Test und Verlässlichkeit Foliensatz 1: Gefährdungen, Gegenmaßnahmen und Kenngrößen

Prof. G. Kemnitz

May 11, 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

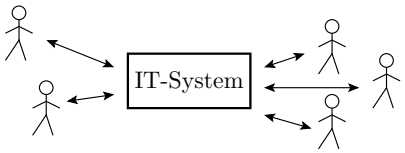
- Prüfung ab 10 Teilnehmer schriftlich.
- Erlaubte Hilfsmittel Prüfungsklausur: Eigene Ausarbeitung incl. Handouts mit eigenen Kommentaren und die eigenen Hausübungen, Taschenrechner.
- Erlaubte Hilfsmittel mündlichen Prüfung: ein A4-Blatt (einseitig) mit eigenen Ausarbeitungen.

Alle weiteren Infos siehe Web-Seite.

Contents

1 Einführung

Vertrauen und Verlässlichkeit



IT-Systeme automatisierten intellektuelle Aufgaben:

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

Einsatzvoraussetzung ist Vertrauen, dass

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

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

+

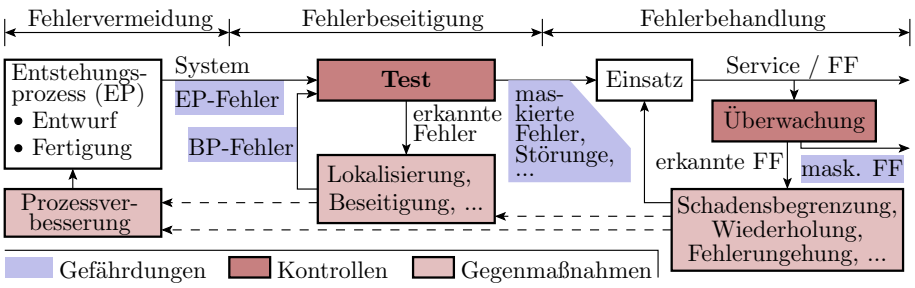
Verlässlichkeit

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

Die Lehrveranstaltung beschränkt sich auf objektive (zählbare, messbare, ...) Aspekte und folgt der Klassifikation nach Lapri[1]:

- Gefährdungen (Threats): Fehlfunktionen (FF) und deren Ursachen. Das sind Fehler, Störungen, Ausfälle. Auch die haben Ursachen, die sich untersuchen, zählen und beseitigen lassen.
- Gegenmaßnahmen zur Gefährdungsminderung (Means):
  - Überwachung, bei FF Wiederholung, Fehlerumgehung, ...
  - Test, Fehlerdiagnose und -beseitigung,
  - Fehlervermeidung durch Verbesserung der Entstehungsprozesse.
- Kenngrößen (Attributes): Zähl- und messbare Größen zur Quantifizierung und zum Vergleich der Gefährdungen und Gegenmaßnahmen.

Die 3 Ebenen zur Sicherung der Verlässlichkeit



1. Fehlervermeidung: Bei Entwurf und Fertigung entstehen Fehler. Ursachenbeseitigung ⇒ weniger entstehende Fehler.

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

2. Fehlerbeseitigung: Beseitigung erkannter Fehler. ⇒ weniger Fehler und FF im Einsatz.
3. Fehlerbehandlung: Schadensbegrenzung, aufrechterhalten der Funktion, bei unvorhergesehenen Eingaben und internen FF.
   - Robustheit: kein unkontrolliertes Verhalten bei internen FF.
   - Fehlertoleranz: automatische Korrektur / Umgehung interner FF.

Was kostet Verlässlichkeit?

Zusätzliche Entwurfs- und Betriebskosten für

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

Zusätzliche Systemfunktionen für

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

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

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

Was dient eigentlich bei einem Auto alles zur Sicherstellung der Verlässlichlichkeit (Zuverlässigkeit, Betriebssicherheit, ...)? – Viel:

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

Auch bei Autos ist Verlässichkeit das teuerste Feature.

