}} \leq x_{\max} - 10$$

Abhängige Zählwerte müssen noch mehr von ihren Minima oder Maxima abweichen, (siehe F3.2.7 *Bereichsschätzung für Zählwerte*).

Verfügbarkeit

$MTTF$, $MTTR$, Verfügbarkeit und PFD

Die Verfügbarkeit ist der Zeitanteil, in dem das System nutzbar, d.h. weder ausgefallen noch abgestürzt noch bei einer FF-Behandlung ist:



- Mittlere Zeit bis zur nächsten FF (**m**ean **t**ime **t**o **m**alfunction):

$$MTTM = \frac{1}{\#MF} \cdot \sum_{i=1}^{\#MF} TTM_i \Big|_{\text{ACR}}$$

$\#TTM_i$	Zeit bis zur nächsten FF (**t**ime **t**o **m**alfunction).
$\#TTR_i$	Reparaturdauer (**t**ime **t**o **r**repair).
$\#MF$	Anzahl der Fehlfunktionen (number of **m**al**f**unctions).
ACR	Geeignete Zählwertgrößen (**a**ppropriate **c**ounting **r**anges).

- Mittlere Reparaturzeit bis Austausch ausgefallene Hardware und/oder Neuinitialisierung (**m**ean **t**ime **t**o **r**epair):

$$MTTR = \frac{1}{\#MF} \cdot \sum_{i=1}^{\#MF} TTR_i \Bigg|_{\text{ACR}}$$

- Verfügbarkeit (**a**vailability):

$$A = \frac{MTTM}{MTTM + MTTR} = \frac{\sum_{i=1}^{\#FF} TTF_i}{\sum_{i=1}^{\#FF} TTM_i + \sum_{i=1}^{\#FF} TTR_i} \Bigg|_{\text{ACR}}$$

- PFD (**p**robability of **f**ailure on **d**emand):

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

$\#TTM_i$	Zeit bis zur nächsten FF (**t**ime **t**o **m**alfunction).
$\#TTR_i$	Reparaturdauer (**t**ime **t**o **r**repair).
$\#MF$	Anzahl der Fehlfunktionen (number of **m**al**f**unctions).
ACR	Geeignete Zählwertgrößen (**a**ppropriate **c**ounting **r**anges).
A	Verfügbarkeit (**a**vailability).

Reparaturzeiten für hochverfügbare Systeme

Verfügbarkeit A	PFD	Summe aller Reperaturzeiten	
		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

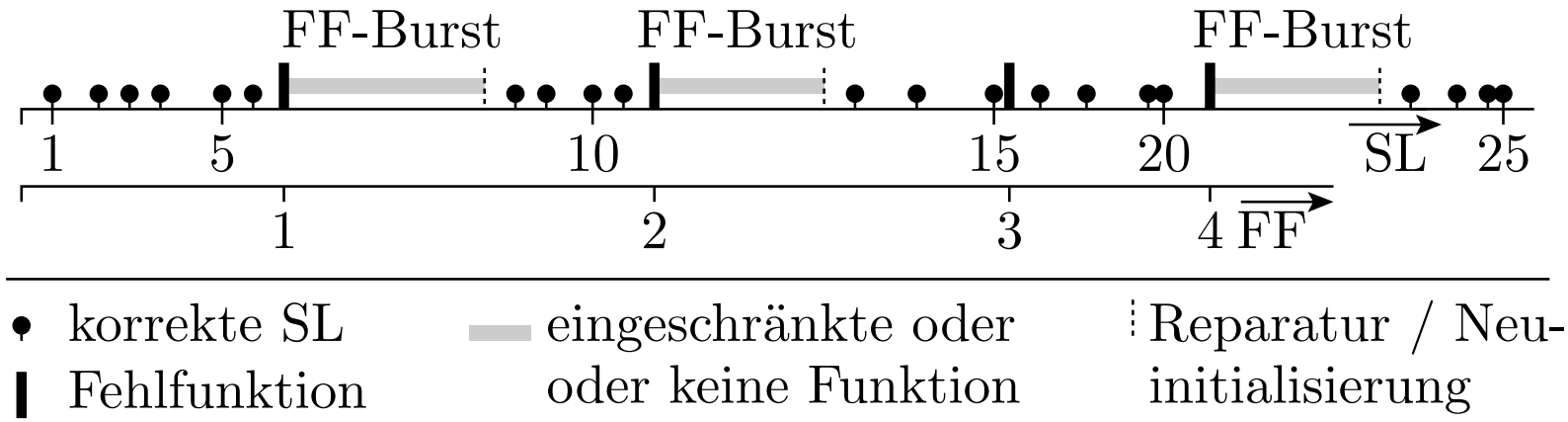
$A \approx 99\%$ ist normal. Hohe Verfügbarkeiten ab 99,9% verlangen spezielle Maßnahmen:

- unterbrechungsfreie Stromversorgung,
- RAID (**R**edundant **A**rray of **I**ndependent **D**isks),
- gespiegelte Server, vorbeugende Wartung, ...

(siehe F7.2 *Ausfall-Toleranz*).

Zuverlässigkeit

Zuverlässigkeit



Zuverlässigkeit sei im weiteren die zu erwartende Anzahl der SL je FF:

$$R = \left.\frac{\#SR}{\#MF}\right|_{\mathrm{ACR}} \tag{1}$$

Während FF-Bursts und anderen Gründen für Nichtverfügbarkeit zählen die SL und FF nicht. Auch weil die Zählwerte für die Fehlfunktionen innerhalb einer Burst voneinander abhängen.

$\#SR$	Anzahl der Service-Leistungen (number of **s**ervice **r**esults).
$\#MF$	Anzahl der Fehlfunktionen (number of **m**al**f**unctions).
ACR	Geeignete Zählwertgrößen (**a**ppropriate **c**ounting **r**anges).

Fehlfunktionsrate und $MTTF$

Die Fehlfunktionsrate ist der Kehrwert der Zuverlässigkeit:

$$\zeta = \frac{1}{R} = \left.\frac{\#MF}{\#SR}\right|_{\mathrm{ACR}} \tag{2}$$

Beides ist auch beschreibbar als Verhältnis aus mittlerer Zeit bis zu nächsten FF und der mittleren Service-Dauer

$$R = \frac{MTTM}{MTS}; \quad \zeta = \frac{MTS}{MTTM}$$

Beispiel 1

Innerhalb von 300 h Programmnutzung 30 FF, $MTS = 0{,}1\,\mathrm{h}$. Schätzen Sie Zuverlässigkeit und FF-Rate.

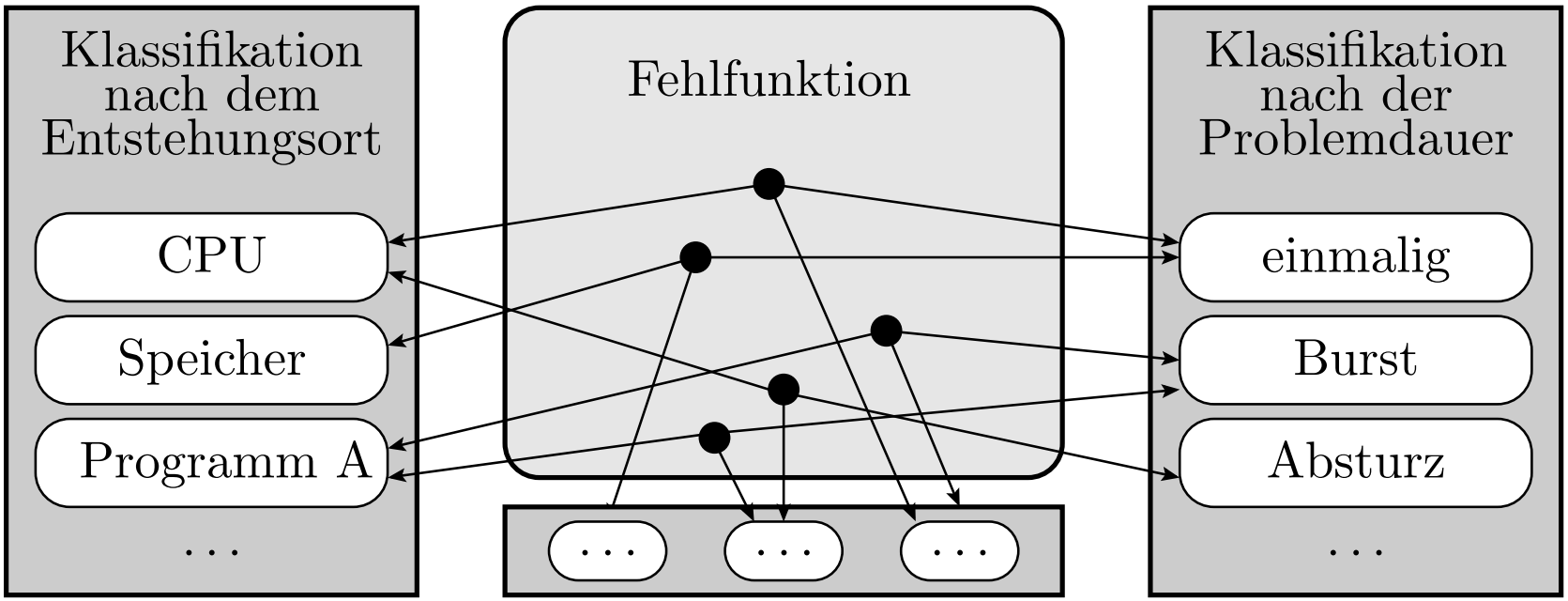
$$MTTM = \frac{300\,\mathrm{h}}{30\,[\mathrm{FF}]} = 10\,\mathrm{h}; \quad R = \frac{10\,\mathrm{h}}{0{,}1\,\mathrm{h}} = 100\,\left[\frac{\mathrm{SR}}{\mathrm{MF}}\right]; \quad \zeta = Z^{-1} \approx 10^{-2}\,\left[\frac{\mathrm{MF}}{\mathrm{SL}}\right]$$

$MTTM$ mittlere Zeit bis zur nächsten FF (**m**ean **t**ime **t**o **m**alfunction).

MTS mittlere Service-Dauer (**m**ean **t**ime to **s**ervice).

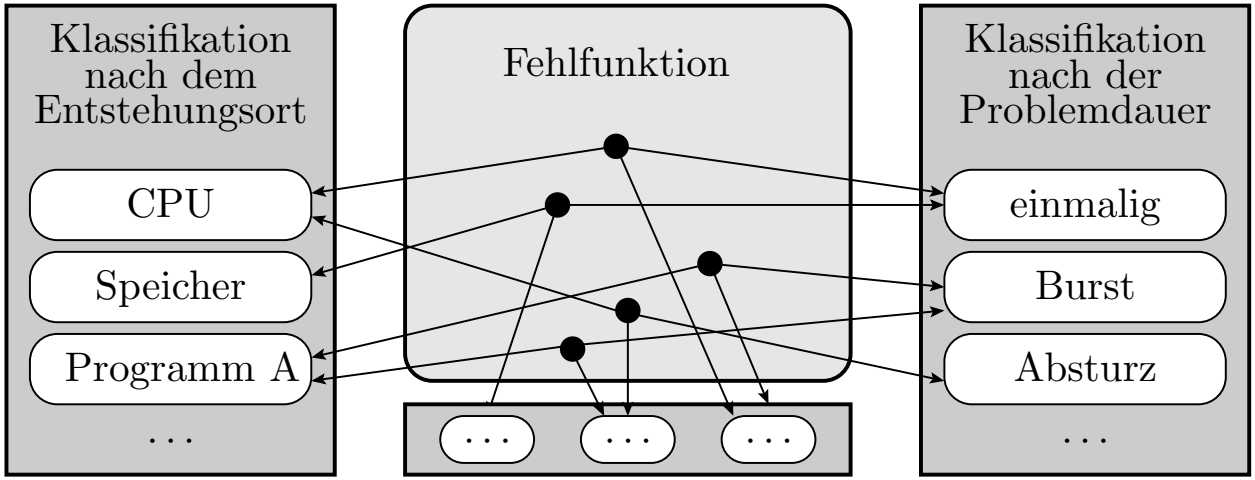
$\left[\frac{\mathrm{SL}}{\mathrm{FF}}\right]$, $\left[\frac{\mathrm{FF}}{\mathrm{SL}}\right]$ in **S**ervice-**L**eitungen pro **F**ehl**f**unktion bzw. ungekehrt.

Teilzuverlässigkeiten



Die Fehlfunktionen (MF) eines Systems können in unterschiedlicher Weise klassifiziert werden, z.B.

- nach Ort, Ursache, Schaden, ... :
- nur FF eines bestimmten Teilsystems,
- nur durch HW, nur durch SW verursachte FFs,
- nur FF, die die Betriebs- / Daten- / Zugangssicherheit mindern, ...



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

$$\#MF = \sum_{i=1}^{\#MFC} \#MF_i$$

Die Fehlfunktionsrate ist die Summe der Fehlfunktionsraten aller Fehlfunktionsklassen:

$$\frac{\#MF}{\#SR} = \sum_{i=1}^{\#MFC} \frac{\#MF_i}{\#SR}; \quad \zeta = \sum_{i=1}^{\#MFC} \zeta_i$$

$\#MFC$ Anzahl der FF-Klassen (number of **m**al**f**unction **c**lasses).

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

$$\frac{1}{R} = \sum_{i=1}^{\#MFC} \frac{1}{R_i}$$

Beispiel 2

FFs seien entweder vom Speicher, vom Prozessor, von der Software oder vom Rest verursacht. Die Teilsysteme haben folgende $MTTM$s:

Teilsystem i	Speicher	Prozessor	Software	alle anderen
$MTTF_i$	500 h	3.000 h	1000 h	2.000 h

Mittlere Service-Dauer $MTS = 1\,\mathrm{min}$.

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

$MTTM$ mittlere Zeit bis zur nächsten FF (**m**ean **t**ime **t**o **m**alfunction).

Lösung

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

Teilsystem i	Speicher	Prozessor	Software	Rest
$MTTM_i$ in min	$3 \cdot 10^{-4}$	$18 \cdot 10^{-4}$	$6 \cdot 10^{-4}$	$12 \cdot 10^{-4}$
$R_i = \frac{MTTM_i}{MTS}$ Verhältnis $\frac{\text{SL}}{\text{FF}}$	$3 \cdot 10^{-4}$	$18 \cdot 10^{-4}$	$6 \cdot 10^{-4}$	$12 \cdot 10^{-4}$
$\zeta_i = \frac{1}{R_i}$, Verhältnis $\frac{\text{FF}}{\text{SL}}$	$3{,}33 \cdot 10^{-5}$	$5{,}56 \cdot 10^{-6}$	$1{,}67 \cdot 10^{-5}$	$8{,}33 \cdot 10^{-6}$

2. FF-Rate und Zuverlässigkeit des Gesamtsystems:

$$\begin{aligned} \zeta &= \left(3{,}33 \cdot 10^{-5} + 5{,}56 \cdot 10^{-6} + 1{,}67 \cdot 1 + 8{,}33 \cdot 10^{-6}\right) \left[\frac{\text{FF}}{\text{SL}}\right] \\ &= 6{,}39 \cdot 10^{-5} \left[\frac{\text{FF}}{\text{SL}}\right] \\ R &= \frac{1}{\zeta} = 1{,}57 \cdot 10^{4} \left[\frac{\text{SL}}{\text{FF}}\right] \end{aligned}$$

$MTTM$ mittlere Zeit bis zur nächsten FF (**m**ean **t**ime **t**o **m**alfunction).

MTS mittlere Service-Dauer (**m**ean **t**ime to **s**ervice).

$\left[\frac{\text{SL}}{\text{FF}}\right]$ in Service-Leistungen pro Fehlfunktion.

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: Kleine Schäden an Anlagen und Eigentum.
- SIL2: Große Schäden an Anlagen, Personenverletzung.
- SIL3: Verletzung von Personen, einige Tote.
- SIL4: Katastrophen, viele Tote, gravierende Umweltschäden.

Die Sicherheitsstufe legt weitere Kenngrößen fest, z.B. Obergrenzen

- PFH (**p**robability of **f**ailure per **h**our) and
- PFD (**p**robability of **f**ailure on **d**emand):

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}

Wir definieren Sicherheit alternativ als eine Teilzuverlässigkeit.

Definition der Sicherheit als Teilzuverlässigkeit

Sicherheit bezieht sich immer auf eine angenommene Gefährdung:

Sicherheit	Sicher vor welchen Gefährdungen?
Betriebssicherheit (safty)	Personen- und Umweltschäden
Datensicherheit (security)	Datendiebstahl
Sicherheit Datenerhalt	Datenverlust
...	...

Sicherheiten sind Teilzuverlässigkeiten, bei denen nur die FF ausgewählter Gefährdungen zählen:

$$S = \left.\frac{\#SR}{\#HM}\right|_{\mathrm{ACR}} \tag{3}$$

Rate der gefährdenden MF als Kehrwert der Sicherheit:

$$\zeta_{\mathrm{S}} = \frac{1}{S}$$

$\#HM$	Anzahl der sicherheitsgefährdenden FF (number of **h**azzardous **m**alfunctions).
ACR	Geeignete Zählwertgrößen (**a**ppropriate **c**ounting **r**anges).

Sicherheit und Zuverlässigkeit

Die sicherheitsgefährdenden FF sind ein kleiner Anteil aller FF:

$$\eta_{\mathrm{SE}} = \frac{\zeta_{\mathrm{S}}}{\zeta} \ll 1$$

Sicherheit und mittlere Zeit bis zur nächsten Gefährdung in Abhängigkeit von η_{g}:

$$S = \frac{R}{\eta_{\mathrm{g}}}; \qquad MTTH_{\mathrm{S}} = \frac{MTTM}{\eta_{\mathrm{SE}}}$$

Maßnahmen zur Erhöhung der Sicherheit eines Systems:

- Zuverlässigkeit ↑, insbesondere sicherheitskritischer Teile,
- Verringerung des Anteils der gefährdenden FF η_{g} durch
  - Überwachung und robuste Reaktion auf erkannte FF,
  - spezielle Funktionen zum Ausschluss gefährdender FF, z.B. Ruhestromprinzip, (siehe F7.2 *Ruhestromprinzip*).

η_{SE} Anteil der sicherheitsgefährdenden FF (percentage of **s**afety **e**ndagering MF).

$MTTH$ mittlere Zeit bis zur nächsten Gefährdung (**m**ean **t**ime **t**o **h**azzard).

$MTTM$ mittlere Zeit bis zur nächsten FF (**m**ean **t**ime **t**o **m**alfunction).

Beispiel 3 (Sicherheit durch Zusatzsteuergerät)

Eine Fahrzeug habe eine $MTTF$ =1000 h bis zu einer FF. Der Anteil der betriebssicherheitsgefährdenden FF sei $\eta_{\mathrm{SE}} = 1\%$ und die mittlere Service-Dauer (mittlere Fahrtdauer) betrage $MTS = 1\,\mathrm{h}$. Ein zusätzliches elektronisches Steuergerät mit Zuverlässigkeit R_{CD} verringert den Anteil der gefährdenden FF auf $\eta_{\mathrm{SE1}} = 10^{-3}\ \frac{\mathrm{GFF}}{\mathrm{FF}}$.

1. Wie groß sind die Zuverlässigkeit R_{MNT} und die Sicherheit S_{MNT} des Systems ohne das zusätzliche Steuergerät?
2. Wie hoch muss die Zuverlässigkeit des Steuergeräts R_{CD} mindestens sein, damit das zusätzliche Steuergerät die Sicherheit des Gesamtsystems mindestens verfünffacht ($S \geq 5 \cdot S_{\mathrm{NMT}}$)?

$MTTM$	mittlere Zeit bis zur nächsten FF (**m**ean **t**ime **t**o **m**alfunction).
MTS	mittlere Service-Dauer (**m**ean **t**ime to **s**ervice).
η_{SE}	Anteil der sicherheitsgefährdenden FF (percentage of **s**afety **e**ndagering MF).
R_{NMT}	Zuverlässigkeit ohne Fehlfuntionsbehandlung (**R**eliability with **n**o **m**ulfunction **t**reatment).
S	Sicherheit mit Fehlfuntionsbehandlung (**s**afety (security) with **m**alfunction **t**reatment).
S_{NMT}	Sicherheit ohne Fehlfuntionsbehandlung (**s**afety with **n**o **m**alfunction **t**reatment).

Lösung Aufgabenteil 1

Zuverlässigkeit ohne zusätzliches Steuergerät:

$$R_{\mathrm{MNT}} = \frac{MTTH}{MTS} = \frac{10^3\mathrm{h}}{1\mathrm{h}} = 10^3 \ \left[\frac{\mathrm{SL}}{\mathrm{FF}}\right]$$

$MTTH$ ohne zusätzliches Steuergerät:

$$MTTH = \frac{MTTF}{\eta_{\mathrm{SE}}} = \frac{1000}{1\%}\mathrm{h} = 10^5\mathrm{h}$$

Betriebssicherheit ohne zusätzliches Steuergerät:

$$S_{\mathrm{MNT}} = \frac{MTTH}{MTS} \approx \frac{10^5\mathrm{h}}{1\mathrm{h}} = 10^5 \ \left[\frac{\mathrm{SL}}{\mathrm{GF}}\right]$$

$MTTM$	mittlere Zeit bis zur nächsten FF (**m**ean **t**ime **t**o **m**alfunction).
MTS	mittlere Service-Dauer (**m**ean **t**ime to **s**ervice).
η_{SE}	Anteil der sicherheitsgefährdenden FF (percentage of **s**afety **e**ndagering MF).
R_{NMT}	Zuverlässigkeit ohne Fehlfuntionsbehandlung (**R**eliability with **n**o **m**ulfunction **t**reatment).
S_{NMT}	Sicherheit ohne Fehlfuntionsbehandlung (**s**afety with **n**o **m**alfunction **t**reatment).

$\left[\frac{\mathrm{SL}}{\mathrm{FF}}\right]$ in Service-Leistungen je Fehlfunktion; $\left[\frac{\mathrm{SL}}{\mathrm{GF}}\right]$ in SL je sicherheitsgefährdender FF.

Lösung Aufgabenteil 2

Ein zusätzliches elektronisches Steuergerät verringert den Anteil sicherheitsgefährdenden FF auf $\eta_{\mathrm{SE1}} = 0.1 \cdot \eta_{\mathrm{SE}} = 10^{-3}$. Welche Zuverlässigkeit R_{SG} muss das Steuergerät mindesten haben, damit sich die Gesamtsicherheit verfünffacht:

$$S \geq 5 \cdot S_{\mathrm{MNT}} = 5 \cdot \frac{R_{\mathrm{MNT}}}{\eta_{\mathrm{SE}}}$$

$$S = \frac{1}{\eta_{\mathrm{SE1}}} \cdot \frac{1}{\frac{1}{R_{\mathrm{MNT}}} + \frac{1}{R_{\mathrm{SG}}}} \geq 5 \cdot \frac{R_{\mathrm{MNT}}}{\eta_{\mathrm{SE}}}$$

$$\frac{1}{\frac{R_{\mathrm{MNT}}}{R_{\mathrm{MNT}}} + \frac{R_{\mathrm{MNT}}}{R_{\mathrm{SG}}}} \geq \frac{5 \cdot \eta_{\mathrm{SE1}}}{\eta_{\mathrm{SE}}} = 0{,}5$$

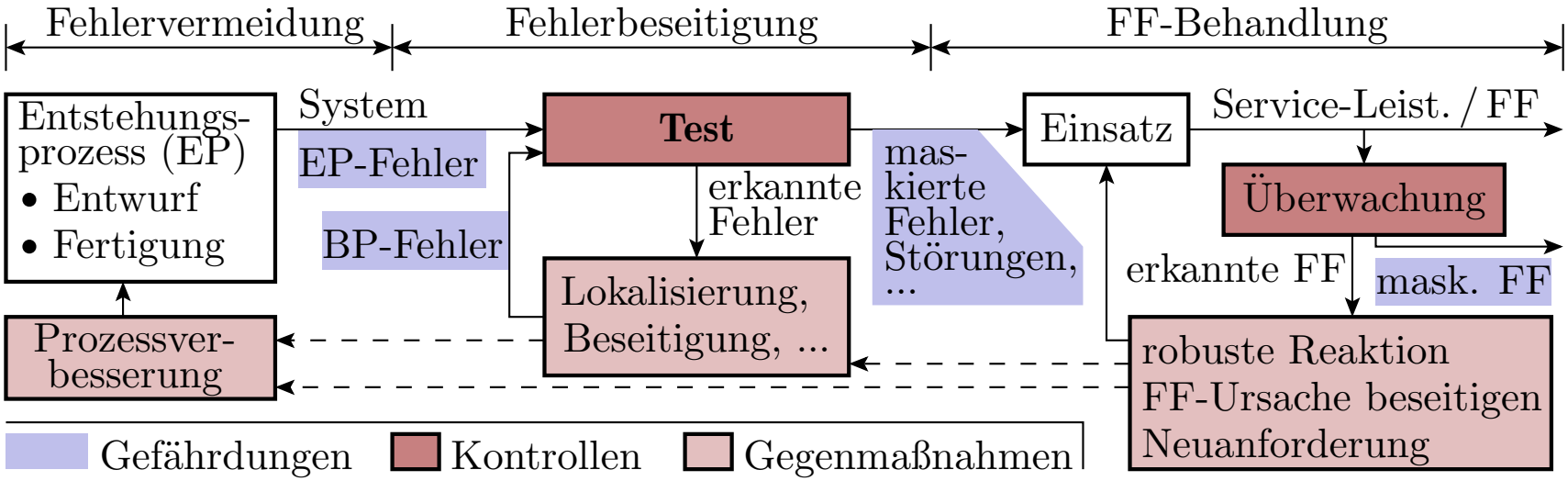
$$1 + \frac{R_{\mathrm{MNT}}}{R_{\mathrm{SG}}} \geq 2$$

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

Alternativ zu aktuellen Ethik-Diskussionen, ob autonome Fahrzeuge Kinder, Rentner, ... überfahren sollen, Einfordern der vielfachen Sicherheit gegenüber fahrergesteuerten Fahrzeugen + Haftpflicht für Fahrzeug.

Zusammenfassung

Verlässlichkeit



Verlässlichkeit wird auf drei Ebenen gesichert.

In der Vorlesung betrachten wir IT-Systeme als Service-Leister:

- Systeme, die auf Anforderung aus Eingaben Ausgaben erzeugen.
- Wesentlich ist nur, dass sich die SL und FF zählen lassen.