Der Preis fehlender Verlässlichkeit

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

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

Urheber der nahen Katastrophe war ein defekter Schaltkreis.

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

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

Lernziele der Vorlesung

1. Überblick Gefährdungen, Gegenmaßnahmen, Kenngrößen
2. Einführung in die Stochastik 1: Wahrscheinlichkeit
3. Einführung in die Stochastik 2: Verteilungen
4. Überwachung
5. HW: Fehlermodellierung, Testsatzberechnung, Selbsttest
6. SW: Spezielle Aspekte, Testauswahl
7. Ausfälle, Wartung, Fehlertoleranz

Warum heißt die Vorlesung »Test und Verlässlichkeit«

- Verlässlichkeit wird auf allen drei Ebenen durch Iterationen aus Kontrolle, Problembeseitigung, Erfolgskontrolle gesichert.
- Bei einer vernünftige Fehlerkultur werden alle erkennbaren relevanten Probleme beseitigt, so dass die Verlässlichkeit von der Anzahl und schwere der verborgenen Probleme bestimmt wird.
- Die Tests sind die Filter dafür, was unerkannt bleibt und damit Hauptansatzpunkt zur Schaffung verlässlicher Systeme.

Foliensätze

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

2 Verlässlichkeit

Kenngrößen zur Beschreibung der Verlässlichkeit

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

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

Weitere Kenngrößen:

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

2.1 Service und FF

Service-Leistungen und Fehlfunktionen

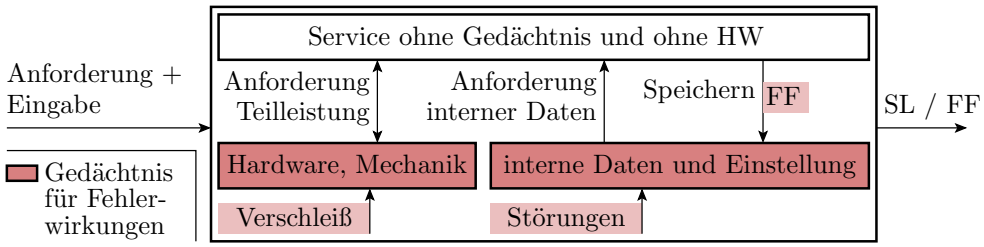
Ein IT-System, aber auch ein Entwurfs- oder Fertigungsprozess für ein IT-System ist ein Service, der im fehlerfreien Fall auf Anforderung aus Eingaben korrekte Ausgaben erzeugt. Fehler, Störungen und Ausfälle können bewirken:

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

Zählen von FF-Bursts und Ausfällen:

- Der Beginn einer FF-Burst zählt im weiteren als eine einzige FF.
- Während einer erkannten FF-Burst (erhöhte FF-Rate) gilt das System bis zur Problembehebung als nicht verfügbar[4].

Ursachen für Burst-FF und Komplettausfälle

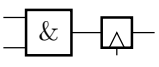
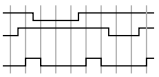


FF-Abhängigkeiten aufeinanderfolgender FF:

- Hardware und Mechanik verschleißt. In Folge entstehen während der Nutzung Fehler, die mit einer gewissen Rate FF verursachen oder die Erbringung der SL unterbinden. Problembeseitigung: Tausch / Reparatur der Mechanik oder HW.
- Software speichert in der Regel Daten und Einstellungen, die durch FF, verursacht durch Fehler, Störungen oder Ausfälle, verfälscht werden können. Problembeseitigung: Neuinitialisierung, Datenwiederherstellung von einem Backup.

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 ...