Kenngrößen:

- Verfügbarkeit: Zeitanteil, den das System verfügbar ist.
- Zuverlässigkeit: Anzahl der SL je Fehlfunktion.
- Sicherheit(en): Anzahl der SL je gefährdende Fehlfunktion.

Vorlesung 2: Geplante Themen

Fehlervermeidung	Fehlerbeseitigung	FF-Behandlung
Beseitigung von Fehler-entstehungsursachen	Test und Beseitigung erkannter Fehler	Überwachung, robuste R. Fehlertoleranz Störungen

Abschn. 2 FF-Behandlung:

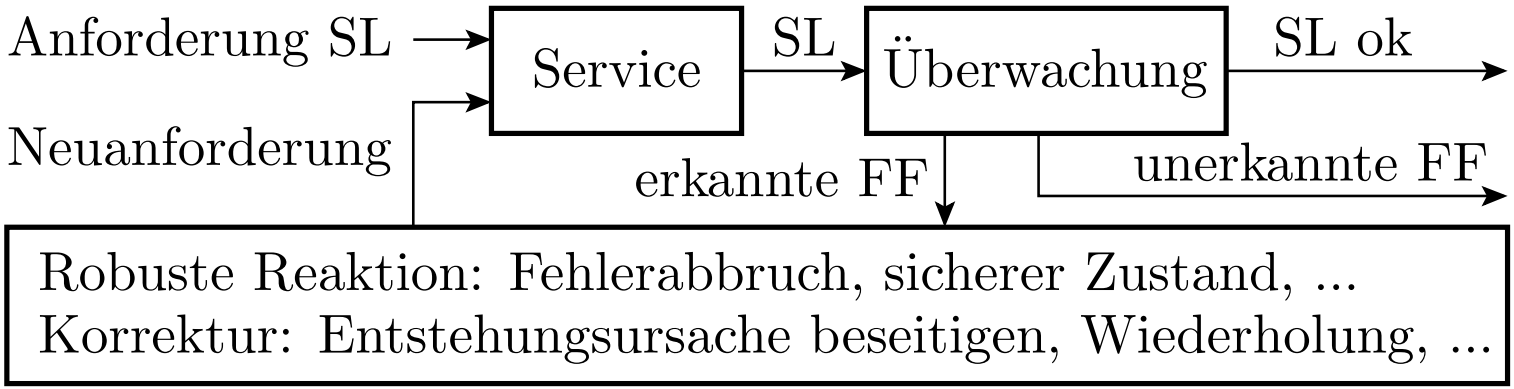
- Modellierung und Parameter Überwachung und Umgang mit FF.
- Möglichkeiten der Überwachung digitaler Service-Ergebnisse: Einteilung, Überblick.
- Umgang mit Fehlfunktionen: Lösungen und Probleme.

Abschn. 3 Fehlerbeseitigung, ersten Themen:

- Der gesamte Fehlerbeseitigungsprozess.
- Techniken der Fehlerlokalisierung.
- Statische und dynamische Tests.
- Testkenngrößen.

FF-Behandlung

FF-Behandlung im laufenden Betrieb



Überwachung der SL

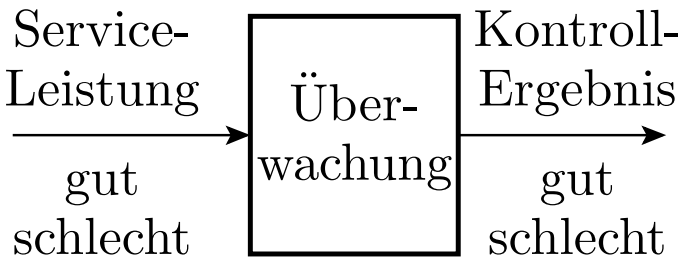
- erkennt nur einen Teil der FF und
- klassifiziert möglicherweise korrekte SL als FF.

Reaktion auf erkannte FF:

- Robuste Reaktion: Kontrolliertes Verhalten, um Schäden und Gefahren zu vermeiden. Sicherheitsverbesserung.
- Fehlfunktionstoleranz: Korrektur von FF. Zuverlässigverbeserung.
- Ausfalltoleranz: Übernahme der Aufgaben ausgefallenen Einheiten durch Reserveeinheiten. Erhöhung der Verfügbarkeit.

Kenngrößen Überwachung

Kenngrößen der Überwachung



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

1. MF-Überdeckung (MF coverage), Anteil nachweisbare FF:

$$MC = \left.\frac{\#DM}{\#MF}\right|_{\mathrm{ACR}}$$

2. Phantom-FF-Rate, Anteil der korrekten SL, die als FF klassifiziert werden:

$$\zeta_{\mathrm{Phan}} = \left.\frac{\#PM}{\#SR}\right|_{\mathrm{ACR}}$$

$\#DM$	Anzahl der erkannten FF (number of **d**etected **M**Fs).
$\#MF$	Anzahl der Fehlfunktionen (number of **m**al**f**unctions).
$\#PM$	Anzahl der Phantom-FF (number of **p**hantom **m**alfunction).
ACR	Geeignete Zählwertgrößen (**a**ppropriate **c**ounting **r**anges).

Aus Phantom-FF können, wenn nicht als solche erkannt, bei Korrekturversuchen echte FF werden.

Beispiel 4

System: FF-Rate ohne Überwachung $\zeta_{\mathrm{NMT}} = 1\% \left[\frac{\mathrm{MF}}{\mathrm{SR}}\right]$,
Kenngrößen der Überwachung: $MC = 80\%$, $\zeta_{\mathrm{Phan}} = 2\% \left[\frac{\mathrm{PM}}{\mathrm{SR}}\right]$.

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

$$\zeta_1 = \zeta_{\mathrm{NMT}} \cdot (1 - MC) = 0{,}2\% \left[\frac{\mathrm{MF}}{\mathrm{SR}}\right]$$

- Wenn Phantom-FF richtige FF werden:

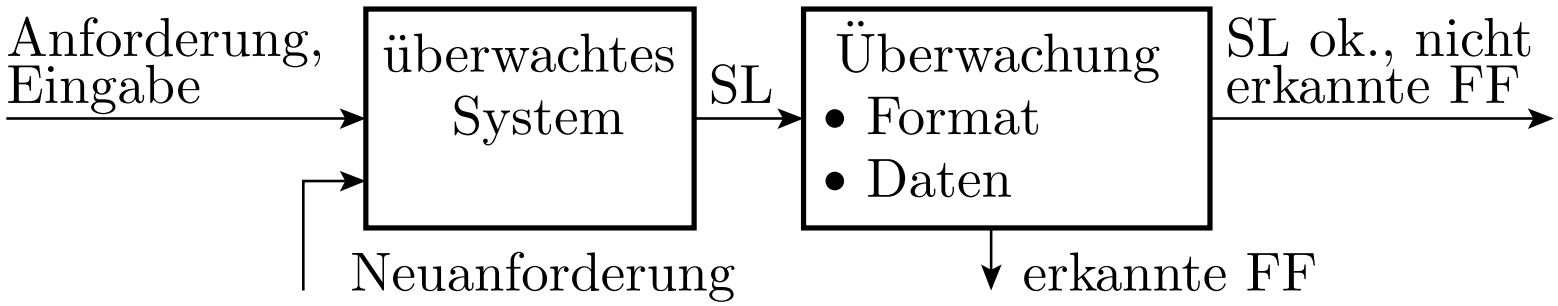
$$\zeta_2 = \zeta_1 + \zeta_{\mathrm{Phan}} = 2{,}2\% \left[\frac{\mathrm{MF}}{\mathrm{SR}}\right]$$

MC	Fehlfunktionsüberdeckung (**m**alfunction **c**overage), Anteil der nachweisbaren FF.
ζ_{NMT}	FF-Rate ohne FF-Behandlung (MF rate with **n**o **m**ulfunction **t**reatment).
ζ_{Phan}	Phantom-FF-Rate (**phan**tom MF rate).

$\left[\frac{\mathrm{FF}}{\mathrm{SL}}\right]$ in FF je Service-Leistung; $\left[\frac{\mathrm{PFF}}{\mathrm{SL}}\right]$ in Phantom-FF je Service-Leistung.

Überwachungstechniken

Überwachungstechniken und ihre Eigenschaften



Service-Leistungen umfassen:

- Format: werteunabhängige Merkmale: Zeitschranken, WB, ...
- Daten: Werte der Datenobjekten.

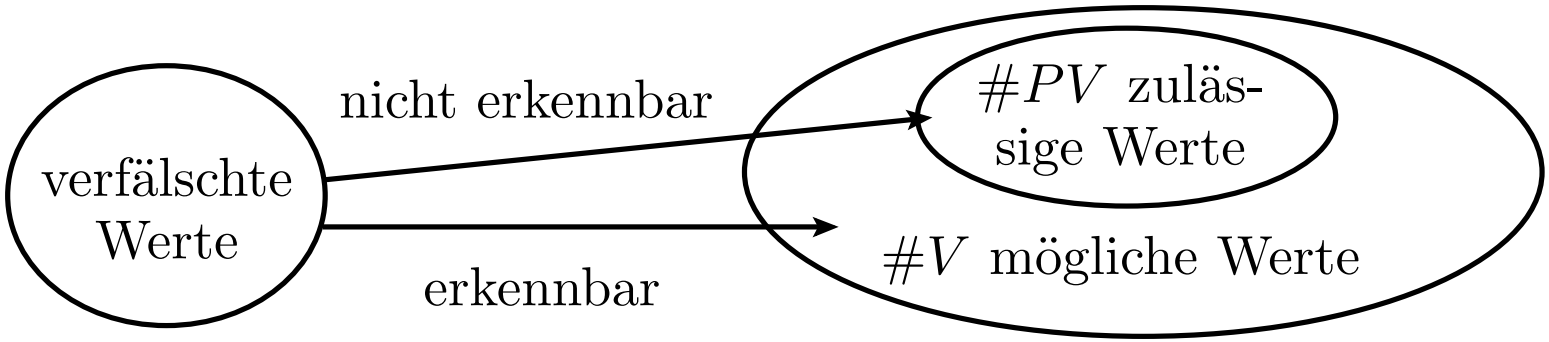
Einteilung der Überwachungsverfahren für digitale SL:

1. Formatkontrollen: nur Kontrolle werteunabhängiger Merkmale. SL mit Formatfehlern sind immer falsch und SL mit korrektem Format können falsche Daten haben, d.h. nur Kontrolle auf Zulässigkeit.
2. Wertekontrollen: (Zusätzliche) Kontrolle von Datenwerten.

Formatkontrollen sind einfache durchzuführen und erzielen bei digitalen SL oft höhere MC und kleinere Phantom-FF-Raten.

Ausnutzung von Datenredundanz

Formatkontrollen (Fehlererkennende Codes, Prüfkennzeichen, Wertebereichskontrolle, ...) nutzen meist Informationsredundanz.



Fehlfunktionsüberdeckung ist der Anteil der auf unzulässige Werte abgebildeten fehlerhaften Werte. Wenn Verfälschungen gleich häufig auf mögliche Werte abgebildet und alle unzulässigen Werte erkannt werden

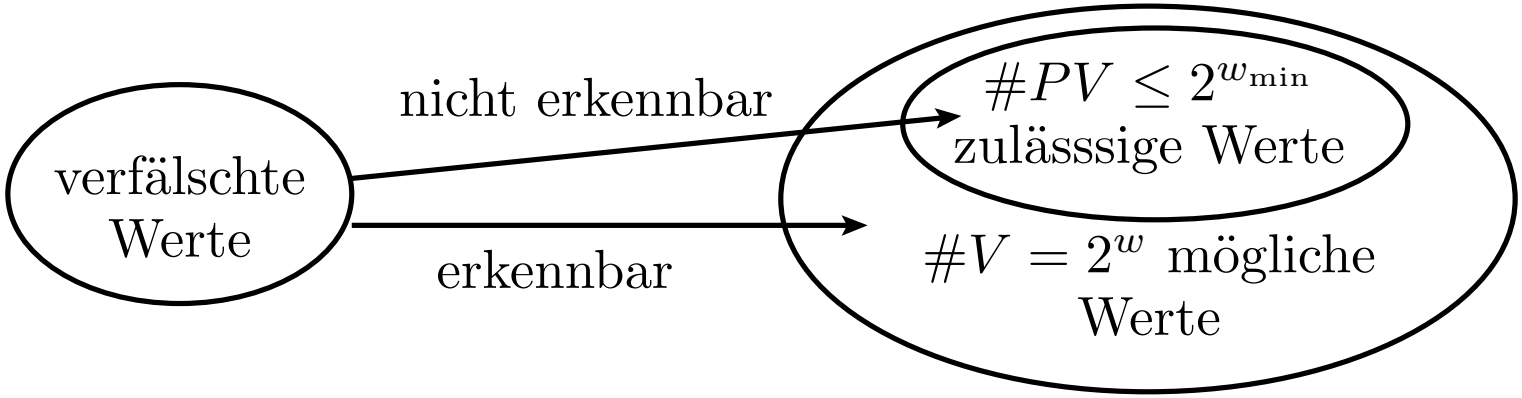
$$MC = 1 - \frac{\#PV}{\#V}$$

MC	Fehlfunktionsüberdeckung (**m**alfunction **c**overage), Anteil der nachweisbaren FF.
$\#PV$	Anzahl der zulässigen Werte (number of **p**ermitted **v**alues).
$\#V$	Anzahl der möglichen Werte (number of possible **v**alues).

Redundante Bits

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

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



$$1 - MC = \frac{\#PV}{\#V} \geq \frac{2^{w_{\min}}}{2^{r+w_{\min}}} = 2^{-r}$$

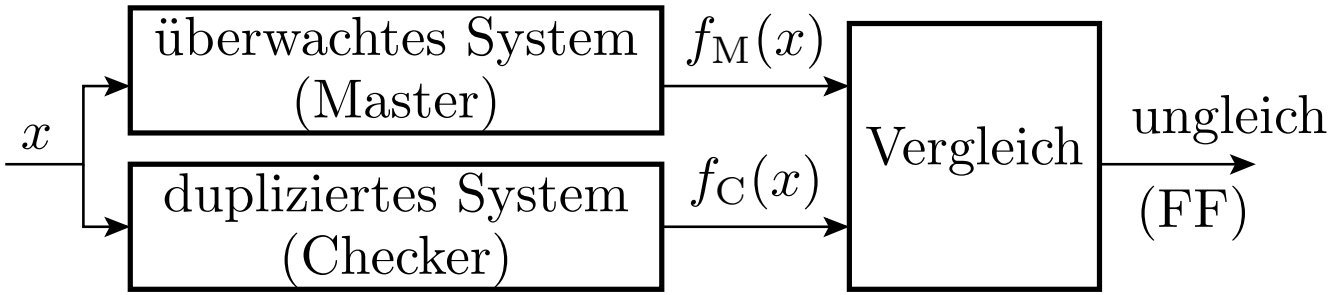
$$MC \geq 1 - 2^{-r}$$

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

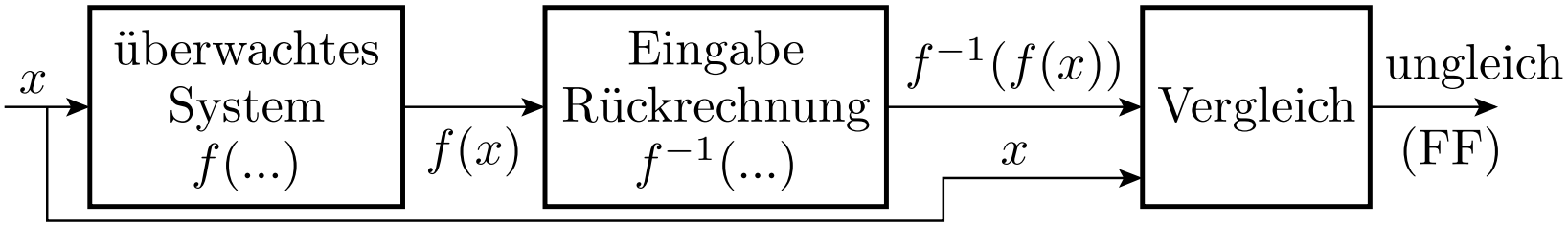
Bei angenommen $w_{\min} = 10^3$ kein nennenswerter Zusatzaufwand.

Verfahren zur Werteüberwachung

- Verdopplung und Vergleich:

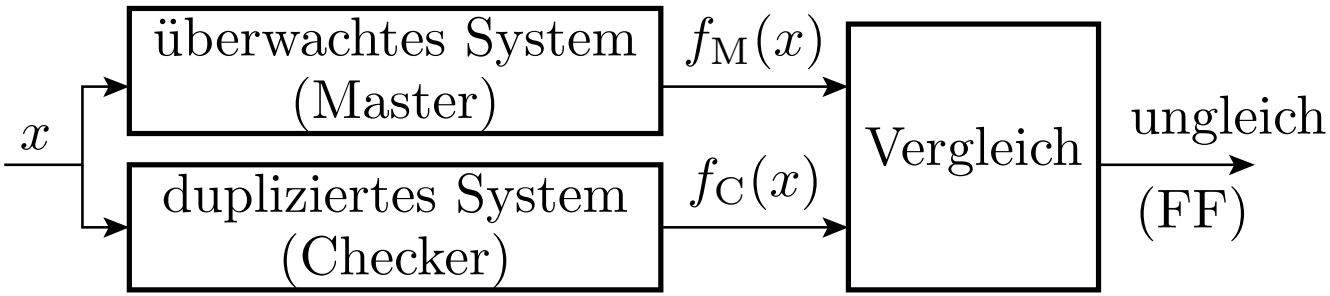


- Eingaberückberechnung und Vergleich, z.B. Überwachung Versenden durch Empfang und Vergleich der empfangenen mit den Sendedaten:



- Datenkorrektheitstest, z.B. für Suche Weg von A nach B durch einen Graphen, Kontrolle »gefundener Weg führt von A nach B«.

Eigenschaften »Verdopplung und Vergleich«



Die FF-Überdeckung ist der Anteil Master-MFs, für die der Slave keine oder abweichte FF hat:

$$MC = \frac{\#MNE}{\#MFM} \tag{4}$$

Phantom-FF-Rate ist die Rate der FF des Vergleichssystems, bei denen der Master nicht dieselbe FF hat:

$$\zeta_{\mathrm{Phan}} \approx \zeta_{\mathrm{Chk}} \cdot (1 - MC)$$

MC	Fehlfunktionsüberdeckung (**m**alfunction **c**overage), Anteil der nachweisbaren FF.
$\#MM$	Anzahl der Master-FF (number of **master** MFs).
$\#MNE$	Anzahl der Master-FF ungleich Checker-SL (number of master **M**Fs **n**ot **e**qual to checker).
ζ_{Chk}	FF-Rate Checker (MF rate **ch**eck**k**er).

Diversität

Die Diversitätsrate η_{Div} zwischen zwei Systemen sei der Anteil der entstehenden Probleme (FF, Ausfälle, ...) ohne gemeinsame Ursache:

$$\eta_{\mathrm{Div}} = 1 - \eta_{\mathrm{CC}}$$

Unter der Annahme sehr vieler Möglichkeiten für Verfälschungen (siehe Formatkontrollen unter Ausnutzung von Informationsredundanz) erkennt Verdopplung und Vergleich fast alle FF ohne übereinstimmende Ursache und keine FF mit gleicher Ursache:

$$MC = \eta_{\mathrm{Div}}$$

FF-Diversität bei Mehrfachberechnung setzt sich zusammen aus

- Grunddiversität: Anteil der FF durch Störungen und Fehler mit zufälliger Wirkung und
- Erweiterter Diversitäten, die durch konstruktive und organisatorische Maßnahmen geschaffen wird.

η_{Div} Diversitätsrate (diversity rate), Anteil der Probleme ohne gemeinsame Ursache.

η_{CC} Anteil Probleme mit gemeinsamer Ursache (common cause rate) .

Erweiterte Diversität

Erweiterete Diversität	konstr. und org. Maßnahmen	zusätzlich erkennbare FF
HW-Diversität	Ausführung auf verschiedener HW	Fertigungsfehler, Ausfälle
HW-Entwurfsdiversität	unabhängig entworfene HW	HW-Entwurfsfehler
Syntaktische Diversität	unterschiedlich übersetzte SW	SW-Übersetzungsfehler
Software-Diversität	unabhängig entworfene SW	SW-Entwurfsfehler
diversitäte Systemnutzung*	unterschiedliche Anforderung	auch identische Fehler

*Aufgaben lassen sich oft mit vielen Arten der Service-Anforderung lösen. Nutzer lernen mit der Zeit, wo, wann und wie FF auftreten und passen das Nutzungsverhalten an. Für Überwachung durch Verdopplung und Vergleich ungeeignet, da oft abweichende Soll-Ausgaben.

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, ...

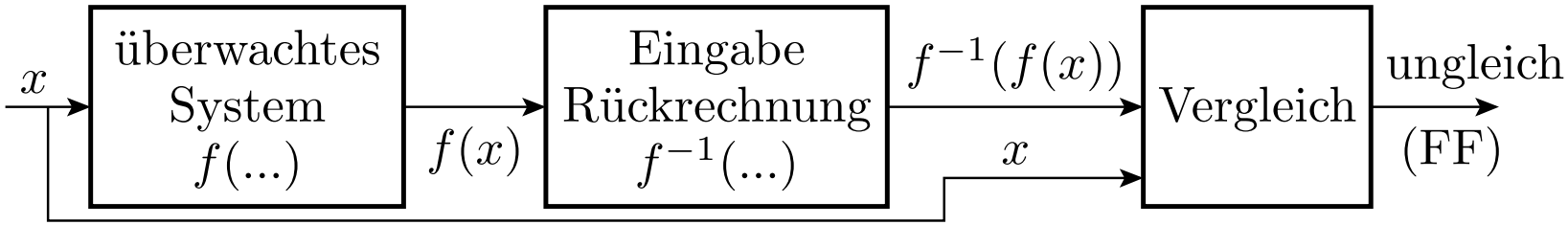
Ursprüngliche euphorische Meinung, dass so Diversität gegenüber allen Fehlern, außer denen in der Spezifikation erzielbar sei, nicht bestätigt. Die direkte oder indirekte Kommunikation der Entwicklungsteams über die Interpretation der Spezifikation, während des Test etc. trägt Gemeinsamkeiten in die Entwürfe. Neigung von Menschen, gewisse Fehler zu wiederholen, ... Erzielbare Divisität[3]

$$\eta_{\mathrm{Div}} = MC \leq 90\%$$

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

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

Eigenschaften »Eingaberückberechnung«



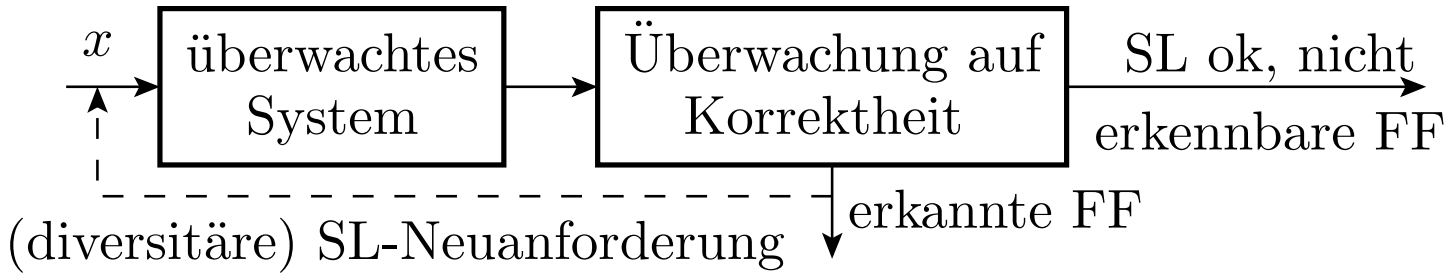
Da $f(\ldots)$ und $f^{-1}(\ldots)$ sich in Algorithmus und Fehlerwirkung unterscheiden, ist eine höhere Diversität als bei Verdopplung und Vergleich zu erwarten und eine Fehlfunktionsüberdeckung

$$MC \gg \eta_{\mathrm{Div}}$$

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.

MC	Fehlfunktionsüberdeckung (**m**alfunction **c**overage), Anteil der nachweisbaren FF.
η_{Div}	Diversitätsrate (diversity rate), Anteil der Probleme ohne gemeinsame Ursache.

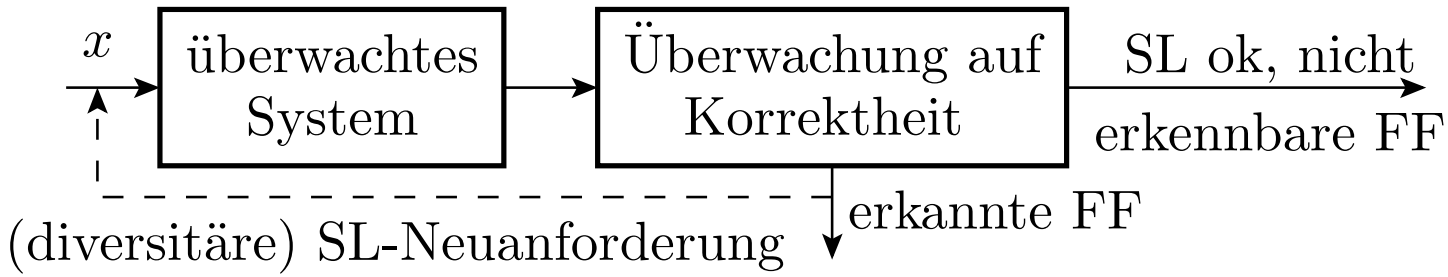
Datenkorrektheitstest



Beispiele:

- Suche Weg durch einen Graphen $\Rightarrow$ zulässiger Weg.
- Suche Test für Fehlernachweis $\Rightarrow$ Fehlersimulation.
- Sortieren einer Liste $\Rightarrow$ Liste sortiert und enthält alle Elemente.

Die Fehlfunktionsüberdeckung MC ist die der Überwachung. Oft sehr hoch, aber für die meisten Zielfunktionen existiert keine solche Kontrollmöglichkeit.



Achtung, für die typische Form von Suchalgorithmen

```
Probiere, bis Kontrolle bestanden
    Errate das Ergebnis
```

strebt die FF-Rate ζ_{NMT} ohne Aussortieren erkannter FF gegen 1 und mit aussortieren gegen:

$$\zeta = \lim_{\zeta_{\mathrm{NMT}} \to \infty} (\zeta_{\mathrm{NMT}} \cdot (1 - MC)) = 1 - MC$$

ζ	Fehlfunktionsrate (malfunction rate) im Einsatz.
ζ_{NMT}	FF-Rate ohne FF-Behandlung (MF rate with **no** **m**ulfunction **t**reatment).
MC	Fehlfunktionsüberdeckung (**m**alfunction **c**overage), Anteil der nachweisbaren FF.

Robuste Reaktion auf FF

Robuste Reaktion

Robuste Reaktion auf eine FF: Kontrolliertes Verhalten zur Vermeidung von unkontrollierbaren Schaden, insbesondere sicherheitsggefährdender FF:

- Keine Weiterverarbeitung fehlerhafter SL, Fehlermeldung.
- Für gesteuerte Maschinen oder Prozesse Übergang in einen sicherer Zustand,
- Daten sichern,
- Beseitigung der Entstehungsursache der FF.

Ursachenbeseitigung:

FF-Ursache	Störung	Ausfall	Fehler
Beseitigung	NI	Reparatur* + NI	Beseitigung** + NI

NI Neuinitialisierung

* Reparatur im laufenden Betrieb verlangt bereitstehende Reserveeinheiten (siehe F7.2 *Ausfall-Toleranz*).

** Fehlerbeseitigung nicht im laufenden Betrieb, sondern im Reifeprozess über eine lange Nutzungsdauer (siehe F1.2.7 *Reifeprozess*).

Sicherheitsverbesserung

Wir unterstellen, dass

- bei robuster Reaktion von sonst gefährdenden FF keine Gefahr mehr ausgeht und
- auf alle erkannten FF robust reagiert wird.

Damit verringert sich die Rate der gefährdenden FF auf den Anteil der nicht erkannten FF:

$$\zeta_{\mathrm{S}} = \zeta_{\mathrm{SNMT}} \cdot (1 - MC)$$

Sicherheitsverbesserung durch FF-Behandlung:

$$S = \frac{S_{\mathrm{NMT}}}{1 - MC}$$

ζ_{S}	Rate der sicherheitsgefährdenden FF (rate of **s**afety endangering MF).
ζ_{SNMT}	Rate der sicherheitsgefährdenden MF ohne FF-Behandlung (rate of **s**afety endangering MF with **n**o **m**alfunction **t**reatment).
MC	Fehlfunktionsüberdeckung (**m**alfunction **c**overage), Anteil der nachweisbaren FF.
S	Sicherheit (**s**afety (security)).
S_{NMT}	Sicherheit ohne Fehlfuntionsbehandlung (**s**afety with **n**o **m**alfunction **t**reatment).

Beispiel 5 (Sicherheitserhöhung durch FF-Behandlung)

Ein IT-System habe eine FF-Rate ohne FF-Behandlung von $\zeta_{\mathrm{NFT}} = 10^{-3}\ \left[\frac{\mathrm{FF}}{\mathrm{SL}}\right]$ mit einem Anteil gefährdende FF von $\eta_{\mathrm{SE}} = 2\%\ \left[\frac{\mathrm{GFF}}{\mathrm{FF}}\right]$. Wie groß müssen die FF-Überdeckung der Überwachung mindestens sein, damit die Sicherheit mindestens $S \geq 10^6\ \left[\frac{\mathrm{SL}}{\mathrm{GFF}}\right]$ beträgt?

$$S = \frac{S_{\mathrm{NFT}}}{1 - MC} \quad \text{mit } S_{\mathrm{NMT}} = \frac{1}{\zeta_{\mathrm{NMT}} \cdot \eta_{\mathrm{SE}}}$$

$$MC = 1 - \frac{S_{\mathrm{NFT}}}{S} = 1 - \frac{1}{S \cdot \zeta_{\mathrm{NFT}} \cdot \eta_{\mathrm{SE}}}$$

$$\geq 1 - \frac{1}{10^6\ \left[\frac{\mathrm{SR}}{\mathrm{DM}}\right] \cdot 10^{-3}\ \left[\frac{\mathrm{MF}}{\mathrm{SR}}\right] \cdot 2 \cdot 10^{-2}\ \left[\frac{\mathrm{DM}}{\mathrm{MF}}\right]} = 95\%$$

Die FF-Überdeckung MC muss auch mindestens 95% betragen, vorausgesetzt, das System reagiert robust, d.h. ohne Sicherheitsgefährdung auf alle erkannten MF.

FF – Fehlfunktion, GFF – gefährdente FF; SL – erbrachte Service-Leistung.

FF-Toleranz

FF-Toleranz

Von lateinisch tolerare »erleiden«, »erdulden«. In der Technik, aufrechterhalten der Funktion bei internen FF durch Eingabefehler, Störungen, Fehler und Ausfälle.

Einteilung der Reaktionen auf FF:

1. fail-unsafe: unvorhersehbares Systemverhalten.
2. fail-safe: Nur Systemsicherheit gewährleistet.
3. fail-soft: Systembetrieb sicher, aber Leistung vermindert.
4. fail-operational: Verbleib in einem betriebsfähigem Zustand.
5. go: System reagiert sicher und korrekt.

Ab 2 robuste Reaktion mit Sicherheitsverbesserrung.

Reaktion 5 verringert die Anzahl der nach außen sichtbaren FF.

Die Reaktionen 3 und 4 sind robuste Reaktionen mit unterschiedlicher Systemleistung nach Ausfällen.

Zuverlässigkeit und Fehlfunktionstoleranz

Zuverlässigkeit nach unserer Definition wird nur durch Reaktion »go« auf erkannte FF verbessert. Die Verbesserung soll durch die FF-Toleranzrate als Anteil der korrigierten (tolerierten) FF beschrieben werden:

$$\eta_{\mathrm{Tol}} \approx \frac{\#Tol}{\#FF} \tag{5}$$

Die FF-Toleranzrate ist maximal so groß wie die FF-Überdeckung

$$\eta_{\mathrm{Tol}} \leq MC$$

Die FF-Rate verringert sich um $1 - \eta_{\mathrm{Tol}}$ auf:

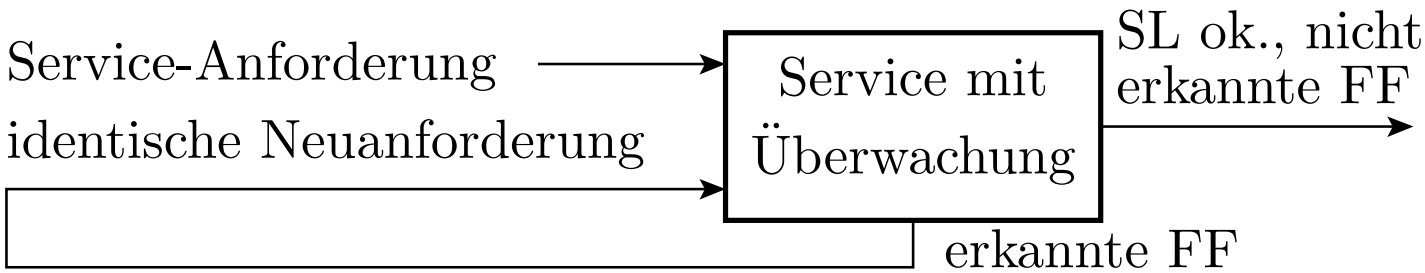
$$\zeta = \zeta_{\mathrm{NMT}} \cdot (1 - \eta_{\mathrm{Tol}})$$

und die Zuverlässigkeit als Kehrwert der FF-Rate erhöht sich auf:

$$R = \frac{R_{\mathrm{NMT}}}{1 - \eta_{\mathrm{Tol}}}$$

η_{Tol}	FF-Toleranzrate (MF **tol**erance rate), Anteil der tolerierten FF.
$\#Tol$	Anzahl der tolerierten FF (number of **tol**erated (corrected) MF).
$\#MF$	Anzahl der Fehlfunktionen (number of **m**al**f**unctions).

Korrektur durch Wiederholung



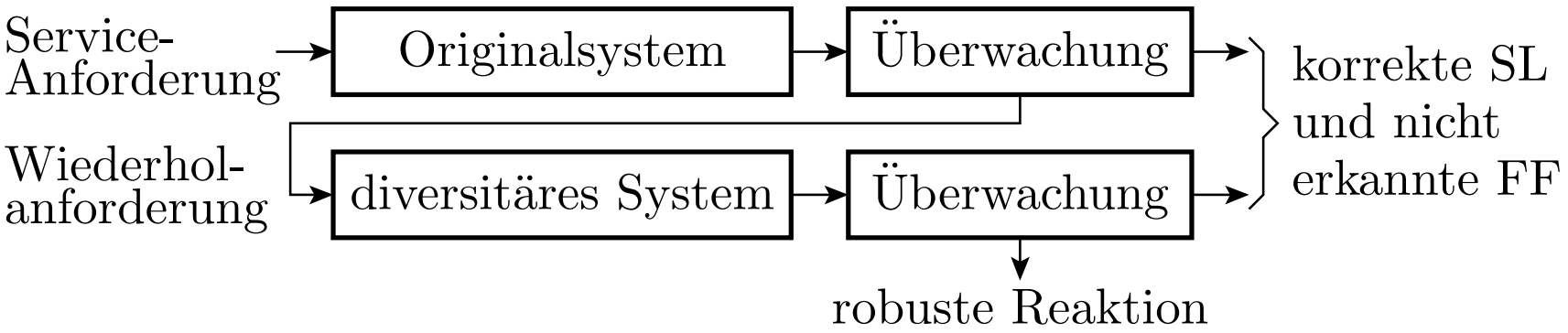
Der Anteil der tolerierten (korrigierten) FF ist die Diversität η_{Div} zwischen Erst- und Wiederholberechnung multipliziert mit der MC der Überwachung:

$$\eta_{\mathrm{Tol}} \approx MC \cdot \eta_{\mathrm{Div}}$$

Bei identischer Wiederholung besteht nur Grundiversität, d.h. nur Korrektur von FF durch Störungen und zufällig wirkende Fehler.
Typische Anwendungen:

- Übertragung über störanfällige Kanäle (Funk, Modem, ...),
- Lesen von Massenspeichern, ...

Wiederholung mit diversitärem System



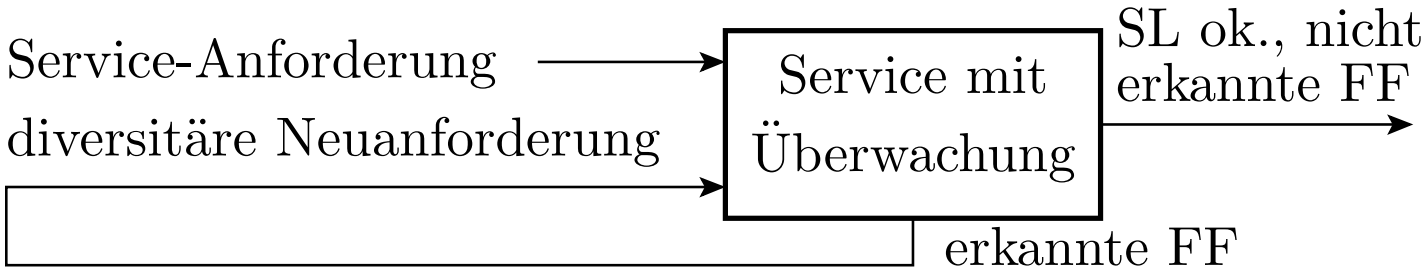
Bei FF identische Anforderung an ein anderes System, das einen gleichwertigen Service anbietet. Die erzielbare FF-Toleranz ist wieder

$$FT \approx MC \cdot Div$$

Gegenüber »Verdopplung und Vergleich« mit unterschiedlichen Systemen höhere Diversität möglich, weil die beiden Systeme im fehlerfreien Fall keine übereinstimmenden Ausgaben liefern müssen.

Alternative: Den Mehraufwand für Entwicklung und Test für diversitäres Zweitsystem in gründlicheren Test des Originalsystems zu investieren.

Fehlfunktionsvermeidung

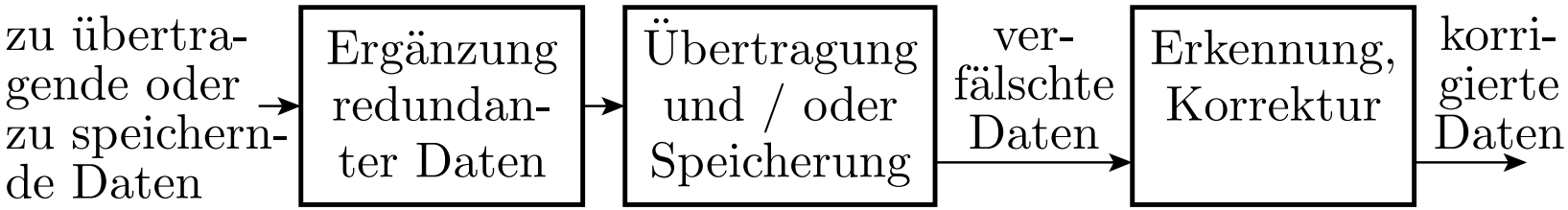


Komplexe IT-Systeme besitzen in der Regel eine große funktionale Redundanz, so dass sich die meisten Aufgaben auf unterschiedlichen Wegen lösen lassen, z.B. über graphische oder Konsolenbedienung. Nutzer lernen in einer Iteration aus Probieren und Suche, was zuverlässig funktioniert und passen ihr Nutzungsverhalten an.

FF-Vermeidung ist eine Anpassung des Nutzerverhaltens so, dass vorhandene Fehler keine / selten FF verursachen. Wenn identische Wiederholung dieselbe FF erzeugt, Umgehung mit einer anderen Service-Anforderung, die dieselbe Aufgabe löst.

Fehlerumgehung ist kaum ohne Nutzerinteraktion realisierbar.

Fehlerkorrigierende Codes



- Einsatz für die Korrektur von Einzelbit- und Burst-Fehlern nach Datenübertragung und Speicherung, auch für RAIDs (siehe F7.2.4 *RAD und Backup*).
- Höhere Datenredundanz als fehlererkennende Codes.
- Gute Lösung für die Korrektur gespeicherter oder empfangener Daten. Für andere Arten von SL ungeeignet.
- Dort wo anwendbar, ist die Fehlertoleranz FT als der Anteil der korrigierbaren Datenverfälschungen sehr hoch und die Phantom-MF-Rate niedrig.

Zuverlässigkeit mit MF-Behandlung

Die »kostengünstigen« Korrekturverfahren »Wiederholung« und »FF-korrigierenden Codes« korrigieren grob überschlagen

- alle FF durch Störungen aber
- kaum solche, die durch Fehler verursacht werden.

FF-Rate und Zuverlässigkeit im Einsatz:

$$\zeta = \zeta_{\mathrm{F}}; \quad Z = Z_{\mathrm{F}}$$

- Diversitäre Neuberechnung (mit gleichartigen Systemen aus unabhängigen Entstehungsprozessen) und
- FF-Umgehung durch durch Anpassung des Nutzerverhaltens

erlauben mit erheblichen zusätzlichen Entwurfs- oder Nutzeraufwand eine weitere Zuverlässigkeitserhöhung um ca. eine Zehnerpotenz.

ζ	Fehlfunktionsrate (malfunction rate) im Einsatz.
ζ_{F}	Rate der durch Fehler verursachten FF (MF rate due to faults).
R	Zuverlässigkeit (**r**eliability).
R_{F}	fehlerbezogene Teilzuverlässigkeit (partitial **r**eliability due to **f**aults).

Zusammenfassung

FF-Behandlung

Kenngrößen der Überwachung:

- Fehlfunktionsüberdeckung MC, Anteil der erkennbaren FF.
- Phantom-FF-Rate ζ_{Phan}, Anteil der Phantom-FF an den SL.

Überwachungsverfahren für digitale SL:

- Formatkontrollen (Zeitschranken, Wertebereich, Prüfsummen, ...).
  - Ausnutzung von Informationsredundanz.
  - Nur Überwachung auf Zulässigkeit.
  - Breite Anwendbarkeit und gutes Aufwand-Nutzen-Verhältnis.
- Datenkontrollen:
  - Verdopplung und Vergleich identische Systeme. Für Störungen [und Ausfälle] gut, aber nicht für Fehler als FF-Ursache.
  - Erweiterte Diversität. Zusätzlich MC für FF durch Fehler bis 90%.
  - Eingaberückberechnung. Gut, wenn anwendbar.
  - Datenkorrektheitstest, Gut, wenn es einen gibt.

Überwachung wird auf Foliensatz 4 ausführlicher behandelt.

Verbesserung der Zuverlässigkeit und Sicherheit

Bei robuster Reaktion auf alle erkannten FF Sicherheitserhöhung:

$$S = \frac{S_{\mathrm{NMT}}}{1 - MC}$$

Geeignete Techniken für die Zuverlässigkeiterhöhung:

- identische Wiederholung mit demselben System und
- fehlerkorrigierende Codes für Speicherung und Übertragung.

Überschlagsweise Korrektur alle FF durch Störungen, aber keine durch Fehler. Die Zuverlässigkeit erhöht sich auf die fehlerbezogene Teilzuverlässigkeit:

$$R = R_{\mathrm{F}}$$

S	Sicherheit (**s**afety (security)).
S_{NMT}	Sicherheit ohne Fehlfuntionsbehandlung (**s**afety with **n**o **m**alfunction **t**reatment).
η_{SE}	Anteil der sicherheitsgefährdenden FF (percentage of **s**afety **e**ndagering MF).
MC	Fehlfunktionsüberdeckung (**m**alfunction **c**overage), Anteil der nachweisbaren FF.
R	Zuverlässigkeit (**r**eliability).
R_{F}	fehlerbezogene Teilzuverlässigkeit (partitial **r**eliability due to **f**aults).

Fehlerbeseitigung

Gefährdungen und Gefährdungsabwendung

- Störungen:
  - Zufällige, nicht reproduzierbare Ursache-Wirkungs-Beziehungen,
  - Gefährdungsabwendung: Überwachung und Korrektur, in der Regel durch identische Wiederholung.

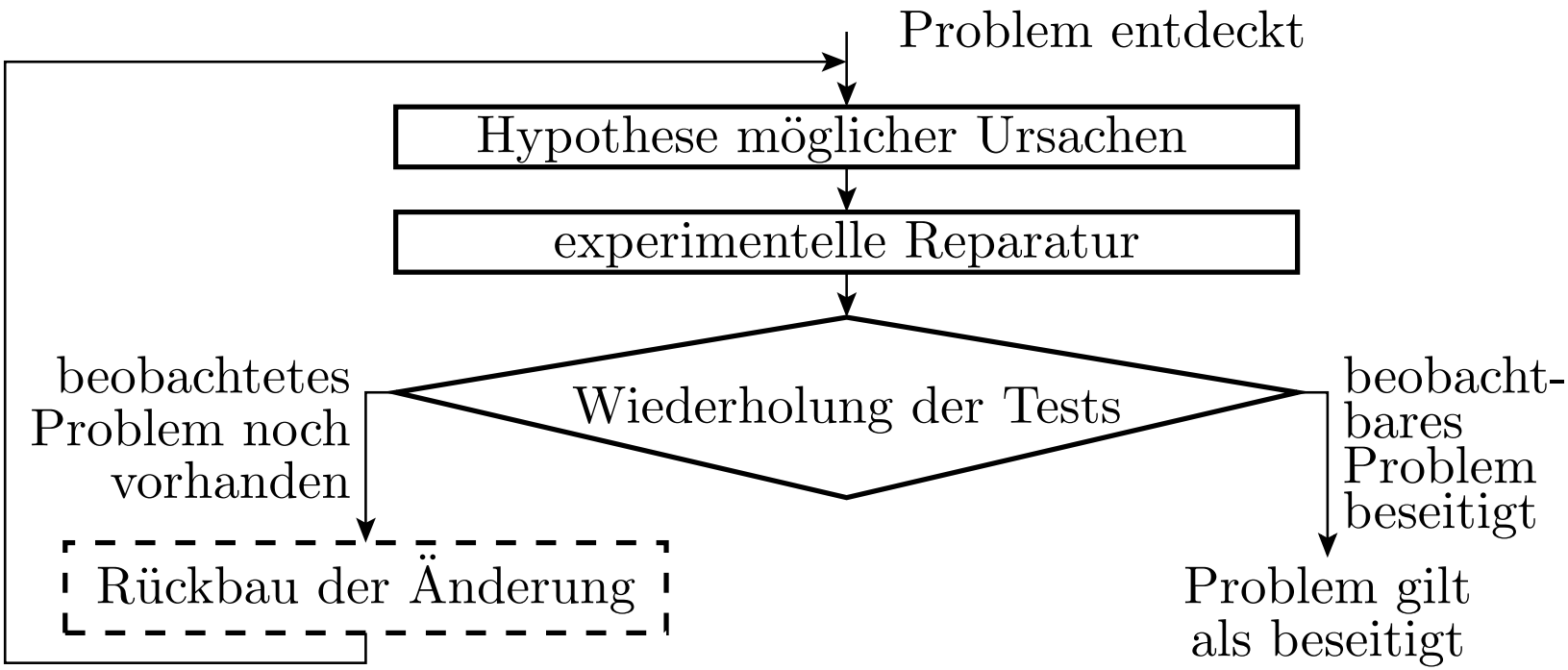
- Fehler:
  - Entstehen mit dem System oder bei der Fehlerbeseitigung,
  - Gefährdungsabwendung: Fehlerbeseitigung, Fehlervermeidung.

- Ausfälle:
  - während des Betriebs entstehende Fehler,
  - Gefährdungsabwendung durch FF-Behandlung, Wartungstest und Redundanzen (siehe F7.2 *Ausfall-Toleranz*).

Für Störungen erfolgt die Gefährdungsabwendung vorzugsweise durch »FF-Behandlung« und für Fehler durch »Fehlerbeseitigung«.

Fehlerbeseitigungsiteration

Fehlerbeseitigungsiteration

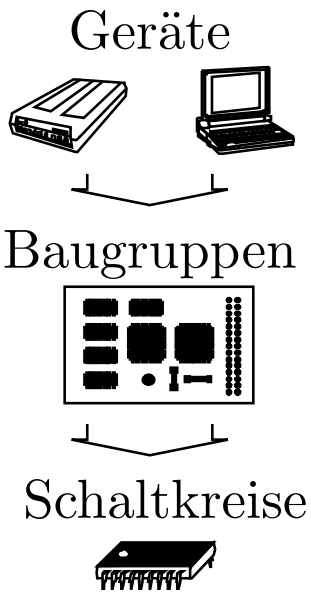


- Iteration aus Beseitigungsversuchen für hypothetische Fehler und Erfolgskontrolle durch Testwiederholung.
- Beseitigt alle vom Test nachweisbaren Fehler.
- Zur Vermeidung der Entstehung neuer Fehler bei der Reparatur Rückbau nach erfolglosen Reparaturversuchen.

Reparatur bei wenig tauschbaren Komponenten

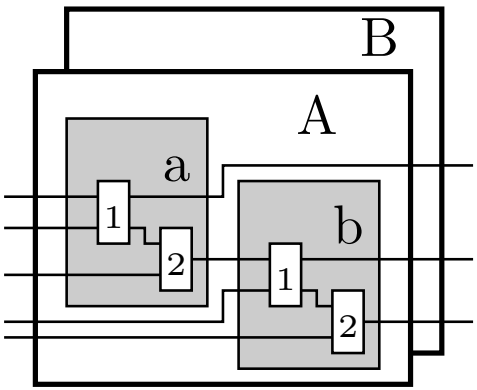
Ein reparaturgerechtes System hat eine hierarchische Struktur aus tauschbaren Komponenten, z.B.

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

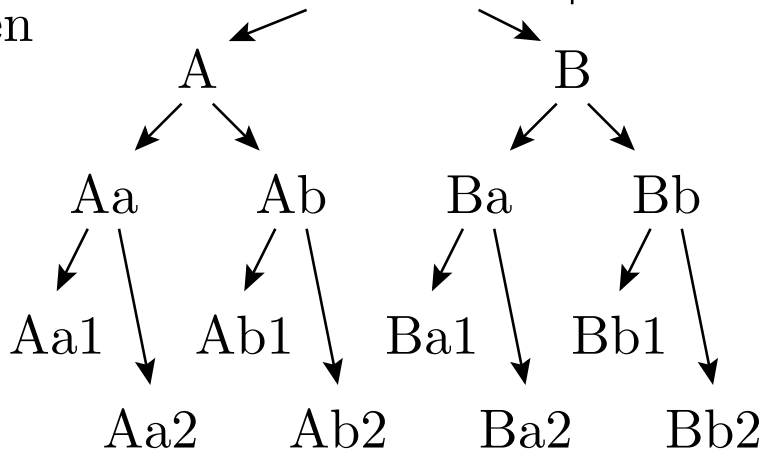


Fehlerlokalisierung durch systematisches Tauschen:

hierarchisches System mit tauschbaren Komponenten



Tauschbaum



Übliches Vorgehen eines Reperateurs

- Grobabschätzung, welches Rechnerteil defekt sein könnte aus den Fehlersymptomen.
- Kontrolle der Steckverbinder auf Kontaktprobleme durch Abziehen, Reinigen, Zusammenstecken, Testwiederholung.
- Ersatz möglicherweise defekter Teile durch Ersatzteile, Testwiederholung, ...

Voraussetzungen:

- Wiederholbare Tests, die den Fehler nachweisen.
- Ausreichend Ersatzteile. Allgemeine Mechnikerkenntnisse*.

Ist der Rücktausch nach erfolglosem Ersatzteileinbau notwendig? Wenn ja, warum?

Günstig ist der Tausch der Hälfte, von der fehlerhaften Hälfte auch die Hälfte, ... Warum?

* Verständnis der Funktion des zu reparierenden Systems nicht zwingend.

Fehlerdiagnose

Fehlerdiagnose

Abschätzung von Ort-, Ursache und Beseitigungsmöglichkeiten von Fehlern aus Testergebnissen.

Eine gute Fehlerdiagnose vor jedem Reparaturversuch mindert

- die Anzahl der Reparaturversuche,
- den Bedarf an Ersatzteilen,
- die Anzahl der bei Reparaturversuchen entstehenden Fehler
- inc. der, die nicht durch Rückbau beseitigt werden.

Das Optimum für Diagnose- plus Beseitigungskosten liegt bei im Mittel wenig mehr als ein Reparaturversuch je Fehler.

Ohne Möglichkeit für »systematisches Tauschen« wie für SW- und HW-Entwurfsfehler gibt es kaum Erfolgschancen ohne ausreichende zielführende Fehlerdiagnose.