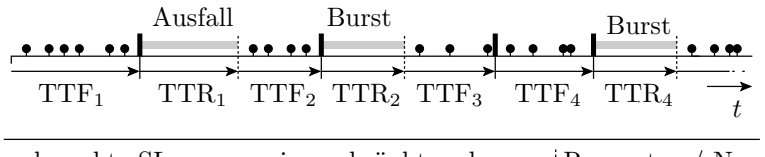
Server	E: z.B. eine Datenbankanfrage A: Ergebnisdatensatz	
Fertigungsprozess	E: Fertigungsauftrag, Material, ... A: gefertigtes Produkt	
Entwurfsprozess	E: Entwurfsauftrag A: Entwurf	

[4] Bei vernüftiger Fehlerkultur werden Systeme mit gravierenden Problemen repariert.

2.2 Verfügbarkeit

MTTF, *MTTR*, **Verfügbarkeit und** *PFD*

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



korrekte SL | Fehlfunktion | eingeschränkte oder oder keine Funktion | Reparatur / Neuinitialisierung

TTF_i Zeit bis zur nächsten FF (Time to Fail)

TTR_i Zeit bis Reparatur / Neuinitialisierung (Time to Repair)

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

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

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

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

  (#FF – Anzahl Nichtverfügbarkeitsintervalle).

- Verfügbarkeit:

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

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

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

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

Reparaturzeiten für hochverfügbare Systeme

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

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

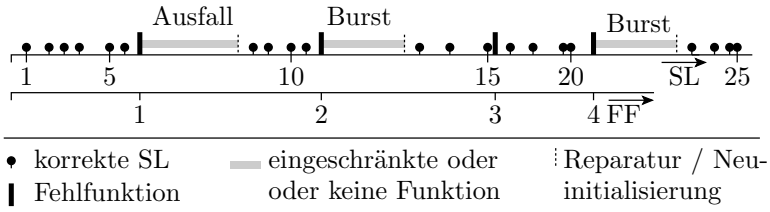
- unterbrechungsfreie Stromversorgung,
- Raid-Speicher,
- gespiegelte Server, vorbeugende Wartung, ...

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

2.3 Zuverlässigkeit

Zuverlässigkeit

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



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

($\#SL$ – Anzahl der Service-Leistungen; $\#FF$ – Anzahl der Fehlfunktionen). Service-Anforderungen und FF, während das System nicht oder nur eingeschränkt funktioniert, werden nicht mit gezählt[5].

Fehlfunktionsrate und $MTTF$

Die Fehlfunktionsrate ist der Kehrwert der Zuverlässigkeit:

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

Beschreibung als Verhältnis zwischen $MTTF$ und MTS:

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

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

($MTTF$ – **M**ean **T**ime **T**o **F**ailture, mittlere Zeit bis zur Fehlfunktionen; MTS – **M**ean **T**ime **T**o **S**ervice, mittlere Service-Dauer).

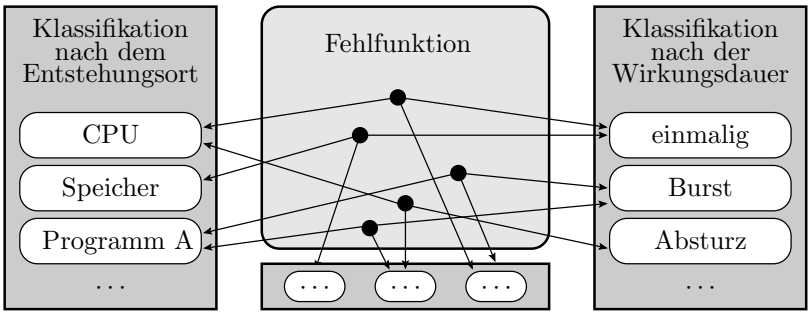
Beispiel 1. Innerhalb von 30 h Programmnutzung 3 FF, $MTS = 0{,}1\,\mathrm{h/SL}$:

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

Teilzuverlässigkeiten

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

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



[5] Annahme einer Fehlerkultur, in der bekannte Probleme vor Weiternutzung abgestellt werden.