Allgemeine Diagnosetechniken:

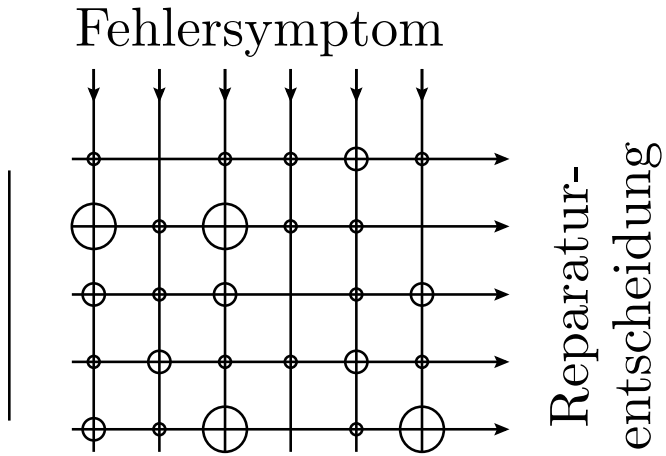
- erfolgsorientiertes Tauschen und
- Rückverfolgung von Verfälschungen gegen den Daten- oder Berechnungfluss.

Erfolgsorientiertes Tauschen

Produkte haben Schwachstellen. Die meisten Probleme geht auf einen kleinen Anteil der möglichen Ursachen zurück, Pareto-Prinzip:

- Zählen der Erfolge unterschiedlicher Reparaturalternativen.
- Bei Reparatur, Beginn mit den erfolgsversprechendsten Möglichkeit.

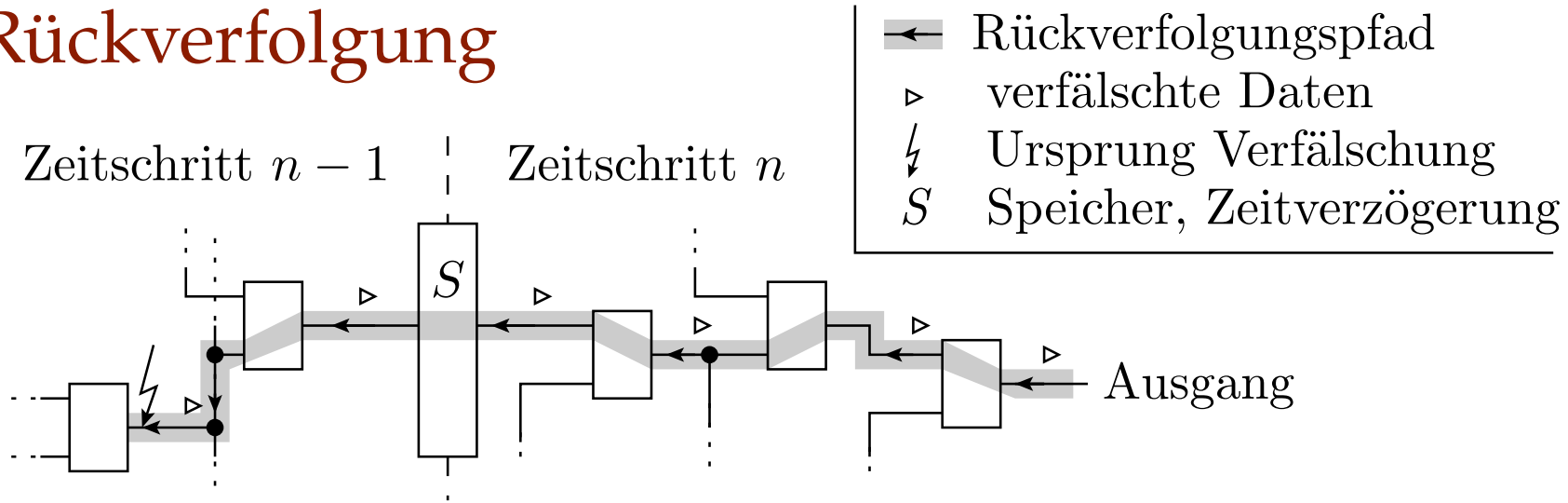
◎ Häufigkeit, mit der die Reparaturentscheidung für das Symstem bisher erfolgreich war



Nach erfolglosen Reparaturversuchen Rückbau der Änderung zur Minderung der Fehlerentstehungsrate bei der Reparatur.

Der italienische Ökonom Vilfredo Pareto untersuchte 1906 die Verteilung des Grundbesitzes in Italien und fand heraus, dass ca. 20 % der Bevölkerung ca. 80 % des Bodens besitzen. Das ist in den Sprachgebrauch als Pareto-20%-80%-Regel eingegangen.

Rückverfolgung



Ausgehend von einer erkannten falschen Ausgabe Rückverfolgung entgegen Berechnungs- bzw. Signalfluss bis zu der Komponente, die richtige Eingaben auf verfälschte Ausgaben abbildet, gegebenenfalls über Zeitschritte und/oder hierarchisch absteigend.

Quelle der Verfälschung kann außer der gefundenen Komponente bei HW z.B. auch ein Kurzschluss oder bei SW ein fehlgeleiteter Schreibzugriff sein.

Reparatur- und prüfgerechter Entwurf

Sammlungen von

- Regeln »of good practise«, zur Ermöglichung / Vereinfachung von Test, Fehlerlokalisierung und Reparatur und
- Antipattern, die die Arbeiten erheblich erschweren.

Einige Regeln »of good practise«:

- Modulares System aus tauschbaren / seperat testbaren Funktionsblöcken.
- Deterministisches Verhalten mit gerichtetem Berechnungsfluss.
- FF-Isolation zur Verhinderung der Ausbreitung von MF über Modulgrenzen.
- Beobacht- und Steuerbarkeit (wichtiger) interner Werte.

Standarbeispiel für ein Antipattern:

- »Big ball of mud«: großes, unstrukturiertes, mangelhaft dokumentiertes System, das niemand mehr richtig versteht.

Die Vorlesung unterstellt einen reparatur- und prüfgerechten Entwurf.

Test

Testen

Verfahren zum Aufspühren von Fehlern. Grundeinteilung:

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

Mit statischen Tests kontrollierbare Merkmale:

- Dokumentationen: Verständlichkeit, Vollständigkeit, ...
- Software: Syntax, Entwurfsregeln, Typenverträglichkeit und API-Benutzerregeln,
- Leiterplatten: keine Kurzschlüsse und Unterbrechungen, ...

Statische Tests sind bereits während der ersten Entwurfsschritte und während der Fertigung möglich, dynamische Tests erst am fertigen Produkt.

Vor dem Einsatz werden die Systeme in der Regel verschiedenen statischen und dynamischen Tests unterzogen.

Kenngrößen von Tests

Wie bei jeder Kontrolle mit den möglichen Ergebnissen gut oder schlecht sind zwei Arten von Fehlklassifikationen möglich:

- Nichterkennbare Fehler. Modelliert durch die Kenngröße Fehlerüberdeckung (Fault Coverage,):

$$FC = \frac{\#F_{\mathrm{D}}}{\#F} \tag{6}$$

- Phantomfehler. Tests, die korrekte Testergebnisse als falsch klassifizieren. Modelliert durch die Phantom-FF-Rate des Tests:

$$\zeta_{\mathrm{PhanT}} = \frac{\#PF}{n} \tag{7}$$

FC	Fehlerüberdeckung (**f**ault **c**overage), Anteil der nachweisbaren Fehler.
$\#F_{\mathrm{D}}$	Anzahl der erkennbaren Fehler (number of **d**etectable **f**aults).
$\#F$	Anzahl der Fehler (number of **f**aults).
ζ_{PhanT}	Phantom-FF-Rate des Tests (**phan**tom MF rate during **t**est).
$\#PM$	Anzahl der Phantom-FF (number of **p**hantom **m**alfunction).
n	Anzahl der Tests (number of tests).

Phantomfehler und Zuverlässigkeit

Eine Phantomfehler (z.B. eine FF beim der Testauswertung)

- startet eine überflüssige Beseitigungsiteration
- in der ein neuer nicht nachweisbarer Fehler entstehen kann.

Kontrolle Testergebnisse meist durch Vergleich mit Sollwerten:

- Maskierungen von Fehlern durch Vergleichs-FF und
- Phantomfehler durch falsche Sollwerte, ...

Für neu entwickelte Tests ist zu kontrollieren, dass

- richtige Testergebnisse als richtig und
- falsche Testergebnisse als falsch klassifiziert werden.

Wenn ein Test einen Fehler erkennt, Kontrolle dass keine Phantomfehler ist.

Bei vernünftigem Umgang mit Phantomfehlern ist deren Einfluss auf die Gesamtanzahl der entstehenden Fehler unerheblich. Wir werden Phantomfehler in später entwickelten Modellen vernachlässigen.

Testauswahl für dynamische Tests

Dynamische Tests kontrollieren die Funktion praktisch immer nur für eine winzige Stichprobe der möglichen Eingaben. Die FC hängt vom Umfang und der Auswahl der Testbeipiele ab.

Strategien der Testauswahl:

- fehlerorientiert.
- zufällig hinsichtlich der zu erwartenden Fehler oder
- Mischform.

Zum Zeitpunkt der Testauswahl sind die zu findenden und nach dem Test die nicht gefundenen Fehler unbekannt[4].

Ohne Kenntnis der zu findenden Fehler:

- erfolgt die fehlerorientierte Auswahl und Bewertung auf Basis von Fehlerannahmen (Modellfehlern oder Mutationen) und
- ist der Nachweis der tatsächlichen Fehler Zufall.

[4] Ausnahme: Testsuche für beobachtete Fehlverhalten. Siehe später Abschn. F1.4.6 Reifeprozesse.

Vorlesung 3: Zusammenfassung Vorlesung 2

Fehlervermeidung	Fehlerbeseitigung	FF-Behandlung
Beseitigung von Fehler-entstehungsursachen	Test und Beseitigung erkannter Fehler	Überwachung, robuste R. Fehlertoleranz Störungen

Überwachung:

- Überwachung: nur Format oder auch Werte.
- Formatkontrollen bevorzugt, da einfacher und hohe FFC durch Ausnutzung von Informationsredundanz möglich.

Reaktion auf erkannte SL:

- mind. robust (kontrolliertes Verhalten). Sicherheitserhöhung:

$$S = \frac{S_{\mathrm{NMT}}}{1-MC}$$

- einfache Wiederholung, Zuverlässigkeiterhöhung:

$$R = R_{\mathrm{F}}$$

Fehlervermeidung	Fehlerbeseitigung	FF-Behandlung
Beseitigung von Fehlerentstehungsursachen	Test und Beseitigung erkannter Fehler	Überwachung, robuste R. Fehlertoleranz Störungen

Fehlerbeseitigung zur Vermeidung von FF durch Fehler

- beseitigt aller erkennbaren Fehler:

$$\#F = \#F_{\mathrm{CR}} \cdot (1 - FC)$$

- Minderung $\#F_{\mathrm{Rep}}$: Fehlerdiagnose, Rückbau nach erfolglosen Reparaturversuchen, reparatur- und prüfgerechten Entwurf.

Test

- Statische Tests: direkte Kontrolle von Merkmalen.
- Dynamische Tests: Ausprobieren mit Beispieleingaben.
- Auswahl der Testbeispiele für angenommene Fehler oder zufällig.
- Auch bei Fehlerannahmen Nachweis tatsächlicher Fehler Zufall.

Vorlesung 3: Geplante Themen

Abschnitt 4.5 Haftfehler

Fehlerorientierte Testauswahl und Bewertung benötigt Fehlerannahmen. Haftfehler sind das klassische Beispiel.

Abschnitt 4.6 Test und Verlässlichkeit

Tests finden bevorzugt Fehler, die häufig FF verursachen. Mit vereinfachten Modellannahmen wird gezeigt, dass die Zuverlässigkeit tendentiell proportional mit der Anzahl der dynamischen Tests und dem Kehrwert der Anzahl der nicht beseitigten Fehler zunimmt.

Abschnitt 4.7 Reifeprozesse

Durch Einbeziehung der Nutzer als Tester wächst die Zuverlässigkeit mit der Nutzungsdauer mehr als proportional.

Abschnitt 4.8 Modulararer Test

Ein hierarchisch aufsteigender Test erst jeweils Bausteine einzeln, dann Gesamtsystem, ... erhöht auch über die effektive Testanzahl die Zuverlässigkeit.

Haftfehler

Modellfehler und Fehlermodell

Ein Fehlermodell ist ein Algorithmus zur Berechnung einer Menge fehlerhafter Beschreibungen aus einer Entwurfsbeschreibung durch Einfügung von Verfälschungen.

Ein Modellfehler ist ein einzelne dieser eingefügten Verfälschungen.

Bestimmung der FC (Fehlersimulation):

- Wiederhole für jeden Test
  - Bestimmung der Sollausgaben
  - Wiederhole für alle modellierten Fehler der Fehlermenge
    - Bestimme ob der Fehler die Ausgabe verfälscht.
    - wenn ja, kennzeichnen oder aus der Fehlermenge löschen.

Fehlerorientierte Testsuche:

- Wiederhole für alle Modellfehler
  - Suche Eingaben für der der Fehler Ausgaben verfälscht

Beide Aufgaben erfordern einen großen Rechenaufwand. Lösungen für HW siehe Foliensatz F5. Für SW sind Fehlermodelle noch ungebräuchlich, siehe Foliensatz F6..

Das Haftfehlermodell

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

Das Haftfehlermodell generiert für eine Schaltung aus Logikgattern für alle Anschlüsse aller Gatter zwei Modellfehler:

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

Die initiale Fehlermenge wird um identisch nachweisbare, implizit nachweisbare und redundate (nicht nachweisbare) Fehlerannahmen reduziert wird

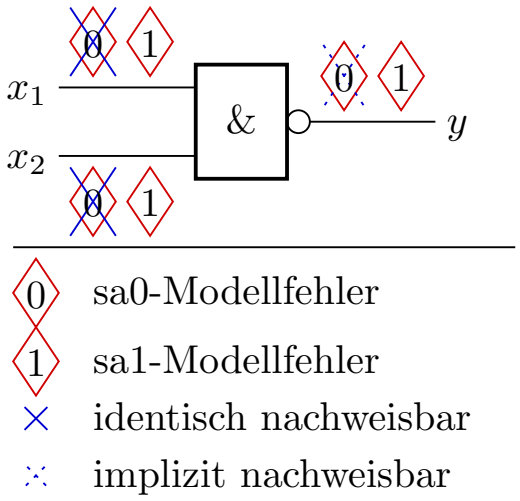
Seit 4 bis 5 Jahrzehnten am weitesten verbreitete Fehlermodell für digitale Schaltkreise.

Bei den sich aktuell entwickelnden Testauswahltechniken für Software lassen sich Parallelitäten zu Haftfehlermodell aufzeigen.

Haftfehler für ein 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

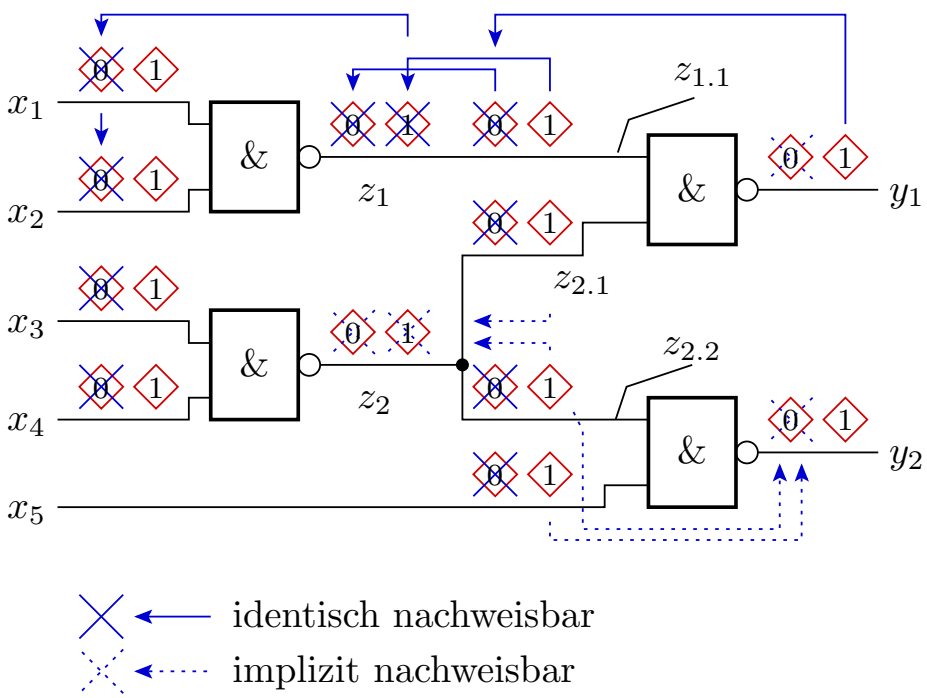
Nachweisidentität (gleiche Nachweismenge)

Nachweisimplikation

zugehörige Eingabe ist Element der Nachweismenge

- Zusammenfassung identisch nachweisbarer Fehler. Optionale Streichung redundanter und implizit nachweisbarer Modellfehler.
- Die generierte Fehlermenge enthält für alle potentiellen Fehler der echten Schaltung ähnlich nachweisbare Modellfehler (siehe F5).
- Software-Mutationen (Off-by-One, Verzweigungsfehler, ...) können auf ähnlich modelliert werden (siehe F6).

Identische und implizit nachweisbarer Fehler im Schaltungsverbund



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

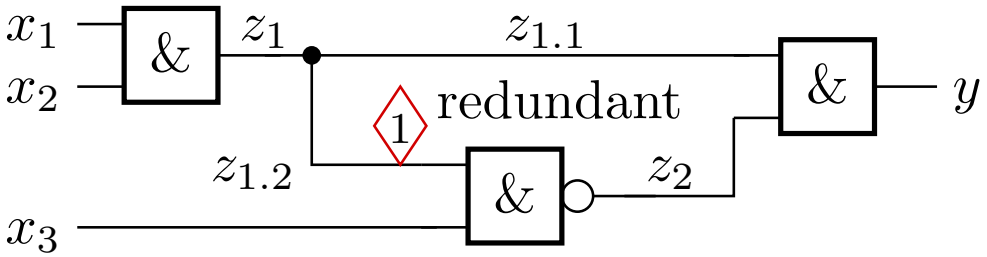
	Mengen von identisch nachweisbaren Fehlern	Nachweis impliziert durch
1	sa0(x_1), sa0(x_2), sa1(z_1), sa1($z_{1.1}$)	
2	sa1(x_1)	
3	sa1(x_2)	
4	sa0(x_3), sa0(x_4), sa1(z_2)	9, 12
5	sa1(x_3)	
6	sa1(x_4)	
7	sa0(z_2)	5, 6, 8, 11
8	sa0(z_1), sa0($z_{1.1}$), sa0($z_{2.1}$), sa1(y_1)	2, 3
9	sa1($z_{2.1}$)	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

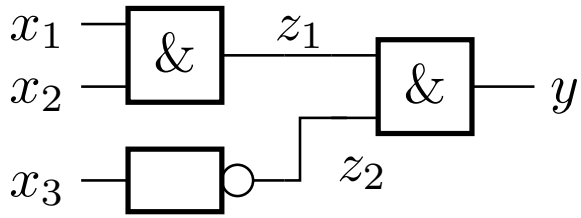
Definition redundanter (Modell-) Fehler

Verfälschung der Systembeschreibung, die die Funktion nicht beeinträchtigt und damit auch nicht mit dynamischen Tests nachweisbar ist.

redundanter Haftfehler



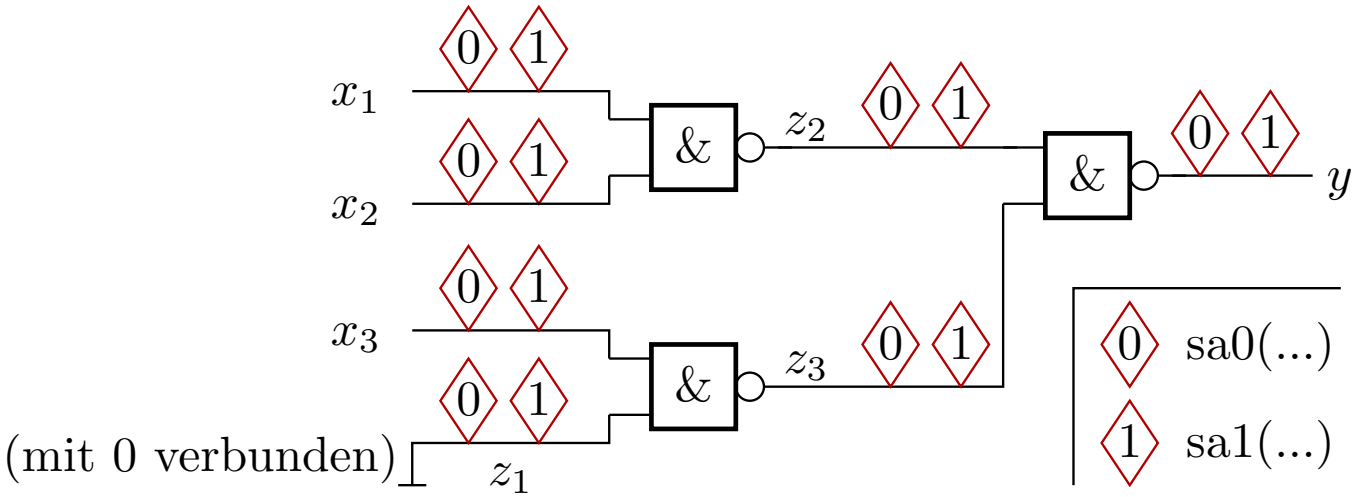
vereinfachte Schaltung



- Die Fehleranregung verlangt $z_1 = 0$ und die Beobachtbarkeit von z_2 an y verlangt $z_2 = 1$. Keine Eingabe $x_3x_2x_1$ kann den Fehler nachweist.
- Umformungen zur Beseitigung redundanter Modellfehler dienen auch zur Systemoptimierung.

Beispielaufgabe

Schaltung mit 12 eingezeichneten Haftfehlern:

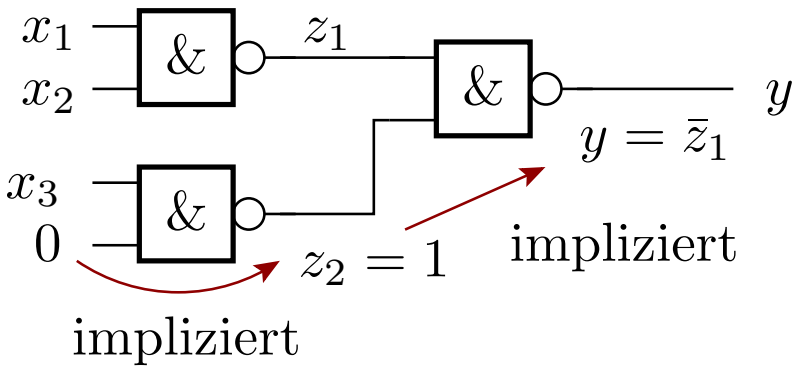


Gesucht:

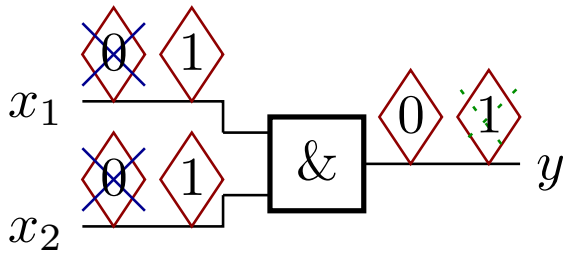
1. Schaltung und initiale Haftfehlermenge nach Beseitigung der Redundanz.
2. Reduzierung der verbleibenden Modellfehlermenge um identisch und implizit nachweisbare Haftfehler.

Lösung

Vereinfachungsmöglichkeiten



Redizierung der Fehlermenge für die vereinfachte Schaltung

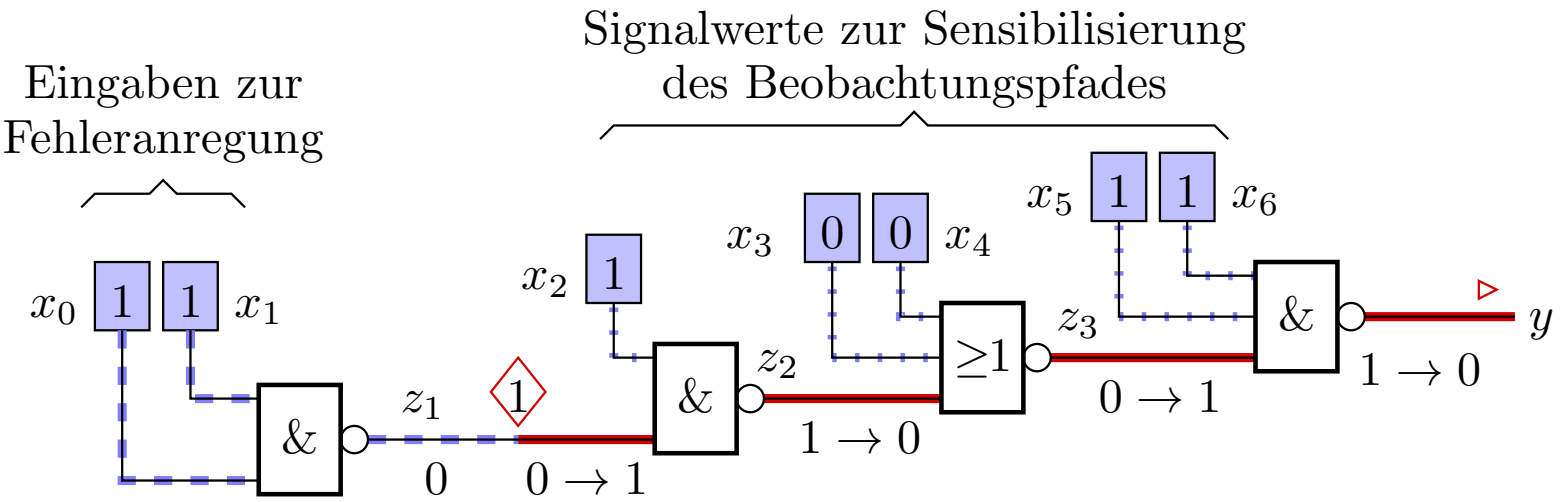


1. Die Funktion hängt nicht von x_3 ab und ist:

$$y = x_1 \wedge x_2$$

2. An dem verbleibenden AND-Gatter sind sa0(x_i) identisch mit sa0(y) nachweisbar und der Nachweis von sa1(x_1) und sa1(x_2) impliziert den von sa1(y).

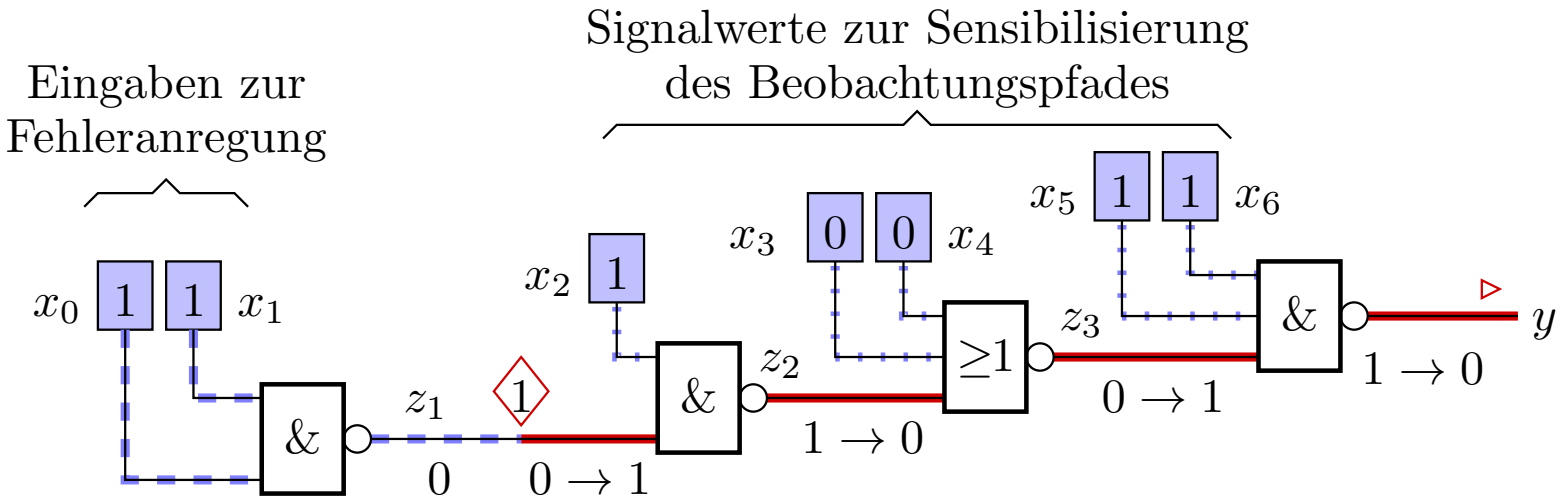
Testsuche und Nachweiswahrscheinlichkeit



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

Suche durch »Pfadsensibilisierung« (siehe später Foliensatz F5):

- Suche von Eingaben zur Einstellung »0« am Fehlerort und
- Sensibilisierung eine Beobachtungspfades zu einem Ausgang.



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

- Eingabemengen für den Fehlernachweis:

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

- Zufallstest (Annahme alle 128 Eingaben gleichwahrscheinlich):
  - Anregung mit $2^5 = 32$ von 128 mögl. Eingaben: $p_A = 2^{-2}$
  - beobachtbar mit $2^2 = 4$ of 128 mögl. Eingaben: $p_B = 2^{-5}$
  - nachweisbar mit einer of 128 mögl. Eingaben: $p_N = p_A \cdot p_B = 2^{-7}$

Test und Verlässlichkeit

Beispiel 6

Programmgröße 10.000 NLOC. 30 ... 100 Fehler je 1000 NLOC. Fehlerüberdeckung der Tests $FC = 70\%$. Zu erwartende Fehleranzahl nach Beseitigung aller erkennbaren Fehler:

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

Wie zuverlässig ist ein System mit 100 bis 300 Fehlern?

Es wird gezeigt, dass sich die Zuverlässigkeit R und die Sicherheit S proportional zur Testanzahl n und umgekehrt proportional zur Anzahl der nicht beseitigten Fehler verhalten:

$$R \sim \frac{n}{\#F_{\mathrm{NE}}} = \frac{n}{\#F_{\mathrm{CR}} \cdot (1 - FC)}; \qquad S \sim R$$

$\#F_{\mathrm{CR}}$	Fehleranzahl aus Entstehung und Reparatur (number of **f**aults from **c**reation and **r**epair).
$\#F_{\mathrm{NE}}$	Anzahl der nicht beseitigten Fehler (number of **f**aults **n**ot **e**liminated).
FC	Fehlerüberdeckung (**f**ault **c**overage), Anteil der nachweisbaren Fehler.

[F] – Wert in Fehlern

Fehlfunktionsrate durch Fehler

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

$$\zeta_{\sum} = \sum_{i=1}^{\#F_{\mathrm{NE}}} \zeta_i$$

ist eine Obergrenze $\zeta \leq \zeta_{\sum}$ und für $\zeta_{\sum} \ll 1$ (fast alle FF werden nur durch einen Fehler verursacht) der gesamten FF-Rate durch Fehler:

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

Zur Vereinfachung sei im weiteren unterstellt, dass die FF-Rate durch Störungen gegenüber der durch Fehler vernachlässigbar ist:

$$\zeta = \zeta_{\mathrm{F}}$$

$\#F_{\mathrm{NE}}$	Anzahl der nicht beseitigten Fehler (number of **f**aults **n**ot **e**liminated).
ζ_i	FF-Rate verursacht durch Fehler i (MF rate due to fault i).
ζ_{F}	Rate der durch Fehler verursachten FF (MF rate due to faults).

Einfache Abschätzung

Unter den Annahmen:

- Fehler haben beim Test dieselbe FF-Rate wie im späteren Betrieb,
- Beseitigung aller nachweisbaren Fehler,
- MF-Rate je nicht beseitigten Fehler max. $\frac{1}{n}$

beträgt die FF-Rate durch die nicht beseitigten Fehler zusammen max.:

$$\zeta \leq \#F_{\mathrm{NE}} \cdot \max\left(\zeta_i\right) = \frac{\#F_{\mathrm{NE}}}{n}$$

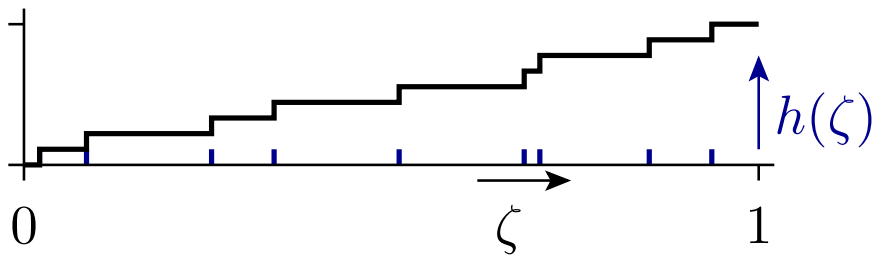
Mindestzuverlässigkeit und Sicherheit bei Tolerierung alle FF durch Störungen und robuster Reaktion auf alle erkannten FF:

$$R \geq \frac{n}{\#F_{\mathrm{NE}}}; \qquad S = \frac{R}{(1 - MC) \cdot \eta_{\mathrm{SE}}}$$

n	Anzahl der Tests (number of tests).
$\#F_{\mathrm{NE}}$	Anzahl der nicht beseitigten Fehler (number of **f**aults **n**ot **e**liminated).
η_{SE}	Anteil der sicherheitsgefährdenden FF (percentage of **s**afety **e**ndagering MF).
MC	Fehlfunktionsüberdeckung (**m**alfunction **c**overage) der Überwachung im Betrieb.

Eine genauere Abschätzung

$$F(\zeta) = \frac{\#F_i|_{\zeta_i > \zeta}}{\#F}$$



Bei Annäherung von $F(\zeta)$ durch eine stetige Verteilungsfunktion:

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

Gesamt-FF-Rate (durch Fehler) für eine bekannte Dichte der FF-Rate:

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

$F(\ldots)$	Verteilungsfunktion (distribution function).
$h(\ldots)$	Dichtefunktion (density function).
$\#F_{\mathrm{NE}}$	Anzahl der nicht beseitigten Fehler (number of **f**aults **n**ot **e**liminated).

Typische Fehlerüberdeckung von Zufallstests

Bei einem Zufallstest erfordert eine Verringerung von $1 - FC\,(n)$ um eine Dekade einen mehr als eine bis mehrere Dekaden größere Anzahl von Tests. Das ist die Eigenschaft einer Potenzfunktion:

$$1 - FC\,(n) \approx \left(\frac{n}{n_{\mathrm{Ref}}}\right)^{-k} \quad \text{mit } n \geq n_{\mathrm{Ref}} \text{ und } 0 < k < 1$$

k	1	0,5	0,33	0,25
$\frac{n}{n_{\mathrm{Ref}}}$ für $1 - FC\,(n) = 0{,}1$	10	100	10^3	10^4

n_{Ref}	Bezugstestlänge (**ref**erence test number) für $FC = 0$.
n	Anzahl der Tests (number of tests) incl. n_{Ref}.
k	Formfaktor (form factor) der Verteilung der FF-Rate $(1 < k < 1)$.

Verteilungfunktion der FF-Rate

Mit der Vereinfachung*, dass ein Zufallstest der Länge n alle Fehler mit einer FF-Rate $\zeta \geq \frac{1}{n}$ nachweist, beträgt die Verteilungsfunktion der FF-Rate vor Beseitigung der mit weiteren Tests erkennbaren Fehler:

$$F(\zeta) = 1 - FC\left(\zeta = \frac{1}{n}\right) = \begin{cases} \left(\frac{1}{n_{\mathrm{Ref}}} \cdot \zeta\right)^k & 0 \leq \zeta < \frac{1}{n_{\mathrm{Ref}}} \\ 1 & \zeta \geq \frac{1}{n_{\mathrm{Ref}}} \end{cases}$$

Wenn alle mit einer Testanzahl $n \geq n_{\mathrm{Ref}}$ erkennbaren Fehler beseitigt werden, übernimmt n die Rolle von n_{Ref} in der Gleichung zuvor:

$$F(\zeta) = \begin{cases} \left(\frac{1}{n} \cdot \zeta\right)^k & 0 \leq \zeta < \frac{1}{n} \\ 1 & \zeta \geq \frac{1}{n} \end{cases}$$

n_{Ref}	Bezugstestlänge (**ref**erence test number) für $FC = 0$.
n	Anzahl der Tests (number of tests) incl. n_{Ref}.
k	Formfaktor (form factor) der Verteilung der FF-Rate $(1 < k < 1)$.
$F(\ldots)$	Verteilungsfunktion (distribution function).

* In Wirklichkeit ist n nicht die exakte, sondern die mittlere Nachweislänge für Fehler mit FF-Rate $\frac{1}{n}$.

Dichte der FF-Rate, mittlere FF-Rate

Verteilungsfunktion, wenn alle Fehler mit $\zeta \geq 1/n$ beseitigt sind

$$F(\zeta) = \begin{cases} (n \cdot \zeta)^k & 0 \leq \zeta < \frac{1}{n} \\ 1 & \zeta \geq \frac{1}{n} \end{cases}$$

Dichte der FF-Rate und mittlere FF-Rate pro nicht beseitigter Fehler:

$$h(\zeta) = \frac{\mathrm{d}F(\zeta)}{\mathrm{d}\zeta} = \begin{cases} k \cdot n^k \cdot \zeta^{k-1} & 0 \leq \zeta < \frac{1}{n} \\ 0 & \text{sonst} \end{cases}$$

$$\bar{\zeta} = \int_0^{\frac{1}{n}} \zeta \cdot h(\zeta) \cdot \mathrm{d}\zeta = \frac{k}{(k+1) \cdot n}$$

k	Formfaktor (form factor) der Verteilung der FF-Rate $(1 < k < 1)$.
$h(\ldots)$	Dichtefunktion (density function).
ζ	Fehlfunktionsrate (malfunction rate) im Einsatz.
$\bar{\zeta}$	mittlere FF-Rate je Fehler (mean malfunction rate per fault).
n	Testanzahl für die erkennbare Fehler beseitigt sind (number of tests for which detectabe faults are **e**liminated).

Zu erwartende Fehleranzahl und FF-Rate

Für die geschätzte Zunahme der Fehlerüberdeckung

$$FC\,(n) \approx 1 - \left(\frac{n}{n_{\mathrm{Ref}}}\right)^{-k} \quad \text{mit } n \geq n_{\mathrm{Ref}} \text{ und } 0 < k < 1$$

beträgt die zu erwartende Anzahl der nicht beseitigten Fehler

$$\mathbb{E}\left[\#F_{\mathrm{NE}}\,(n)\right] = \mathbb{E}\left[\#F_{\mathrm{CR}}\right] \cdot \left(\frac{n}{n_{\mathrm{Ref}}}\right)^{-k}$$

und die FF-Rate durch diese:

$$\zeta = \mathbb{E}\left[\#F_{\mathrm{NE}}\,(n)\right] \cdot \bar{\zeta} = \mathbb{E}\left[\#F_{\mathrm{NE}}\right] \cdot \frac{k}{(k+1)\cdot n} = \mathbb{E}\left[\#F_{\mathrm{CR}}\right] \cdot \frac{k \cdot n_{\mathrm{Ref}}^{k}}{(k+1)\cdot n^{k+1}}$$

n_{Ref}	Bezugstestlänge (**ref**erence test number) für $FC = 0$.
n	Anzahl der Tests (number of tests) incl. n_{Ref}.
k	Formfaktor (form factor) der Verteilung der FF-Rate $(1 < k < 1)$.
$\#F_{\mathrm{CR}}$	Fehleranzahl aus Entstehung und Reparatur (number of **f**aults from **c**reation and **r**epair).
$\#F_{\mathrm{NE}}$	Anzahl der nicht beseitigten Fehler (number of **f**aults **n**ot **e**liminated).
$\mathbb{E}\left[\ldots\right]$	Erwartungswert (expected value).

Elimination probabilities for real tests

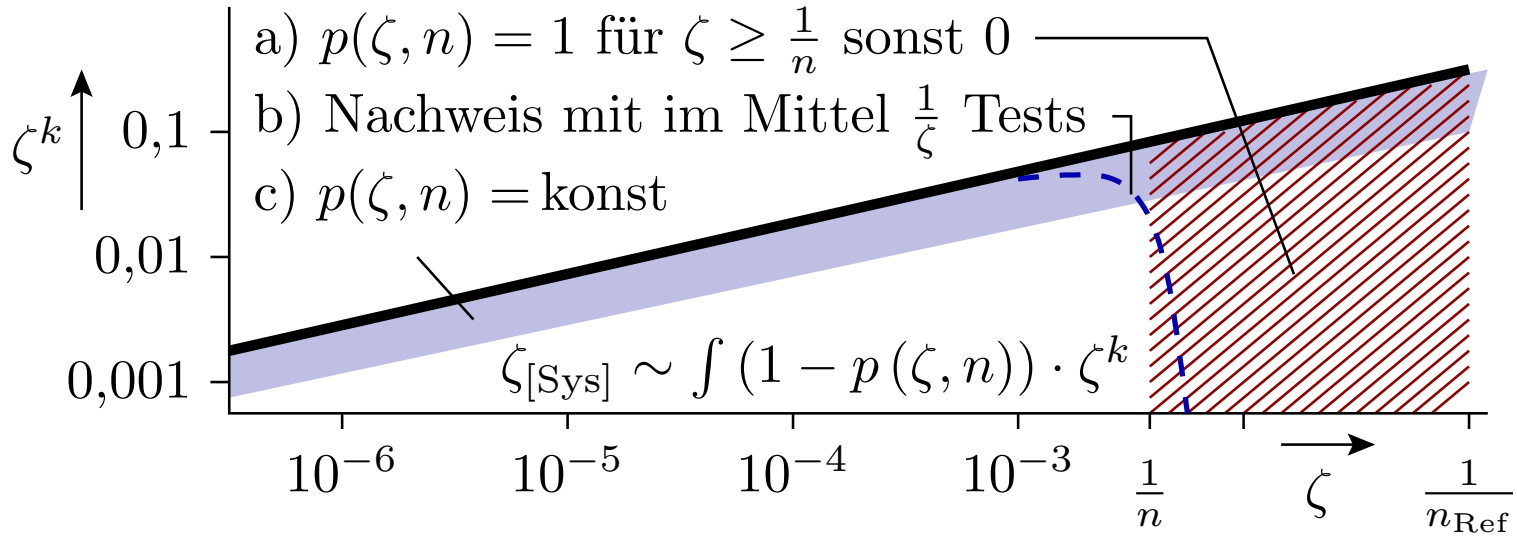
Statt Beseitigung aller Fehler mit $\zeta \geq \frac{1}{n}$ Annahme einer von ζ und n abhängigen Beseitigungswahrscheinlichkeit $p\left(\zeta, n\right)$:

$$\zeta_{[\mathrm{Sys}]} = \mathbb{E}\left[\#F_{\mathrm{CR}}\right] \cdot \int_0^{n_{\mathrm{Ref}}} \left(1 - p\left(\zeta, n\right)\right) \cdot \zeta \cdot h\left(\zeta, n_{\mathrm{Ref}}\right) \cdot \mathrm{d}\zeta$$

Für $h\left(\zeta, n_{\mathrm{Ref}}\right) = k \cdot n_{\mathrm{Ref}}^{k} \cdot \zeta^{k-1}$ ist die die FF-Rate proportional zu:

$$\zeta_{[\mathrm{Sys}]} \sim \int_0^{n_{\mathrm{Ref}}} \left(1 - p\left(\zeta, n\right)\right) \cdot \zeta^{k} \cdot \mathrm{d}\zeta$$

n_{Ref}	Bezugstestlänge (**ref**erence test number) für $FC = 0$.
$p\left(\zeta, n\right)$	Fehlererkennungswahrscheinlichkeit (**p**robability of **f**ault **e**limination) in Abhängigkeit von der FF-Rate ζ des Fehlers und der Testanzahl n.
$\#F_{\mathrm{CR}}$	Fehleranzahl aus Entstehung und Reparatur (number of **f**aults from **c**reation and **r**epair).
$\zeta_{[\mathrm{Sys}]}$	Fehlfunktionsrate (malfunction rate) durch alle nicht beseitigten Fehler.
ζ	FF-Rate pro Fehler (MF rate per fault), Integrationsvariable.
k	Formfaktor (form factor) der Verteilung der FF-Rate $(1 < k < 1)$.
$\#F_{\mathrm{SF}}$	Fehleranzahl nach Beseitigung der von statischen und fehlerorientiert gesuchten Tests gefundenen Fehler.



a) Nachweis genau mit $n = \frac{1}{\zeta}$ Tests (unsere Vereinfachung):

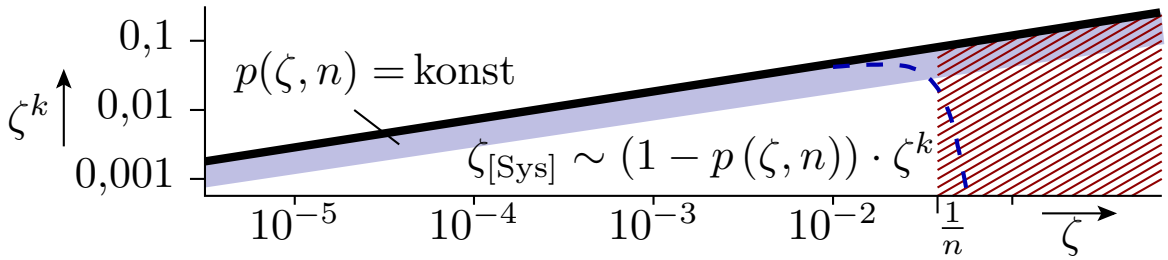
$$\zeta = \mathbb{E}\left[\#F_{\mathrm{NE}}\right] \cdot \frac{k}{(k+1) \cdot n}$$

b) Nachweis mit im Mittel $\frac{1}{\zeta}$ Tests (siehe F4.5 *Test & Zuverlässigkeit*):

$$\zeta = \mathbb{E}\left[\#F_{\mathrm{NE}}\right] \cdot \frac{k}{\cancel{(k+1)} \cdot n}$$

Wir verwenden im weiteren die Formel ohne den Term $(k+1)$, auch wenn die Herleitung erst später folgt.

Statische und fehlerorientiert gesuchte Tests



- Statische Tests suchen Fehler, ohne Funktionen auszuprobieren. $p(\zeta, n)$ unabhängig von der FF-Rate im Einsatz.
- Der Erfolg der fehlerorientierten Testsuche sei auch annähernd unabhängig von fehlerverursachten MF-Rate im Einsatz.
- Gefundene Test sind für andere Fehler Zufallstests.

$$\mathbb{E}\left[\#F_{\text{NE}}(n)\right] = \mathbb{E}\left[\#F_{\text{CR}}\right] \cdot (1 - FC_{\text{SF}}) \left(\frac{n}{n_{\text{AF}}}\right)^{-k}$$

$\#F_{\text{NE}}$	Anzahl der nicht beseitigten Fehler (number of **f**aults **n**ot **e**liminated).
$\#F_{\text{CR}}$	Fehleranzahl aus Entstehung und Reparatur (number of **f**aults from **c**reation and **r**epair).
FC_{SF}	Fehlerüberdeckung (fault coverage) der **s**tatischen und **f**ehlerorientiert gesuchten Tests.
n_{AF}	Anzahl der Tests, die **a**ngenommen **F**ehler gesucht wurden.
n	Testanzahl, für die alle erkannten Fehler beseitigt sind, incl. n_{AF}.

Zusammenfassung bis hierher

$$\mathbb{E}\left[\#F_{\mathrm{NE}}\left(n_{\mathrm{AF}}\right)\right] = \mathbb{E}\left[\#F_{\mathrm{CR}}\right] \cdot \left(1 - FC_{\mathrm{SF}}\right)$$

$$\mathbb{E}\left[\#F_{\mathrm{NE}}\left(n\right)\right] = \mathbb{E}\left[\#F_{\mathrm{NE}}\left(n_0\right)\right] \cdot \left(\frac{n}{n_0}\right)^{-k} \qquad \zeta\left(n\right) = \mathbb{E}\left[\#F_{\mathrm{NE}}\left(n\right)\right] \cdot \frac{k}{n}$$

$$\zeta\left(n\right) = \zeta\left(n_0\right) \cdot \left(\frac{n}{n_0}\right)^{-(k+1)} \qquad k = \frac{\log\left(\frac{\zeta(n_0)}{\zeta(n_1)}\right)}{\log\left(\frac{n_1}{n_0}\right)} - 1$$

$$R\left(n\right) = R\left(n_0\right) \cdot \left(\frac{n}{n_0}\right)^{k+1} \qquad S = \frac{R}{\eta_{\mathrm{SE}} \cdot (1 - MC)}$$

$\#F_{\mathrm{CR}}$	Fehleranzahl aus Entstehung und Reparatur (number of **f**aults from **c**reation and **r**epair).
FC_{SF}	Fehlerüberdeckung (fault coverage) der **s**tatischen und **f**ehlerorientiert gesuchten Tests.
n_0, n_1	Testanzahl mit bekannter FF-Rate bzw. Fehleranzahl.
n_{AF}	Anzahl der Tests, die **a**ngenommen **F**ehler gesucht wurden.
k	Formfaktor (form factor) der Verteilung der FF-Rate $(1 < k < 1)$.
ζ	Fehlfunktionsrate (malfunction rate) im Einsatz.
$\#F_{\mathrm{NE}}$	Anzahl der nicht beseitigten Fehler (number of **f**aults **n**ot **e**liminated).
MC	Fehlfunktionsüberdeckung (**m**alfunction **c**overage) der Überwachung im Betrieb.
η_{SE}	Anteil der sicherheitsgefährdenden FF (percentage of **s**afety **e**ndagering MF).

Beispiel 7

1. Um welchen Faktor verringern sich FF-Rate und Fehleranzahl , wenn die Anzahl der dynamischen Tests verdreifacht wird? Formfaktoren der Verteilung der FF-Rate $k \in \{0{,}3,\ 0{,}5\}$.
2. Welcher Zuverlässigkeitsgewinn ist zu erwarten, wenn das Personal der Testabteilung verdreifacht wird?

1. Geschätzte Werte für die Reduzierung der MF-Rate und der Fehlerzahl sowie die Erhöhung der Zuverlässigkeit:

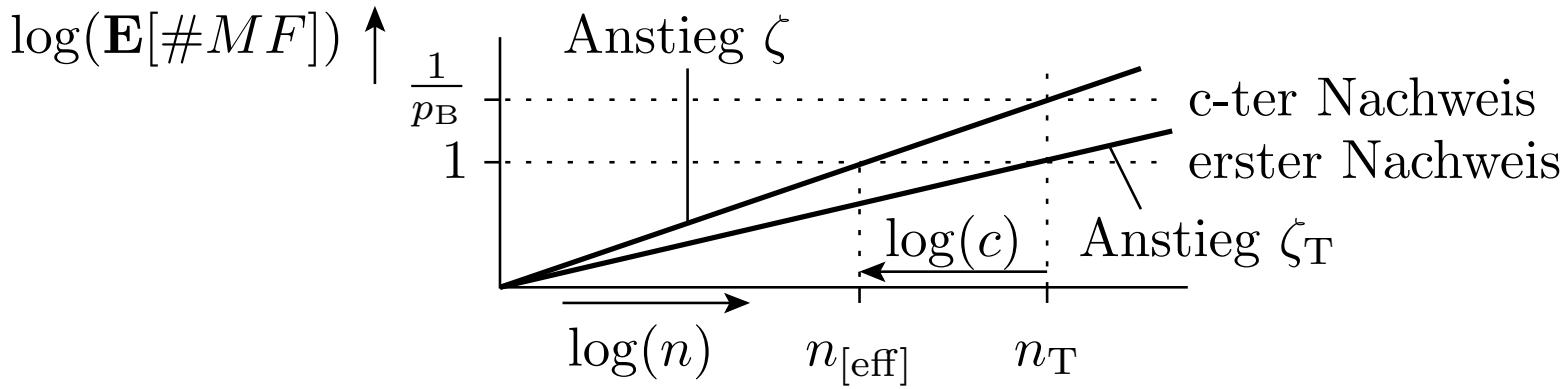
$$\frac{\zeta\left(3 \cdot n_0\right)}{\zeta\left(n_0\right)} = 3^{-(k+1)}; \quad \frac{\#F\left(3 \cdot n_0\right)}{\#F\left(n_0\right)} = 3^{-k}; \quad \frac{Z\left(3 \cdot n_0\right)}{Z\left(n_0\right)} = 3^{k+1}$$

	$\frac{\#F(3\cdot n_0)}{\#F(n_0)}$	$\frac{\zeta(3\cdot n_0)}{\zeta(n_0)}$	$\frac{Z(3\cdot n_0)}{Z(n_0)}$
$k = 0{,}3$	0,72	0,24	4,17
$k = 0{,}5$	0,56	0,19	5,19

2. Ein 3-facher Personaleinsatz für Tests und Fehlersuche erhöht die Zuverlässigkeit bei der Produktfreigabe um das 4- bis 5-fache.