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

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

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

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

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

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

Beispiel 2. Die Fehlfunktionen seien entweder vom Speicher, vom Prozessor, von der Software oder vom Rest verursacht. Es liegen folgende $MTTF$-Werte für Teilsysteme vor:

Teilsystem i	Speicher	Prozessor	Software	Rest
$MTTF_i$	500 h/FF	3.000 h/FF	1000 h/FF	2.000 h/FF

Mittlere Service-Dauer $MTS = 1\,\text{min/SL}$.

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

Lösung

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

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

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

2. FF-Rate und Zuverlässigkeit des Gesamtsystems:

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

2.4 Sicherheit

Schaden durch Fehlfunktionen

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

- SIL1, AK 2 & 3: Kleine Schäden an Anlagen und Eigentum.

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

Mit der Sicherheitsstufe sind u.a. Obergrenzen der $PFH = \zeta \cdot \frac{MTS}{1\,\mathrm{h}}$ **P**robability of **F**ailure per **H**our, Wahrscheinlichkeit einer FF in einer Stunde und der PFD – **P**robability of **F**ailure on **D**emand, Wahrscheinlichkeit der Nichtverfügbarkeit und Mindestanforderungen an die Fehlerbehandlung verbunden:

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

Sicherheit

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

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

Rate der gefährdenden FF als Kehrwert der Sicherheit:

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

($\#GFF$ – Anzahl der gefährdenden FF). Arten von Sicherheiten:

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

Sicherheit und Zuverlässigkeit

Die gefährdenden FF sind ein kleiner Anteil $\eta_{\mathrm{g}} \ll 1$ aller FF abschätzbar aus der Anzahl aller FF und der gefährdenden FF innerhalb eines Testzeitraums:

$$\eta_{\mathrm{g}} = \frac{\#GFF}{\#FF} \tag{7}$$

Sicherheit S und zu erwartende (mittlere) Zeit bis zu nächsten sicherheitsgefährdenden FF $MTTF_{\mathrm{S}}$ sind um den Kehrwert von η_{g} höher als Zuverlässigkeit bzw. zu erwartenden Zeit bis zur nächsten FF:

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

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

Die Sicherheit eines System lässt sich erhöhen durch

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

Beispiel: Sicherheit durch Zusatzsteuergerät

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

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

Lösung Aufgabenteil 1

Zuverlässigkeit nach Gl. 4:

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

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

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

Betriebssicherheit:

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

Lösung Aufgabenteil 2

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

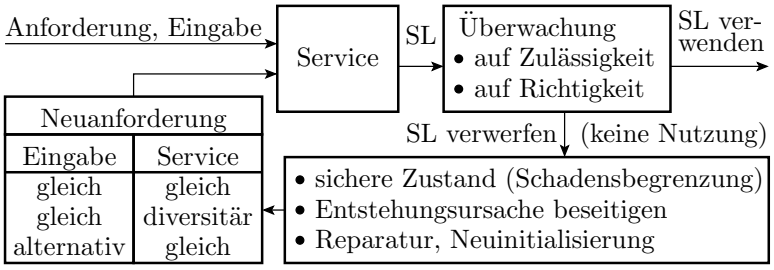
$$S_\text{MSG} \geq 5 \cdot S$$

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

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

3 FF-Behandlung

FF-Behandlung



Robuste Reaktion auf FF:

- Überwachung Eingaben, innerer Zustände und SL auf Zulässigkeit oder Richtigkeit,
- bei FF definierten (sicheren) Zustand herstellen (Robustheit).