Effektive Testanzahl

Effektive Testanzahl $n_{[\text{eff}]}$ ist die äquivalente Anzahl der Tests, für die alle erkennbaren Fehler beseitigt werden.



Testlängenvergrößerung*:

$$n_{[\text{eff}]} = c \cdot n_{\text{T}} \quad \text{mit } c = \frac{\zeta}{\zeta_{\text{T}}}$$

$n_{[\text{eff}]}$	effektive Testanzahl (effective number of tests) für die erkannte Fehler beseitigt werden.
$\#MF$	Anzahl der Fehlfunktionen (number of **m**al**f**unctions) während des Tests bzw. im Einsatz.
c	Testlängenvergrößerung (test number enlargement).
ζ	Fehlfunktionsrate (malfunction rate) im Einsatz.
ζ_{T}	Fehlfunktionsrate (malfunction rate) während des **T**est.

* Annahme: c wenig abhängig von der Anzahl der Tests, für die erkannten Fehler beseitigt sind.

- Wenn Fehler während des Test die halbe MF-Rate haben wie im Betrieb, dann erreicht man im Betrieb mit der halben Anzahl von Tests die gleiche Fehlerabdeckung:

$$c = \frac{\zeta}{\zeta_{\mathrm{T}}} = 2; \quad n_{[\mathrm{eff}]} = 2 \cdot n_{\mathrm{T}}$$

- Beseitigung erkannter Fehler nur mit Wahrscheinlichkeit $p_{\mathrm{Bes}} < 1$, z.B. wegen einer Fehlerkultur ohne Beseitigungserfolgkontrolle:

$$c = p_{\mathrm{FE}}; \quad n_{[\mathrm{eff}]} = p_{\mathrm{FE}} \cdot n_{\mathrm{T}}$$

- Tendentiell abweichende FF-Rate der Modellfehler ζ_{MF} von der der tatsächlichen Fehler ζ:

$$c = \frac{\zeta}{\zeta_{\mathrm{MF}}}; \quad n_{[\mathrm{eff}]} = c \cdot n_{\mathrm{T}}$$

Die Testanzahl n ist im weiteren die effektive Testanzahl.

$n_{[\mathrm{eff}]}$	effektive Testanzahl (effective number of tests) für die erkannte Fehler beseitigt werden.
$\#MF$	Anzahl der Fehlfunktionen (number of **m**al**f**unctions) während des Tests bzw. im Einsatz.
c	Testlängenvergrößerung (test number enlargement).
ζ	Fehlfunktionsrate (malfunction rate) im Einsatz.
ζ_{T}	Fehlfunktionsrate (malfunction rate) während des **T**est.
ζ_{MF}	FF-Rate (MF rate)verursacht durch **M**odell**f**ehler.

Zusammenfassung

Für den Fall, den wir im weiteren immer annehmen wollen, dass

- nach den statischen Tests und fehlerorientierten dynamischen Tests ein langer Zufallstest folgt, bei dem
- die Anzahl der nicht nachweisbaren Fehler etwa mit n^{-k} abnimmt,
- alle erkannten Fehler beseitigt werden und
- FF durch Störungen im Betrieb toleriert werden

gelten für den Beginn der Einsatzphase für die Anzahl der nicht beseitigten Fehler, die FF-Rate, die Zuverlässigkeit und die Sicherheit zu Beginn der Einsatzphase folgende Überschläge:

$$
\begin{aligned}
\mathbb{E}\left[\#F_{\mathrm{NE}}\left(n_{\mathrm{AF}}\right)\right] &= \mathbb{E}\left[\#F_{\mathrm{CR}}\right] \cdot \left(1 - FC_{\mathrm{SF}}\right) & \qquad n &= c \cdot n_{\mathrm{T}} \\
\mathbb{E}\left[\#F_{\mathrm{NE}}\left(n\right)\right] &= \mathbb{E}\left[\#F_{\mathrm{NE}}\left(n_0\right)\right] \cdot \left(\frac{n}{n_0}\right)^{-k} & \qquad \zeta\left(n\right) &= \mathbb{E}\left[\#F_{\mathrm{NE}}\left(n\right)\right] \cdot \frac{k}{n} \\
\zeta\left(n\right) &= \zeta\left(n_0\right) \cdot \left(\frac{n}{n_0}\right)^{-(k+1)} & \qquad k &= \frac{\log\left(\frac{\zeta(n_0)}{\zeta(n_1)}\right)}{\log\left(\frac{n_1}{n_0}\right)} - 1 \\
R\left(n\right) &= R\left(n_0\right) \cdot \left(\frac{n}{n_0}\right)^{k+1} & \qquad S &= \frac{R}{\eta_{\mathrm{SE}} \cdot (1 - MC)}
\end{aligned}
$$

$$\mathbb{E}\left[\#F_{\mathrm{NE}}\left(n_{\mathrm{AF}}\right)\right] = \mathbb{E}\left[\#F_{\mathrm{CR}}\right] \cdot \left(1 - FC_{\mathrm{SF}}\right) \qquad n = c \cdot n_{\mathrm{T}}$$

$$\mathbb{E}\left[\#F_{\mathrm{NE}}\left(n\right)\right] = \mathbb{E}\left[\#F_{\mathrm{NE}}\left(n_0\right)\right] \cdot \left(\frac{n}{n_0}\right)^{-k} \qquad \zeta\left(n\right) = \mathbb{E}\left[\#F_{\mathrm{NE}}\left(n\right)\right] \cdot \frac{k}{n}$$

$$\zeta\left(n\right) = \zeta\left(n_0\right) \cdot \left(\frac{n}{n_0}\right)^{-(k+1)} \qquad k = \frac{\log\left(\frac{\zeta(n_0)}{\zeta(n_1)}\right)}{\log\left(\frac{n_1}{n_0}\right)} - 1$$

$$R\left(n\right) = R\left(n_0\right) \cdot \left(\frac{n}{n_0}\right)^{k+1} \qquad S = \frac{R}{\eta_{\mathrm{SE}} \cdot (1 - MC)}$$

n	effektive Testanzahl (effective number of tests) für die erkannte Fehler beseitigt werden.
n_0, n_1	Effektive Testanzahl mit bekannter FF-Rate bzw. Fehleranzahl.
$\#MF$	Anzahl der Fehlfunktionen (number of **m**al**f**unctions) während des Tests bzw. im Einsatz.
c	Testlängenvergrößerung (test number enlargement).
k	Formfaktor (form factor) der Verteilung der FF-Rate $(1 < k < 1)$.
ζ	Fehlfunktionsrate (malfunction rate) im Einsatz.
$\#F_{\mathrm{NE}}$	Anzahl der nicht beseitigten Fehler (number of **f**aults **n**ot **e**liminated).
R	Zuverlässigkeit (**r**eliability).
S	Sicherheit (**s**afety (security)).
MC	Fehlfunktionsüberdeckung (**m**alfunction **c**overage) der Überwachung im Betrieb.
η_{SE}	Anteil der sicherheitsgefährdenden FF (percentage of **s**afety **e**ndagering MF).
$\mathbb{E}\left[\ldots\right]$	Erwartungswert (expected value).

Reifeprozesse

Das Problem immer größerer IT-Systeme

Zuverlässigkeit der Systeme im Einsatz sinkt mit der Anzahl der Fehler aus den Entstehungs- und Reparaturprozessen:

$$R \sim \frac{n^{k+1}}{\#F_{\mathrm{CR}}}$$

die mindestens proportional mit der Systemgröße zunimmt

$$\#F_{\mathrm{CR}} \approx \xi_{\mathrm{C}} \cdot N$$

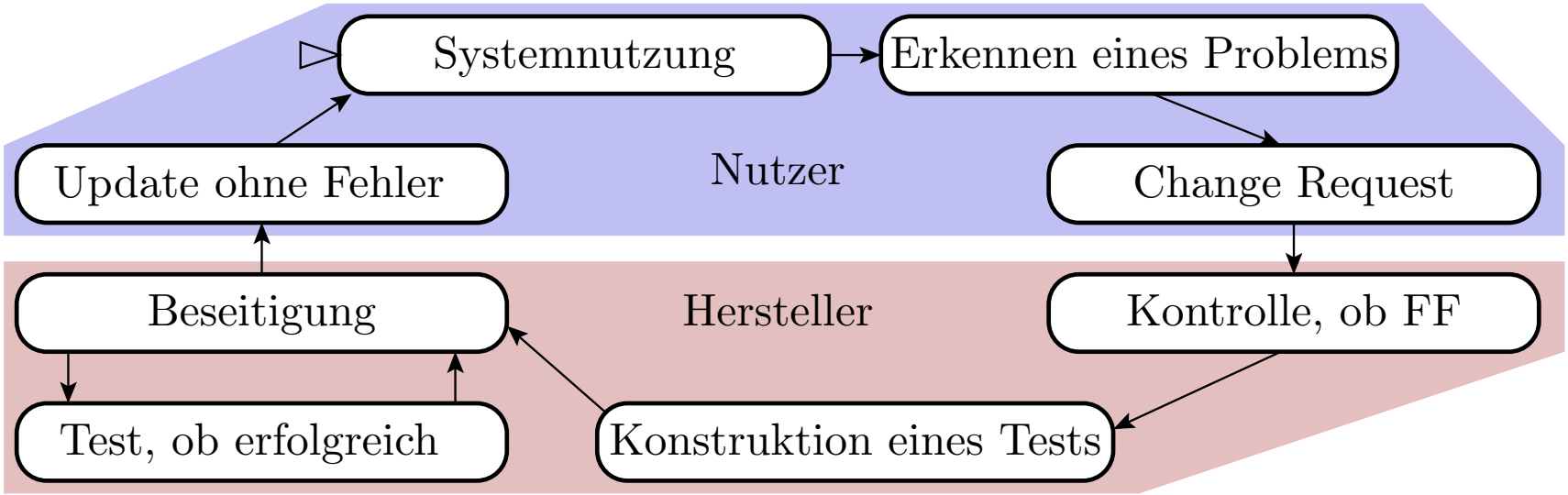
Kompensation des Zuverlässigkeitsverlust durch die wachsende Systemgröße verlangt eine Vergrößerung der Testanzahl um etwa:

$$\frac{n}{n_0} \approx \left(\frac{N}{N_0}\right)^{\frac{1}{k+1}}$$

n	effektive Testanzahl (effective number of tests) für die erkannte Fehler beseitigt werden.
$\#F_{\mathrm{CR}}$	Fehleranzahl aus Entstehung und Reparatur (number of **f**aults from **c**reation and **r**epair).
N	Systemgröße (system size) z.B. in NLOC.
$\frac{n}{n_0}$	Erhöhung der Testanzahl (test number enlagement).
$\frac{N}{N_0}$	Systemvergrößerung (system size enlagement).

Reifeprozess

Die Alternative zu immer längeren Testzeiten vor dem Einsatz ist die Installation eines Reifeprozesses mit den Nutzern als Tester.



- Erfassen der FF in der Einsatzphase.
- Sammeln der Daten, um die FF nachzustellen.
- Übermittlung an den Hersteller.
- Suche von Tests für reproduzierbaren Fehlernachweis.
- Beseitigung durch experimentelle Reparatur.
- Herausgabe und Installieren von Updates.

Effektive Testsatzlänge

1. Bei einer vermuteten Fehlfunktion stellt der Nutzer einen Änderungsanforderung (Change Request). Alternativ sendet das System einer FF-Report. FF-Reports werden in Schubladen vermuteter gleicher Ursache gesammelt.
2. Der Hersteller bevorzugt bei der Beseitigung Schubladen, die Fehler mit häufigen schwerwiegenden FF vermuten lassen.
3. Suche von Tests, die die FFs reproduzierbar anregen. Die Tests dienen zur Fehlerlokalisierung und Erfolgskontrolle.
4. Experimentelle Reperatur. Installation von Updates.

Ein Fehler wird im Mittel erst, wenn er viele FF verursacht hat, beseitigt. Die effektiven Testanzahl:

$$n_{\mathrm{M}} = p_{\mathrm{FE}} \cdot \#SR \quad \text{mit } p_{\mathrm{FE}} \ll 1$$

n_{M}	effektive Anzahl SL im Reife- (**m**atury) Prozess, für die erkannten Fehler beseitigt werden.
p_{FE}	Fehlerbeseitigungswahrscheinlichkeit (**p**robability of **f**ault **e**limination).
$\#SR$	Anzahl der Service-Leistungen (number of **s**ervice **r**esults).

Zunahme der Zuverlässigkeit und Sicherheit

Ein System im Einsatz hat viele Nutzer die über eine längere Zeit viel mehr SL nutzen, als der Hersteller in der Testphase ausprobiert kann. Unter den Annahmen:

- von Nutzungsdauern t mit einer effektiven Testanzahl des Reifeprozesses weit größer als die Testanzahl vor dem Einsatz

$$n_{\mathrm{M}} = p_{\mathrm{FE}} \cdot \#SR \gg n_{\mathrm{T}}$$

- unveränderte Nutzeranzahl und Nutzungsintensität über die Zeit und keine Änderung in der Organisation des Reifeprozesses:

$$n_{\mathrm{M}} \sim t_{\mathrm{M}}$$

Nimmt die Fehleranzahl mit Exponente $k \in (0, 1)$ mit der Reifezeit t_{M} ab. FF-Rate, Zuverlässigkeit und Sicherheit nehmen mit Exponent $k + 1$ ab bzw. zu.

n_{M}	effektive Anzahl SL im Reife- (**m**atury) Prozess, für die erkannten Fehler beseitigt werden.
n	effektive Testanzahl (effective number of tests) für die erkannte Fehler beseitigt werden.
t_{M}	Reifedauer (**m**aturing **t**ime).

$$\mathbb{E}\left[\#F_{\mathrm{NE}}\left(t_{\mathrm{M}}\right)\right] = \mathbb{E}\left[\#F_{\mathrm{NE}}\left(t_{\mathrm{M0}}\right)\right] \cdot \left(\frac{t_{\mathrm{M}}}{t_{\mathrm{M0}}}\right)^{-k}$$

$$\zeta\left(t_{\mathrm{M}}\right) = \zeta\left(t_{\mathrm{M0}}\right) \cdot \left(\frac{t_{\mathrm{M}}}{t_{\mathrm{M0}}}\right)^{-(k+1)}$$

$$R\left(t_{\mathrm{M}}\right) = R\left(t_{\mathrm{M0}}\right) \cdot \left(\frac{t_{\mathrm{M}}}{t_{\mathrm{M0}}}\right)^{k+1}$$

$$S\left(t_{\mathrm{M}}\right) = \frac{R\left(t_{\mathrm{M}}\right)}{\eta_{\mathrm{SE}} \cdot (1 - MC)}$$

t_{M}	Reifedauer (**m**aturing **t**ime).
t_{M0}	Bezugsreifedauer (**r**eference **m**aturing **t**ime), Reifedauer mit bekannter Fehleranzahl bzw. FF-Rate.
$\#F_{\mathrm{NE}}$	Anzahl der nicht beseitigten Fehler (number of **f**aults **n**ot **e**liminated).
k	Formfaktor (form factor) der Verteilung der FF-Rate $(1 < k < 1)$.
R	Zuverlässigkeit (**r**eliability).
S	Sicherheit (**s**afety (security)).
η_{SE}	Anteil der sicherheitsgefährdenden FF (percentage of **s**afety **e**ndagering MF).
MC	Fehlfunktionsüberdeckung (**m**alfunction **c**overage) der Überwachung im Betrieb.

Zuverlässigkeit gereifter Systeme

Hohe Zuverlässigkeit verlangt viele Nutzer, lange Reifezeit t_{M} und eine hohe Wahrscheinlichkeit p_{FE}, dass, wenn eine FF beobachtet wird, der verursachenden Fehler beseitigt wird.

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 ...

t_{M}	Reifedauer (**m**aturing **t**ime).
p_{FE}	Fehlerbeseitigungswahrscheinlichkeit (**p**robability of **f**ault **e**limination).

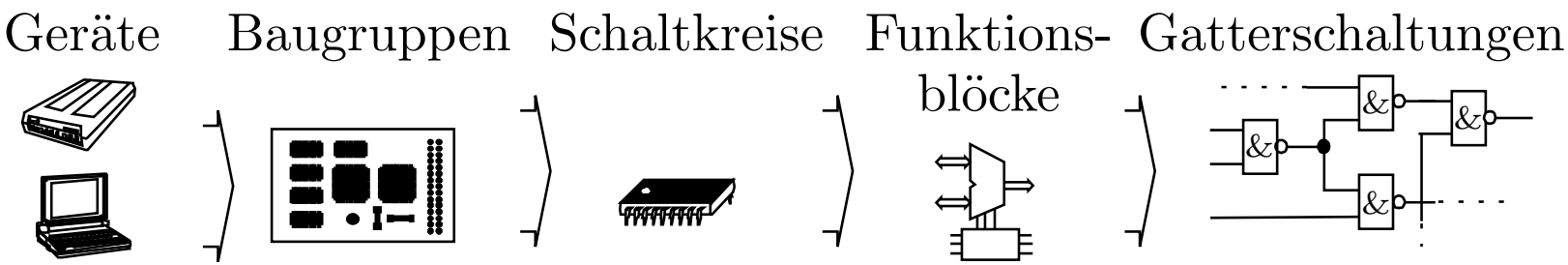
FF-Vermeidung – Lernprozesse der Nutzer

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

Wenn Wissen über Fehlerumgehungsmöglichkeiten weitergegeben wird, z.B. über Foren, FAQ-Seiten, lernt die gesamte Nutzergemeinschaft. Summierung der Nutzungsdauern t_{M} vieler Nutzer.

Modularer Test

IT-Systeme sind modular



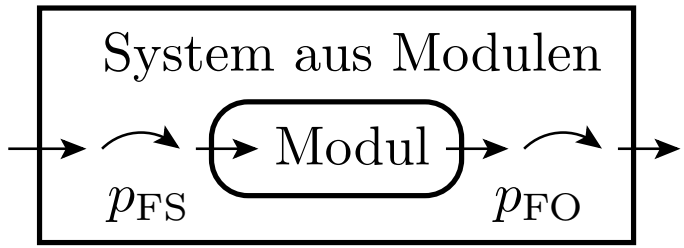
- Rechner-Systeme bestehen aus Rechnern, EA-Geräten, Druckern, Netzwerkkomponenten, ...
- Rechner und Zubehör, ...bestehen aus Hard- und Software.
- Software besteht aus Programmbausteinen, diese sind aus Anweisungen zusammengesetzt, die ihrerseits mit Maschinenbefehlen nachgebildet werden.
- Maschinenbefehle sind Service-Leistungen der Hardware. Die Hardware besteht aus Funktionsbausteinen, diese meist aus Gattern und diese wiederum aus Transistoren.
- Übergeordnete Systeme erben die Funktionen und Fehler ihrer Teilsysteme.

Modularität ist wichtig für ...

- Entwurfsprozess: Aufspaltung in Teilaufgaben, Nachnutzung von Teilentwürfen, ...
- Test: Gründlicher Test der Komponenten vor Einfügung in das übergeordnete System.
- Reparatur: Austauschbare Komponenten.
- Erhöhung der effektive Testsatzlänge für komponenteninterne Fehler.

Effektive Testanzahl, Testorganisation

FF-Rate des Systems im Betrieb für Fehler in einem Modul:



$$\zeta_{\mathrm{Sys}} = p_{\mathrm{FS}} \cdot p_{\mathrm{FO}} \cdot \zeta_{\mathrm{Mod}}$$

Erhöhung der effektiven Testanzahl:

$$n_{[\mathrm{eff}]} = c \cdot n_{\mathrm{T}} \quad \text{mit } c = \frac{\zeta_{\mathrm{Mod}}}{\zeta_{\mathrm{Sys}}} = \frac{1}{p_{\mathrm{FE}} \cdot p_{\mathrm{FO}}} \gg 1$$

In einer vernünftige Prüftechnologie werden Module vor Einbau in das übergeordnete System gründlich getestet und die Tests des übergeordneten Systems überprüft hauptsächlich die Verbindungen zwischen den Modulen.

p_{FS}	Fehleranregungswahrscheinlichkeit (**p**robability of **f**ault **s**timulation).
p_{FO}	Fehlerbeobachtbarkeitswahrscheinlichkeit (**p**robability of **f**ault **o**bservation).
ζ_{Mod}	FF-Rate des Moduls (MF rate of the **mod**ule).
$n_{[\mathrm{eff}]}$	effektive Testanzahl (effective number of tests) für die erkannte Fehler beseitigt werden.
n_{T}	tatsächliche Anzahl der Tests (actual **n**umber of **t**ests).

Fehleranteil, Ausbeute

Fehleranteil

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

Fehleranteil:

$$DL = \left.\frac{\#DP}{\#P}\right|_{\mathrm{ACR}}$$

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

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

Für zu erwartende Fehleranzahl $\mathbb{E}[\#F] \ll 1$ (fast nie mehr als ein Fehler je Produkt):

$$DL = \mathbb{E}[\#F]$$

DL	Fehleranteil (**d**efect **l**evel).
$\#P$	Anzahl der Produkte (number of **p**roducts).
$\#DP$	Anzahl der davon defekten Produkte (number of **d**efective **p**roducts thereof).
ACR	Geeignete Zählwertgrößen (**a**ppropriate **c**ounting **r**anges).
$\mathbb{E}[\ldots]$	Erwartungswert (expected value).
$\#F$	Anzahl der Fehler (number of **f**aults).

Ausbeute (Yield)

Anteil der als gut befundenen gefertigten gleichartigen Objekte:

$$Y = 1 - DL \cdot DC$$

Die Ausbeute hängt von der Defektüberdeckung DC des Tests ab, mit dem die fehlerhaften Teile aussortiert werden. Ohne Test ist $DC = 0$ und die Ausbeute $Y = 1$.

Beispiel 8

Ausbeute $Y = 95\%$, abgeschätzt mit einem Test, der $DC = 50\%$ der fehlerhaften Objekte erkennt. Fehleranteil:

$$DL = \frac{1 - Y}{DC} = 10\%$$

Y	Ausbeute (**y**ield).
DL	Fehleranteil (**d**efect **l**evel).
DC	Defectüberdeckung (**d**efect **c**overage), Anteil der fehlerhaften Produkte.

Fehleranteil nach Aussortieren

Beim Aussortieren der erkannten fehlerhaften Objekte verringern sich die Anzahl der fehlerhaften Objekte in Zähler und Nenner jeweils um die Anzahl der erkannten fehlerhaften Objekte $\#P \cdot DL_{\mathrm{M}} \cdot DC$:

$$DL_{\mathrm{SO}} = \frac{\#P \cdot DL_{\mathrm{M}} - \#P \cdot DL \cdot DC}{\#P - \#P \cdot DL_{\mathrm{M}} \cdot DC} = \frac{DL_{\mathrm{M}} \cdot (1 - DC)}{1 - DL_{\mathrm{M}} \cdot DC}$$

Der Fehleranteil vor dem Test kann auch durch die Ausbeute beschrieben werden:

$$DL_{\mathrm{M}} = \frac{1 - Y}{DC}$$

$$DL_{\mathrm{SO}} = \frac{(1 - Y) \cdot (1 - DC)}{DC \cdot Y}$$

$\#P$	Anzahl der Produkte (number of **p**roducts).
DL_{M}	Fehleranteil nach der Fertigung (**d**efect **l**evel after **m**anufacturing).
DL_{SO}	Fehleranteil nach Aussortieren (**d**efect **l**evel after **s**orting **o**ut) erkannter fehlerhafter Teile.
DC	Defectüberdeckung (**d**efect **c**overage), Anteil der fehlerhaften Produkte.
Y	Ausbeute (**y**ield).

Beispiel 9

Schaltkreisausbeute $Y = 80\%$, Fehleranteil nach Test und Aussortieren der erkannten defekten ICs $DL_{\mathrm{SO}} = 1000\,\mathrm{dpm}$. Gesucht DC.

$$DL_{\mathrm{SO}} = \frac{(1-Y)\cdot(1-DC)}{DC\cdot Y} = 1000\,\mathrm{dpm}$$

$$DC = \frac{1-Y}{DL_{\mathrm{SO}}\cdot Y + 1 - Y} - \frac{1-80\%}{10^{-3}\cdot 80\% + 1 - 80\%} = 99{,}6\%$$

Für getestete ICs findet man in der Literatur $DL_{\mathrm{SO}} = 200\,\mathrm{dpm}$ bis $1000\,\mathrm{dpm}$, für die Haftfehlerüberdeckungen der Testsätze nur $FC_{\mathrm{SA}} = 95\%$ bis 099%. Der Anteil der nicht nachweisbaren Haftfehler ist offenbar eine Zehnerpotenz größer ist als der Anteil der nicht erkennbaren defekten ICs (siehe F5.2.4 *Nachweisbeziehungen*):

- DC viel höher als die Haftfehlerüberdeckung oder
- die Angaben für DL_{SO} der getestenen ICs viel zu optimistisch?

DC	Defectüberdeckung (**d**efect **c**overage), Anteil der fehlerhaften Produkte.
DL_{SO}	Fehleranteil nach Aussortieren (**d**efect **l**evel after **s**orting **o**ut) erkannter fehlerhafter Teile.
Y	Ausbeute (**y**ield).

Systeme aus getesteten Teilsystemen

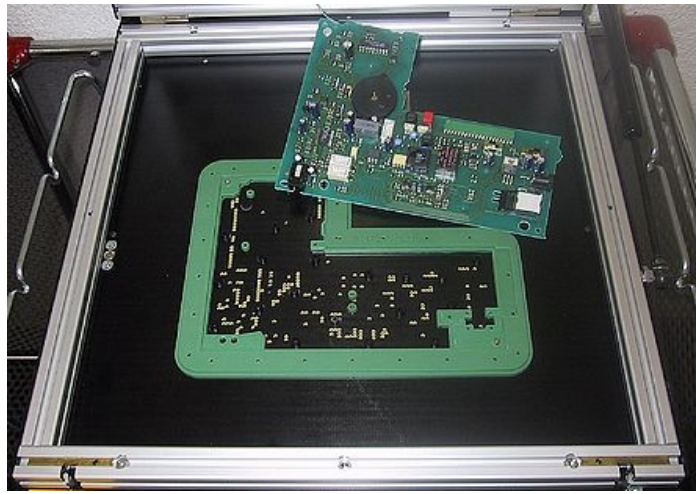
System aus getesteten als gut befundenen Bauteilen. Jedes Bauteil hat einem kleinen Fehleranteil $DL_i \ll 1$. Der Baugruppentest kontrolliert hauptsächlich auf Verbindungsfehler, aber fast nicht mehr auf Bauteilfehler. Warum?
Zu erwartende Fehleranzahl des Gesamtsystem:

$$\mathbb{E}\left[\#F_{\mathrm{Sys}}\right] = \mathbb{E}\left[\#F_{\mathrm{Con}}\right] \cdot \left(1 - FC_{\mathrm{Con}}\right) + \sum_{i=1}^{\#Prt} DL_i$$

Für $\mathbb{E}\left[\#F_{\mathrm{Sys}}\right] \ll 1$ hat das Gesamtsystem den Fehleranteil $DL_{\mathrm{Sys}} = \mathbb{E}\left[\#F_{\mathrm{Sys}}\right]$.

$\#F_{\mathrm{Sys}}$	Fehleranzahl (number of faults) Gesamtsystem.
$\mathbb{E}\left[\ldots\right]$	Erwartungswert (expected value).
$\#F_{\mathrm{Con}}$	Anzahl der Verbindungsfehler (number of **con**nection faults).
FC_{Con}	Fehlerübedeckung für Verbindungsfehler (fault coverange for **con**nection faults).
$\#Prt$	Anzahl der Bauteile (number of **p**ar**t**s).
DL_i	Fehleranteil von Bauteil i (**d**efect **l**evel of component i).

Leiterplatten



Bestückte Leiterplatten Baugruppen bestehen aus geprüften Bauteilen und werden für den Test in der Regel auf einem Nadelbett gespannt. Zielfehler: Leitungsunterbrechungen, Kurzschlüsse und Bestückungsfehler.

(Kurzschüsse und Unterbrechungen) und Bestückungsfehler praktisch $FC_{\mathrm{Con}} = 1$ und kein Nachweis für defekte Bauteile:

$$\mathbb{E}\left[\#F_{\mathrm{Sys}}\right] = \sum_{i=1}^{\#Prt} DL_i$$

Für $\mathbb{E}\left[\#F_{\mathrm{Sys}}\right] \ll 1$ hat das Gesamtsystem den Fehleranteil:

$$DL_{\mathrm{Sys}} = \mathbb{E}\left[\#F_{\mathrm{Sys}}\right]$$

$\#F_{\mathrm{Sys}}$	Fehleranzahl (number of faults) Gesamtsystem.
$\#Prt$	Anzahl der Bauteile (number of **par**ts).
DL_i	Fehleranteil von Bauteil i (**d**efect **l**evel of component i).

Beispiel Fehlerananteil einer Baugruppe

Beispiel 10

Anzahl und Fehleranteil für alle Bauteiltypen:

Typ	Anzahl	DL_i
Leiterplatte	1	20 dpm
Schaltkreise	20	200 dpm
diskrete Bauteile	35	10 dpm
Lötstellen	560	1 dpm

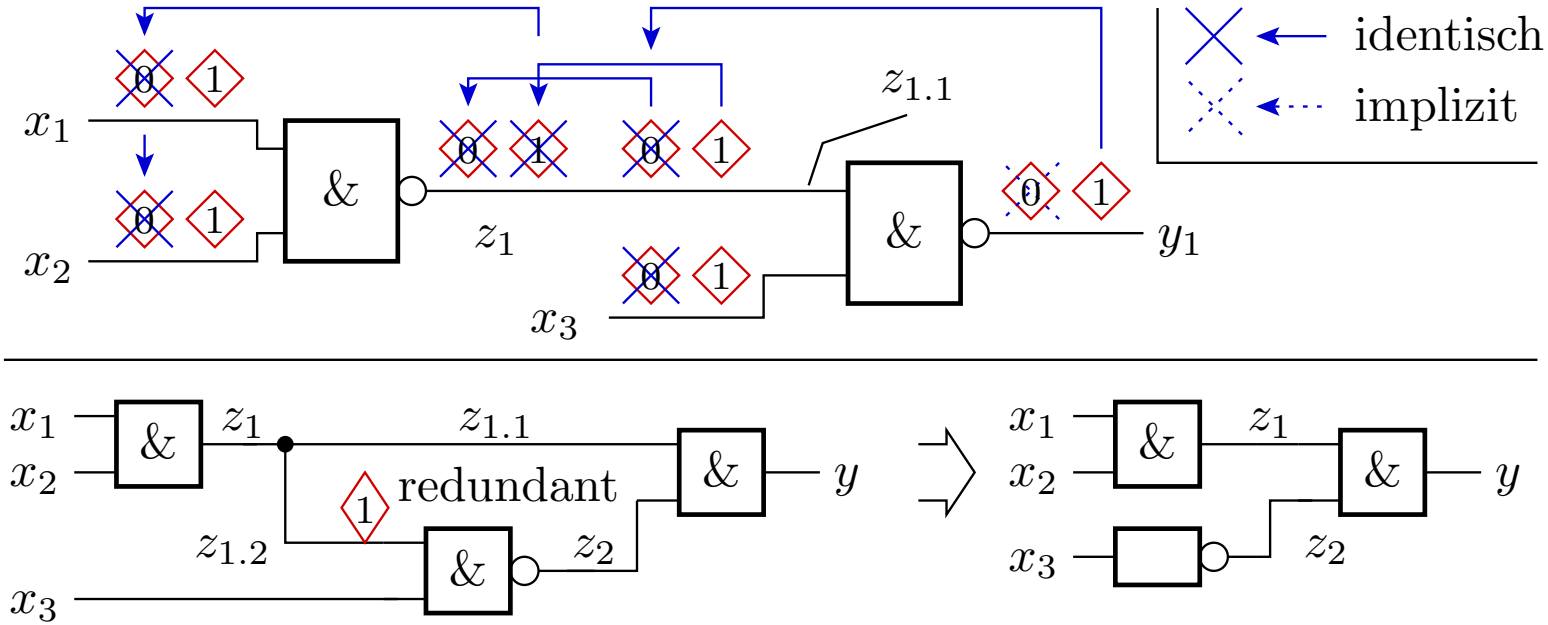
$$DL_{\mathrm{Sys}} = \mathbb{E}\left[\#F_{\mathrm{Sys}}\right] = 10\,\mathrm{dpm} + 20 \cdot 200\,\mathrm{dpm} + 35 \cdot 10\,\mathrm{dpm} + 560 \cdot 1\,\mathrm{dpm}$$
$$= 5000\,\mathrm{dpm} = 0{,}005\,\mathrm{dpu}$$

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

Rechner-Hardware kann defekte Schaltkreise enthalten, aber nur solche, die ganz selten FF verursachen.

Zusammenfassung

Abschn. 4.5: Haftfehler



- Beispiel für ein Fehlermodell zur Berechnung einer Modellfehlermenge aus einer Systembeschreibung.
- Initialfehlermenge: je Gatteranschluss sa0 und sa1.
- Zusammenfassen identisch nachweisbarer Fehler, streichen redundanter und implizit nachweisbarer Fehler.
- Die resultierende Fehlermenge dient zur Fehlersimulation und Testberechung.

Abschn. 4.6: Test und Verlässlichkeit

$$\mathbb{E}\left[\#F_{\mathrm{NE}}\left(n_{\mathrm{AF}}\right)\right] = \mathbb{E}\left[\#F_{\mathrm{CR}}\right] \cdot \left(1 - FC_{\mathrm{SF}}\right) \qquad n = c \cdot n_{\mathrm{T}}$$

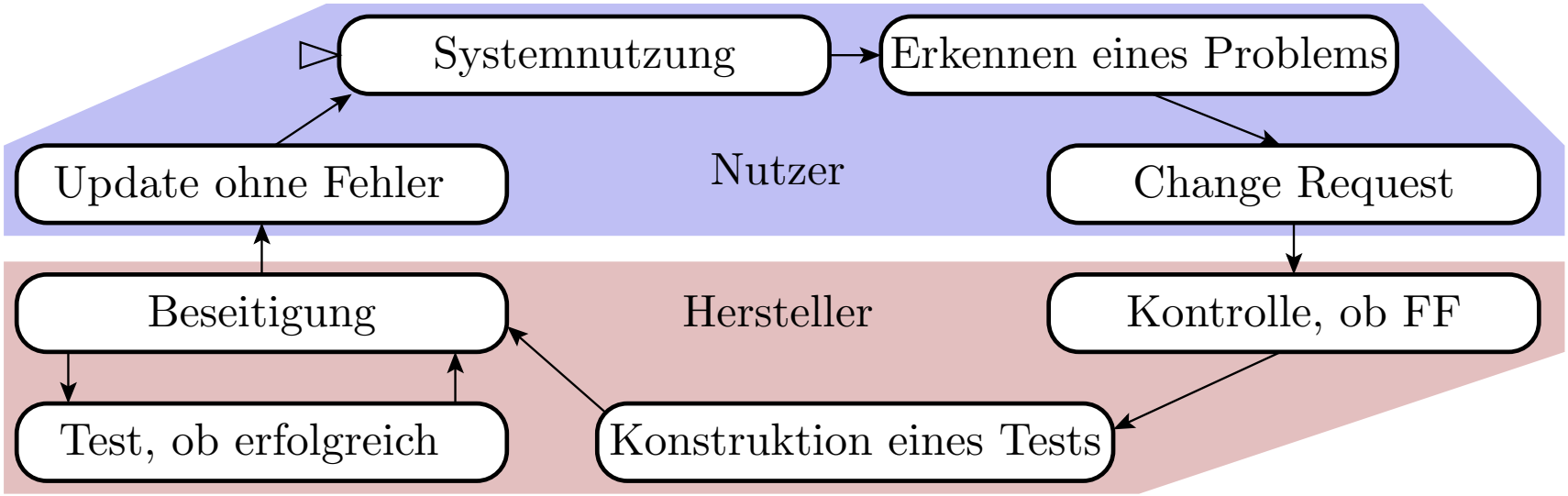
$$\mathbb{E}\left[\#F_{\mathrm{NE}}\left(n\right)\right] = \mathbb{E}\left[\#F_{\mathrm{NE}}\left(n_0\right)\right] \cdot \left(\frac{n}{n_0}\right)^{-k} \qquad \zeta\left(n\right) = \mathbb{E}\left[\#F_{\mathrm{NE}}\left(n\right)\right] \cdot \frac{k}{n}$$

$$\zeta\left(n\right) = \zeta\left(n_0\right) \cdot \left(\frac{n}{n_0}\right)^{-(k+1)} \qquad k = \frac{\log\left(\frac{\zeta(n_0)}{\zeta(n_1)}\right)}{\log\left(\frac{n_1}{n_0}\right)} - 1$$

$$R\left(n\right) = R\left(n_0\right) \cdot \left(\frac{n}{n_0}\right)^{k+1} \qquad S = \frac{R}{\eta_{\mathrm{SE}} \cdot (1 - MC)}$$

n	effektive Testanzahl (effective number of tests) für die erkannte Fehler beseitigt werden.
n_0, n_1	Effektive Testanzahl mit bekannter FF-Rate bzw. Fehleranzahl.
$\#MF$	Anzahl der Fehlfunktionen (number of **m**al**f**unctions) während des Tests bzw. im Einsatz.
k	Formfaktor (form factor) der Verteilung der FF-Rate $(1 < k < 1)$.
ζ	Fehlfunktionsrate (malfunction rate) im Einsatz.
$\#F_{\mathrm{NE}}$	Anzahl der nicht beseitigten Fehler (number of **f**aults **n**ot **e**liminated).
R	Zuverlässigkeit (**r**eliability).
S	Sicherheit (**s**afety (security)).
MC	Fehlfunktionsüberdeckung (**m**alfunction **c**overage) der Überwachung im Betrieb.
η_{SE}	Anteil der sicherheitsgefährdenden FF (percentage of **s**afety **e**ndagering MF).

Zusammenfassung Abschn. 4.7: Reifeprozesse



Fortsetzung der Fehlerbeseitigung im Einsatz mit Nutzern als Tester.

$$\mathbb{E}\left[\#F_{\mathrm{NE}}\left(t_{\mathrm{M}}\right)\right] = \mathbb{E}\left[\#F_{\mathrm{NE}}\left(t_{\mathrm{M0}}\right)\right] \cdot \left(\frac{t_{\mathrm{M}}}{t_{\mathrm{M0}}}\right)^{-k}$$

$$R\left(t_{\mathrm{M}}\right) = R\left(t_{\mathrm{M0}}\right) \cdot \left(\frac{t_{\mathrm{M}}}{t_{\mathrm{M0}}}\right)^{k+1}$$

Technik zur Erzielung sehr hoher Zuverlässigkeiten und Sicherheiten.

Abschn. 4.7: Modularer Test
Abschn. 4.8: Fehleranteil und Ausbeute

- Modularer Test erhöht u.a. die effektive Testsatzlänge für modulinterne Fehler. Daraus folgt die Regel, große Systeme aus gründlich getesteten Modulen zusammenzusetzen.
- Bei nicht reparierbaren Systemen und tauschbaren Komponenten interessiert statt der zu erwartenden Fehleranzahl, der Fehleanteil bzw. die Ausbeute, d.h. der Anteil der fehlerhaften bzw. fehlerfreien Teile.
- Der Fehleranteil von Baugruppen ist etwa die Summe der Fehleranteile aller Bauteile.

Vorlesung 4: Geplante Themen

Fehlervermeidung	Fehlerbeseitigung	FF-Behandlung
Beseitigung von Fehler-entstehungsursachen	Test und Beseitigung erkannter Fehler	Überwachung, robuste R. Fehlertoleranz Störungen

5.1 Fehlerentstehung
Modellierung von Entstehungsprozessen als Service-Leister und Fehler als dessen FF.

5.2 Determinismus und Zufall:
Fehlervermeidung ist ein Reifeprozess für ein Entstehungsprozess. Insbesondere manuellen Arbeitsschritten fehlt jedoch der Determinismus und damit die ja/nein-Kontrolle für den Fehlerbeseitigungserfolg. Wie reifen nicht deterministische Entstehungsprozesse?

5.3 Projekte, Vorgehensmodelle
Reifeprozess benötigen eine große Wiederholanzahl gleicher Abläufe, um aus erkannten Fehlern lernen zu können. Projekte sind einmalige Entstehungsabläufe. Vorgehensmodelle vereinheitlichen das Vorgehen, um dennoch aus Fehlern lernen zu können.

5.5 Qualität und Kreativität

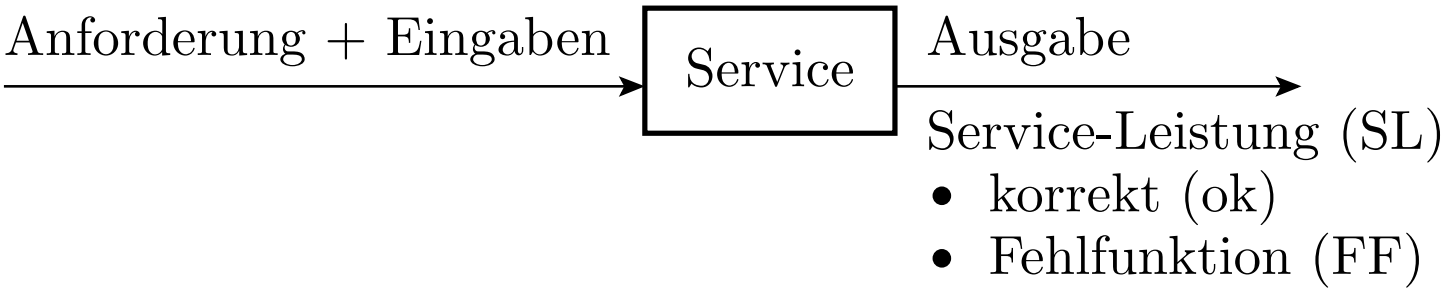
Vorgehensmodelle findet man überall dort, wo ein beständiges Lernen aus Fehlern angestrebt wird, also auch in Verwaltungen, Schulen, ...

Vereinheitlichtes Vorgehen schränkt die Kreativität ein. Der Entwurf von IT, die Wissenschaft, Ausbildung an Hochschulen, ... verlangen Kreativität. Kreativität ist in Vorgehensmodelle, so unterzubringen, dass das Lernen aus Fehlern nicht beeinträchtigt wird.

Fehlervermeidung

Fehlerentstehung

Fehler als FF des Entstehungsprozesses



Ein Entstehungsprozess ist auch ein Service
- mit Entwurfsvorgaben bzw. Material (-Eigenschaften) als Eingabe
- und Entwurfsergebnissen bzw. Produkten (oder ihre Eigenschaften als Ausgabe.

und erbt die Kenngrößen zur Beschreibung der Verlässlichkeit:
- Verfügbarkeit, FF-Rate als Fehlerentstehungsrate,
- Zuverlässigkeit, Sicherheit, ...

und die Maßnahmen zur Sicherung der Verlässlichkeit.

FF-Vermeidung ist ein Reifeprozess für ein Entstehungs-Service:
- Überwachung der entstehenden Entwürfe oder Produkte.
- Beseitigung erkannter Fehlerentstehungsursachen.

Fehlerentstehungsraten und -metriken

Ein Entstehungsprozess hat wie jeder Service eine FF-Rate, hier die Anzahl der entstehenden Fehler je SL, Produkt oder Zeitaufwand.

Für grobe Abschätzungen gibt es entstehungsprozessunabhängige Metriken für entstehende Fehler je Systemgröße oder 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, ...

$$\#F_{\mathrm{CR}} \approx \xi \cdot N$$

Beispiel 11

$\xi = 30$ Fehler / 1000 NLOC, Programm mit $N = 2000$ NLOC. Zu erwartende Anzahl der entstehenden Programmfehler: $\mathbb{E}\left[\#F_{\mathrm{CR}}\right] = 60$

$\#F_{\mathrm{CR}}$	Fehleranzahl aus Entstehung und Reparatur (number of **f**aults from **c**reation and **r**epair).
ξ	Fehlergenerierungsrate (fault creation rate).
N	Systemgröße (system size) z.B. in NLOC.

Example 12

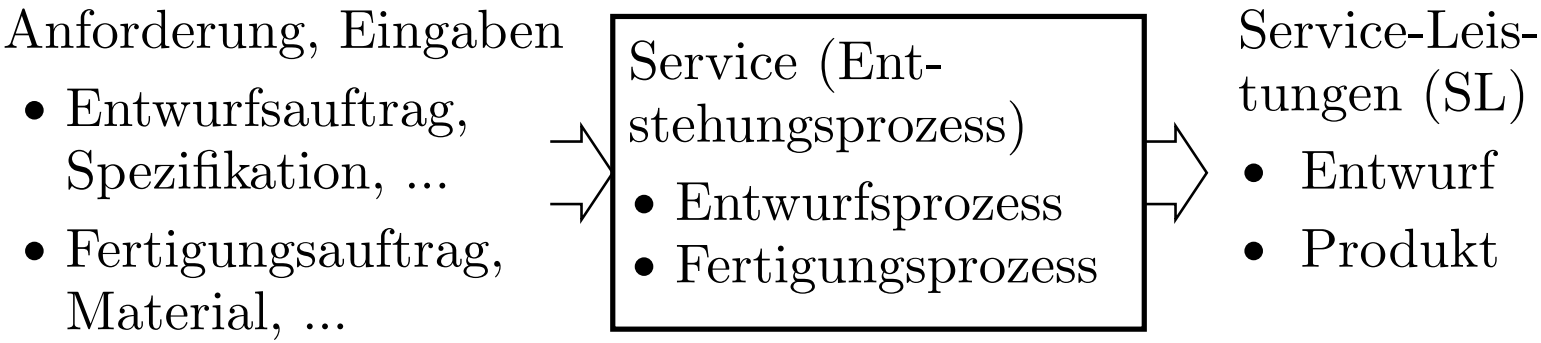
$\zeta = 1$ Fehler je 10^6 Transistoren. Schaltkreis mit $N = 10^5$ Transistoren. Zu erwartende Fehleranzahl je Schaltkreis: $\mathbb{E}\left[\#F_{\mathrm{CR}}\right] = 0{,}1$.

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.

Für unsere Modellwelt wäre es besser, die Anzahl der entstehenden Fehler aus der Art und Anzahl der Entwurfs- oder Fertigungsschritte abzuschätzen. Unüblich und deshalb gibt es in der Literatur keine anschaulichen Beispielzahlen.

Determinismus und Zufall

Fehlerentstehung

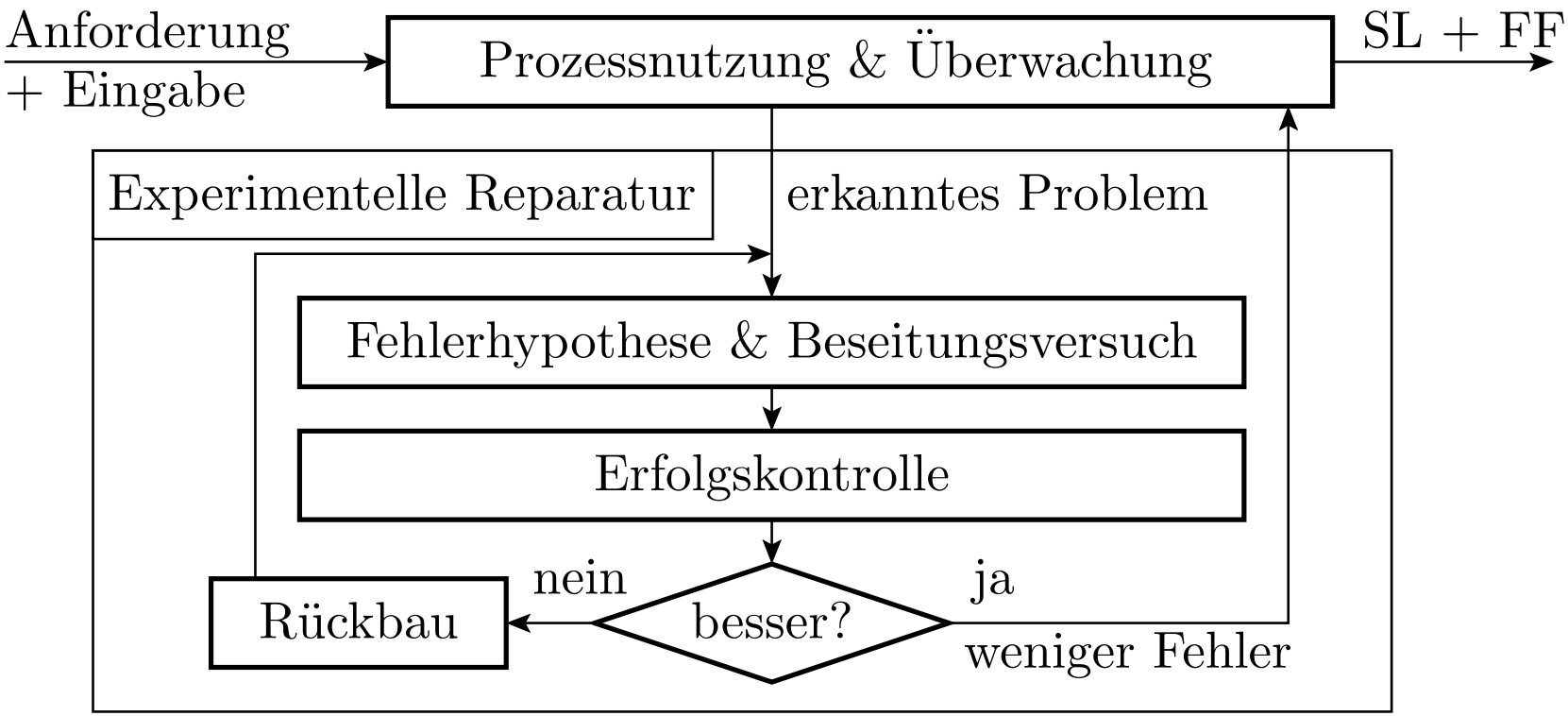


Ursachen für die Fehlerentstehung:

- Fehler: deterministische Ursache-Wirkungsbeziehung
  - beseitigbare Ursachen,
  - Erfolgskontrolle durch Testwiederholung, ...
- Störungen: zufällige Ursache-Wirkungsbeziehung
  - FF durch Wiederholung beseitigbar,
  - Erfolgskontrolle Ursachenbeseitigung 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 ein Reifeprozess für einen Entstehungsprozess mit experimenteller Reparatur zur Problembeseitigung. Iteration aus:

- Problemerkennung, Lokalisierung, Beseitigungsversuchen,
- Erfolgkontrolle durch Wiederholung der Entstehungsabläufe und
- Rückbau nach erfolglosen Beseitigungsversuchen.

Experimentelle Reparatur und Determinismus

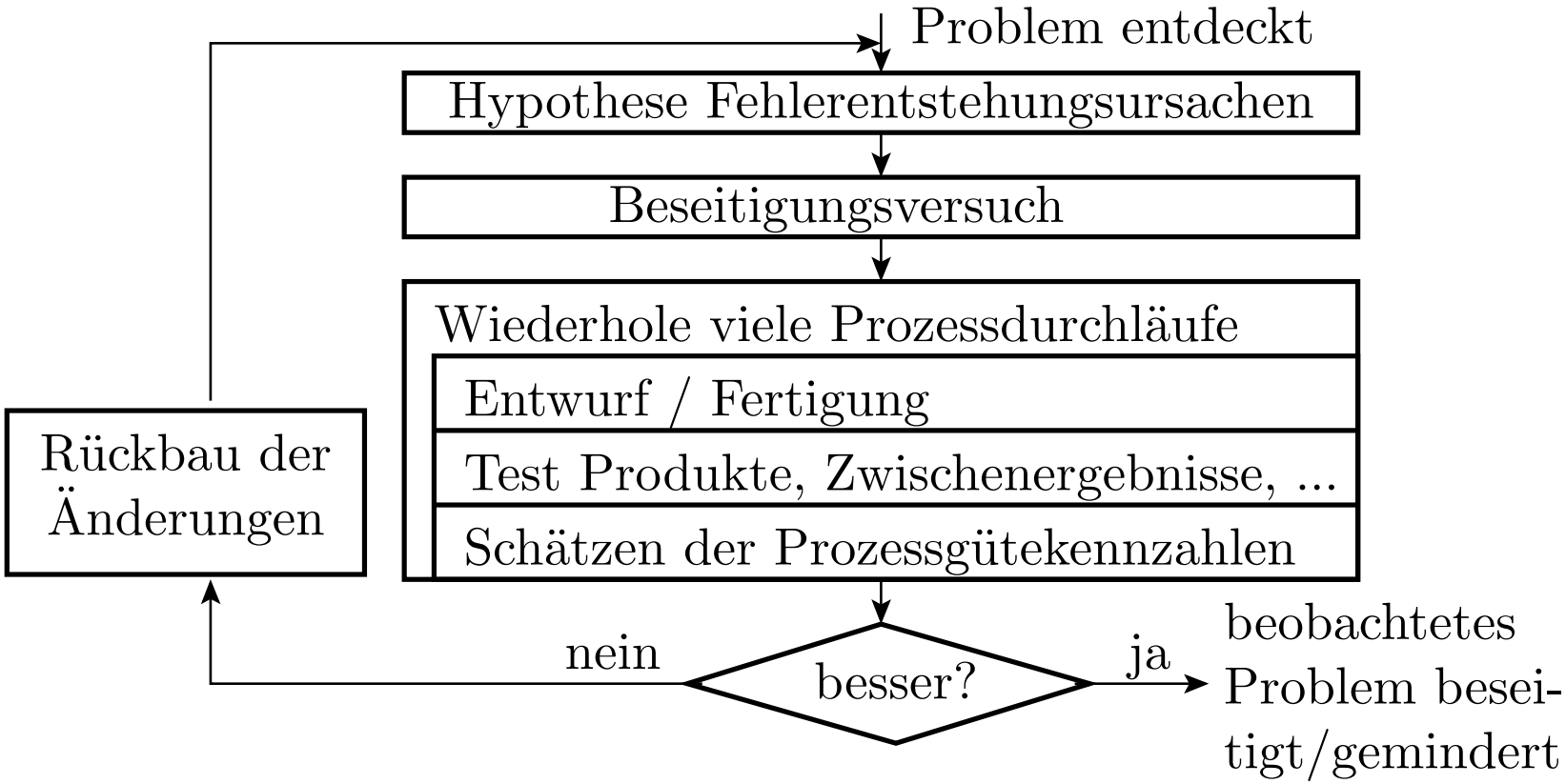
Determinismus bedeutet, dass das fehlerfreie System für denselben Entwurfs- oder Fertigungsauftrag (nach derselben Spezifikation, mit demselben Material, ...) immer dieselben Ausgaben (dasselbe Entwurfsergebnis, ein identisches Produkt, ...) liefert.

Für Fehler in deterministischen Prozessen lassen sich in der Regel Prozessabläufe 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 der erfolgreichen Problembeseitigung in der Regel

- eine statistisch signifikannte Stichprobe von Prozessdurchläufen zur Bestimmung von Prozessgütekennzahlen (typ. 1000) und
- Entscheidungen mit Irrtumswahrscheinlichkeiten, typ. wenige %).

... nicht deterministische Prozesse, Fehlerwirkungen, Störungen:



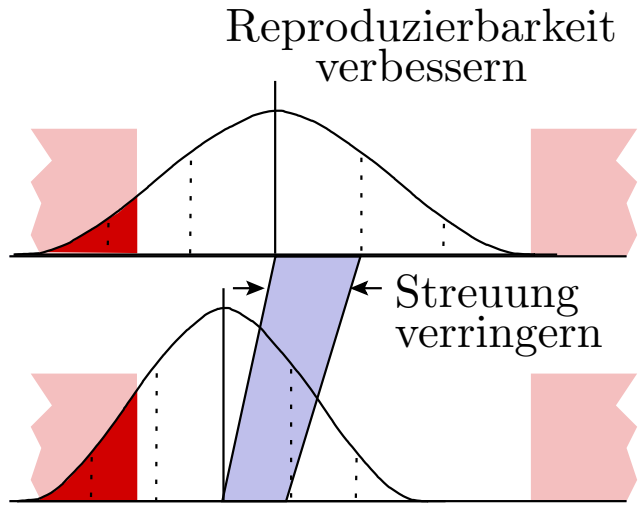
Nicht deterministische Prozesse benötigen für dieselben Absenkung der FF-Rate um Zehnerpotenzen mehr Prozessdurchläufe und haben ein deutlich höheres Risiko, dass Beseitigungsversuchen neue Fehler entstehen, die nicht durch Rückbau beseitigt werden.

Prozesszentrierung und -verbesserung

Es gibt einfach und schwer zu beseitigende Fehlerentstehungsursachen, Beispiel Prozesszentrierung / Verbesserung.

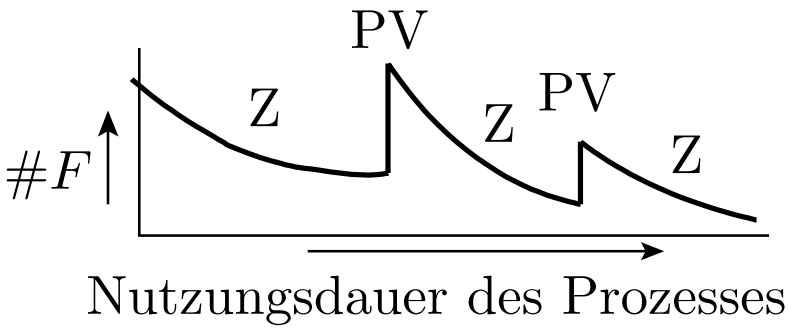
Bei der mechanischen Fertigung haben die Zielparameter, z.B. bei einer Bohrung Durchmesser und Tiefe, eine Verteilung und einen Toleranzbereich. Entstehungshäufigkeit eines Parameterfehlers ist etwa die Wahrscheinlichkeit, Parameter außerhalb Toleranzbereich:

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



Bei einer technologischen Neuerung geht die Zentrierung verloren. Sprunghafte Zunahme der Fehlerenstehungsrate.

Sägezahnverlauf Fehlerentstehungsrate



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.
- Mit der Prozesszentrierung nimmt die Fehlerentsehungsrate ab.

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 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 13

Am qualitativ hochwertigsten sind tendentiell Produkte, kurz vor technologischen Neuerungen entstehen (Maxima der Prozesszuverlässigkeit).

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[5].

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

hilft nicht nur bei der Kostensenkungung, sondern ist auch eine wesentliche Säule der Fehlervermeidung.

[5] 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 derselbe Ablauf sehr oft wiederholt wird, um möglichst viele Fehler zu erkennen und den Beseitigungserfolg zu kontrollieren.

Wie verhält es sich mit Projekten:

- Manuelle kreative Teile der Entwurfsprozesse[6] 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 Projekte von der Fehlervermeidung durch Lernen aus Fehlern aus?

[6] 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:

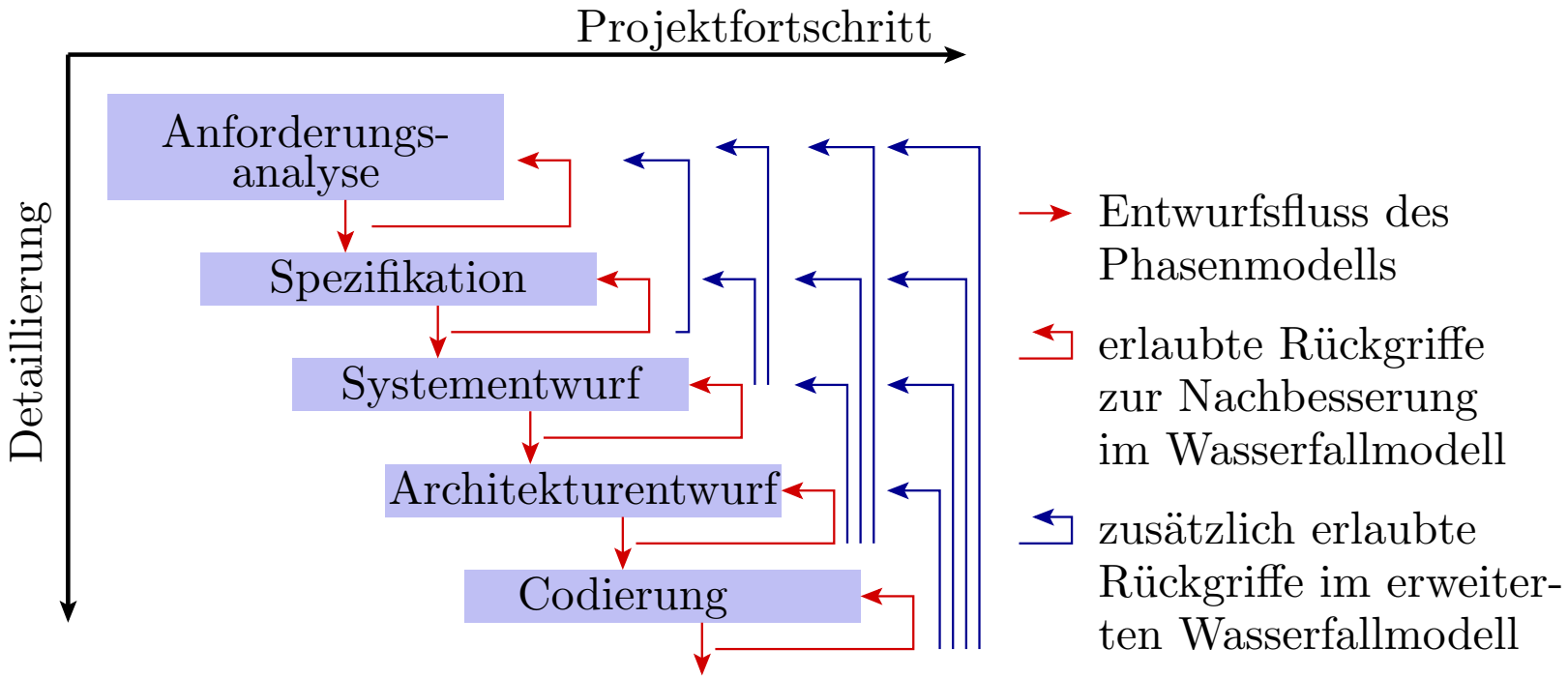
- Aufteilung in Schritte und Phasen,
- Referenzabläufe,
- Definition von Zwischen- und Endkontrollen, ...

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

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

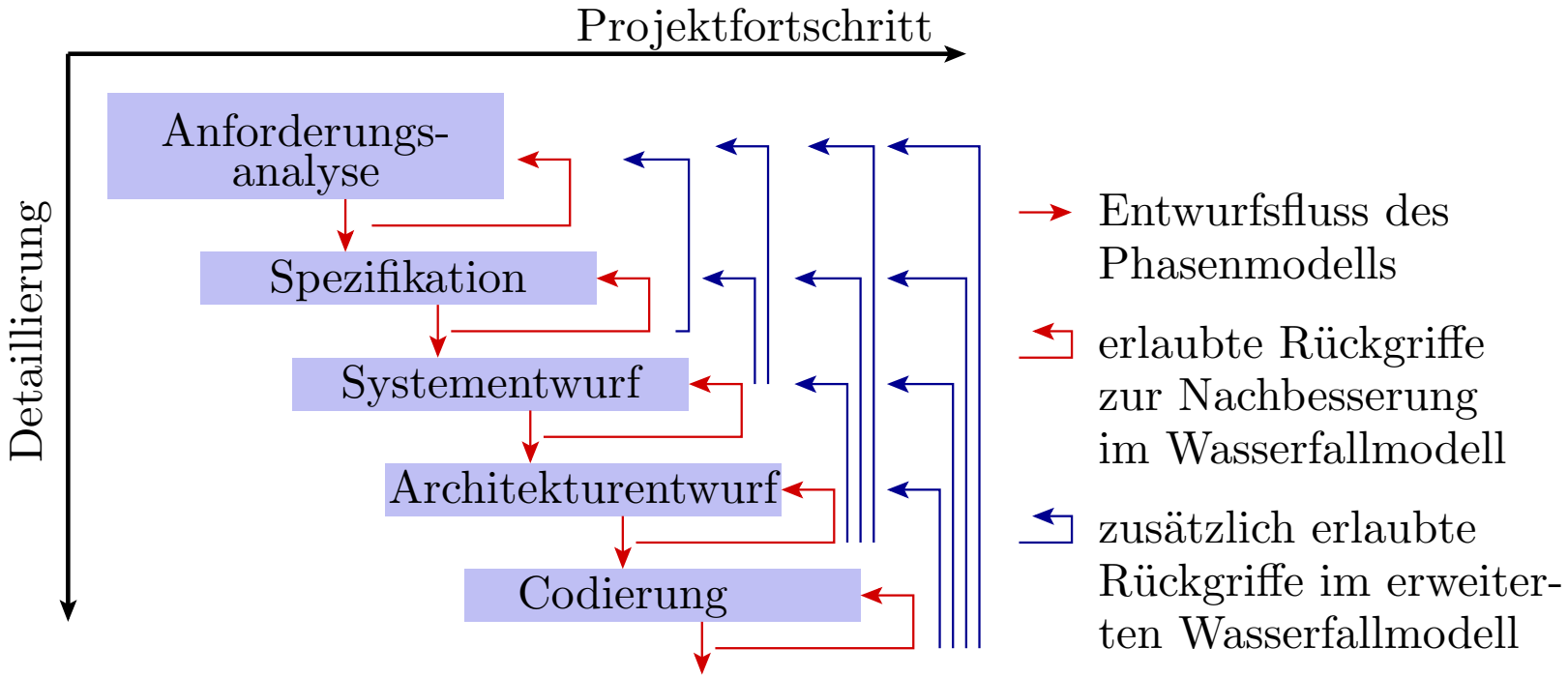
Fehlervermeidung bei Projektarbeit ist die empirische Suche nach einem guten Vorgehensmodell und seine Einhaltung.

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, ...

Rückgriffe verlängern die Anzahl der Entstehungsschritte für einen Entwurf, und darüber die Anzahl der Fehler. Ein Workaround um einen Fehler kann jedoch auch den Arbeitsaufwand erheblich erhöhen und darüber die Fehleranzahl. Schwieriger Kompromiss.

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 realen Projekten und Vorgehensmodellen vergleichbar sein:

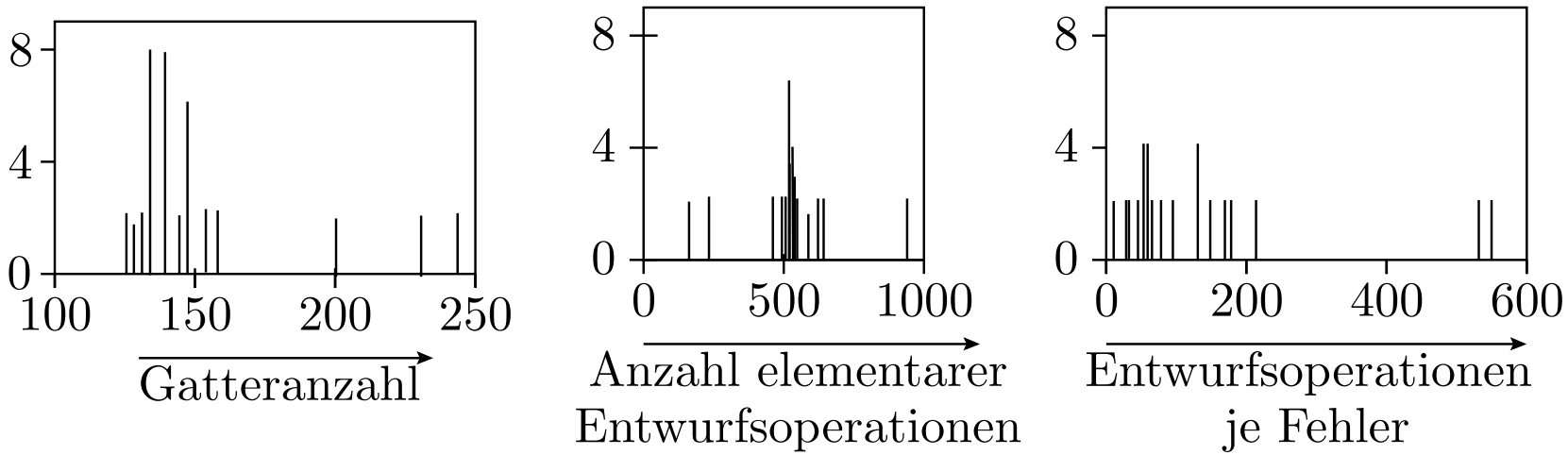
- Dauer, Kosten bezogen auf die Projektgröße
- Arbeitsschritte je entstehender Fehler, Umfrageergebnisse, ...

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

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

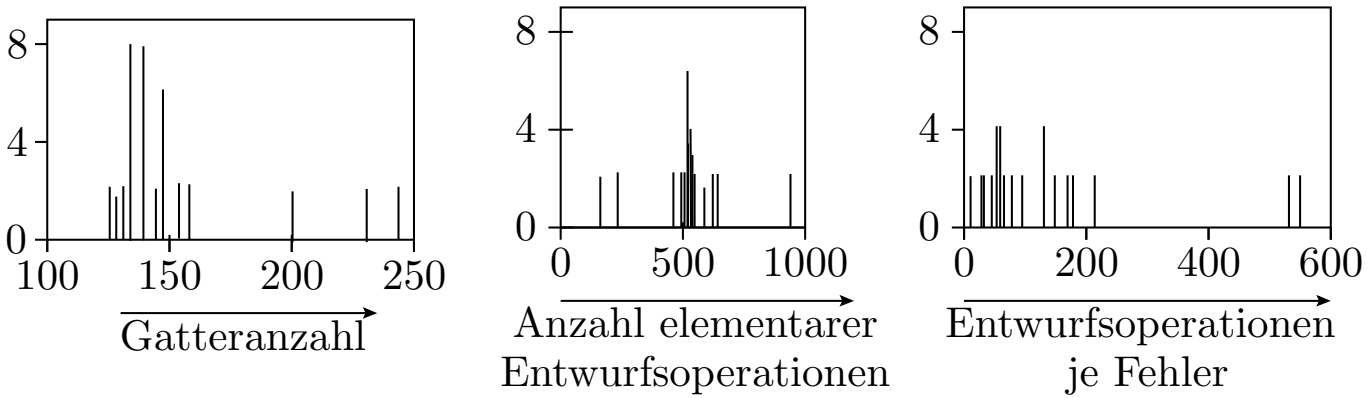
Ein Experiment [7]

Eine Gruppe von 72 Studenten sollte aus einer PLA- (**P**rogrammable **L**ogic **A**rray) Beschreibung eine Gatterschaltungen entwickeln und diese über eine GUI in ein CAD-System eingeben. 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, ...



[7] 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 bestimmt und
- Verteilung, Erwartungswert und Varianz verglichen.
- Unterschiede statistisch signifikant?

Aus den Vergleichergebnissen ließe sich schlussfolgern, ob und an welcher Hochschule Studierende für diese Aufgabe besser ausgebildet werden.

Qualität und Kreativität

Qualität und Kreativität

Qualität verlangt Fehlervermeidung. Fehlervermeidung verlangt:

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

Kreativität verlangt »Einzigartigkeit«:

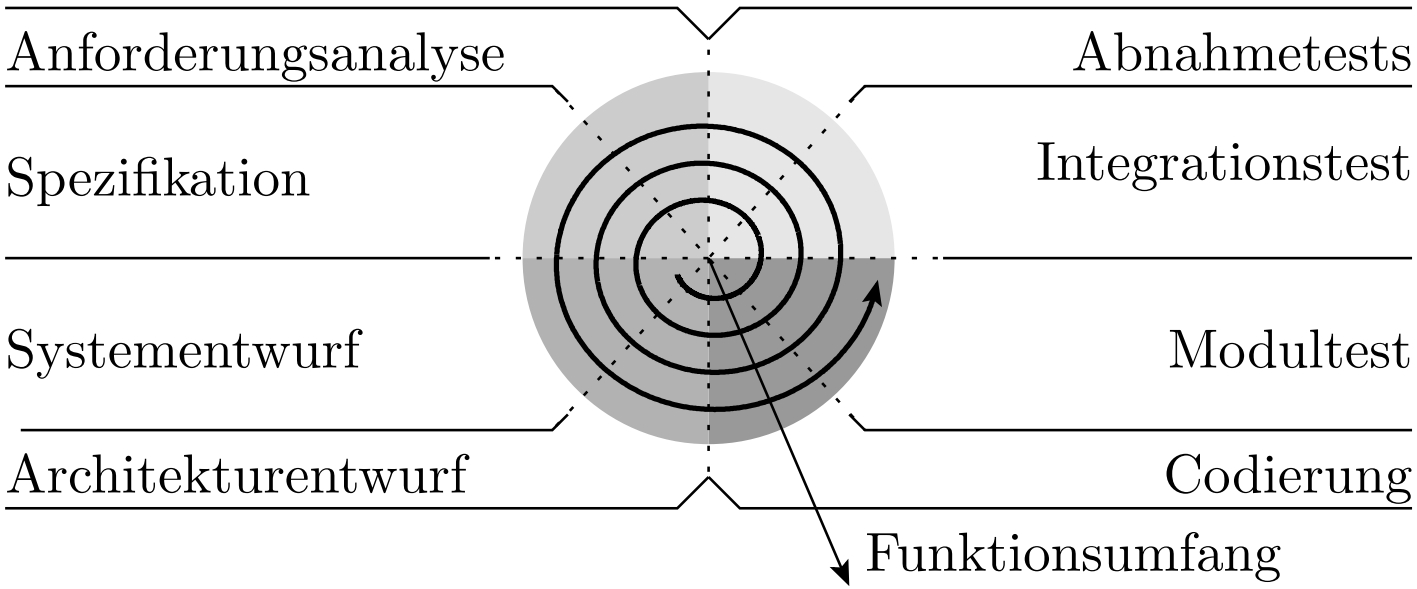
- Einbringen neuer Konzepte,
- Ausprobieren neuer Lösungswege,
- flexible Anpassung an sich ändernde Anforderungen.

Schlussfolgerung

Qualität und Kreativität haben entgegengesetzte Anforderungen an die Gestaltung von Arbeitsabläufen. IT-Entwurf verlangt Qualität und Kreativität. Wie lässt sich beides in einem Vorgehensmodell unterbringen?

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 auf mehrmalige Durchläufe 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.

Ziel:

- Minimierung der Anzahl der Entstehungsschritte und der Anzahl der entstehenden Fehler je Stufenmodelldurchlauf.
- Kreativer Freiraum in Form einer Ideensammlung für den nächsten Stufenmodelldurchlauf.

Idealerweise dürften nach jedem Stufenmodelldurchlauf an bereits implementierten Features keine Änderungen vorgenommen werden, außer Fehlerbeseitigung.

Grundidee gut, der tatsächlich erzielbare Nutzen steckt in den Umsetzungsdetails.

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.

Wie ist das an unserer Uni:

- Reift die Organisation der Lehr- und Forschungsabläufe?
- Welche Arten von Kreativität werden eingeschränkt und welche nicht? ...

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 Service-Leistungen pro FF und Zeit.

Lernphasen:

1. Wissenvermittlung: anlesen, erklärt bekommen, ...
2. Training, bis Ergebnisse vorhersagbar.
3. Professionalisierung: Prozessüberwachung; Beseitigung von Schwachstellen ud Bugs in den Abläufen.

An Universitäten:

- Phase 1: Vorlesung, Seminare, Selbststudium, ...
- Phase 2: Übung, Klausurvorbereitung[8], Praktika.
- Phase 3: Aus Zeitgründen erst in Verlässlichkeitder Berufspraxis für den eigenen eingeschränkten Tätigkeitsbereich.

[8] 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 nicht für »fehlerarme Arbeitsabläufe« trainiert.
- In industriellen Software-Projekten entstehen durch Akademiker tendenziell mehr Fehler je Aufgabengröße.
- Die Kosten für die Fehlerbeseitigung trägt der Industriepartner.
- Deshalb rechnet es sich normalerweise 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.

Fehlervermeidung eröffnet interessante Blickwinkel auf Technologien, Institutionen, Behörden, ... und deren Weiterentwicklung.

Zusammenfassung

Fehlervermeidung	Fehlerbeseitigung	FF-Behandlung
Beseitigung von Fehler-entstehungsursachen	Test und Beseitigung erkannter Fehler	Überwachung, robuste R. Fehlertoleranz Störungen

Abschn: 5.1: Fehlerentstehung

Modellierung von Entstehungsprozessen als Service-Leister und Fehler als dessen FF. Abschätzung der Anzahl der entstehenden Fehler aus Systemgröße, Anzahl der Arbeitsschritte, ... und Erfahrungswerten z.B. »Fehler je 1000 NLOC«.

Abschn. 5.2: Determinismus und Zufall

Fehlt der Determinismus erfordert die Erfolgskontrolle statistische Untersuchungen an tausenden von entstandenen Produkten. Das verlangsamt die Reifeprozesse. Aus dem üblichen Ablauf

- Prozessverbesserung alle paar Jahre und
- kontinuierliche Suche nach Möglichkeiten zur Minderung von Fehlerentstehungsursachen

folge ein sägezahnfömiger Verlauf der Fehlerentstehungsrate mit Spitzen an den Zeitpunkten der Prozessumstellungenen.

Abschn. 5.3: Projekte, Vorgehensmodelle

Reifeprozess benötigen eine große Wiederholanzahl gleicher Abläufe. Um auch bei Projekten aus erkannten Fehlern lernen zu können, erfolgt Projektarbeit nach Vorgehensmodellen. Klassiker sind die Stufenmodelle, die Entwürfe in Phasen teilen und Kontrollen und Aktivitäten beim Stufenübergang definieren. Problematisch ist die Überprüfung, ob eine Änderung einer Verbesserung bewirkt.

Abschn. 5.4: Qualität und Kreativität

Vorgehensmodelle findet man überall dort, wo ein beständiges Lernen aus Fehlern angestrebt wird, also auch in Verwaltungen, Schulen, ... In den evolutionären Vorgehensmodellen wird Kreativität so untergebracht, dass neue Ideen für die Spezifikation von Folgeprojekte gesammelt werden, die dann wieder idealweise nach einem rückgriffreien Stufenmodell ablaufen.

Fehlervermeidung eröffnet interessante Blickwinkel, wie und wohin sich Technologien, Institutionen, Behörden und z.B. auch an unserer Hochschule die Ausbildung sich weiterentwickelt.