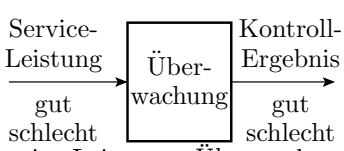
Fehlertoleranz:

- Beseitigung FF-Ursache: Reparatur, Neuinitialisierung
- Wiederholung, Erfolgskontrolle,... ‚Problem »FF durch Fehler«

3.1 Kenngrößen

Kenngrößen der Überwachung

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



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

Mögliche FF der Service-Leistung »Überwachung«:

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

$$FFC = \frac{\#EFF}{\#FF} \tag{8}$$

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

2. Phantom-FF, vermeindliches Erkennen nicht vorhandener FF. Aus Phantom-FF können bei der Korrektur echte FF werden.Kenngröße Phantom-FF-Rate:

$$\zeta_{\text{Phan}} = \frac{\#PFF}{\#SL} \tag{9}$$

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

Beispiel 3. System: FF-Rate ohne Überwachung $\zeta_{\text{OÜ}} = 1\,\%\,^{\text{FF}}/_{\text{SL}}$, Kenngrößen der Überwachung: $FCC = 80\%$, $\zeta_{\text{Phan}} = 2\,\%\,^{\text{PFF}}/_{\text{SL}}$.

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

$$\zeta_{\text{MÜ. min}} = \zeta_{\text{OÜ}} \cdot (1 - FFC) = 0{,}2\,\%\,^{\text{FF}}/_{\text{SL}}$$

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

$$\zeta_{\text{MÜ. max}} = \zeta_{\text{MÜ}} + \zeta_{\text{Phan}} = 2{,}2\,\%\,^{\text{(P)FF}}/_{\text{SL}}$$

Robustheit und Fehlertoleranz

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

$$ROB = \frac{\#FFR}{\#FF} \tag{10}$$

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

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

$$FT = \frac{\#FFT}{\#FF} \tag{11}$$

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

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

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

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

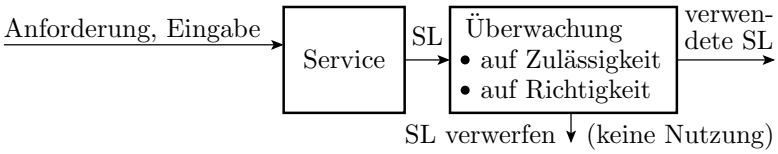
$$Z_{\text{FB}} = \frac{Z}{1 - FT}$$

Fehlertoleranz setzt Robustheit und Robustheit Nachweisbarkeit der FF voraus:

$$FT \leq ROB \leq FFC$$

3.2 Überwachungsverfahren

Überwachungsverfahren und ihre Eigenschaften



Service-Eingaben und Service-Leistungen bestehen aus:

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

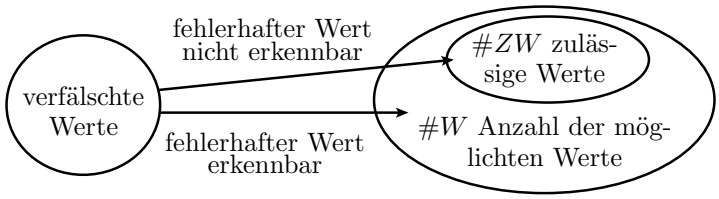
Einteilung der Überwachungsverfahren:

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

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

Kontrolle auf Zulässigkeit und Datenredundanz

Fehlererkennende Codes, Prüfkennzeichen, Wertebereichskontrolle, ...



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

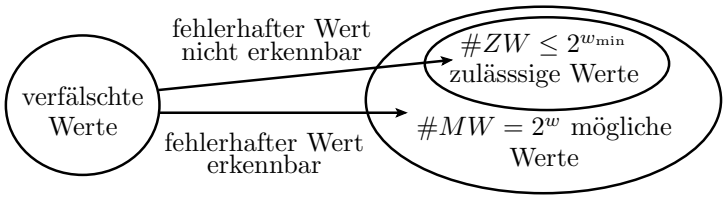
$$FFC \approx 1 - \frac{\#ZW}{\#W}$$

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

Redundante Bits

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

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



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

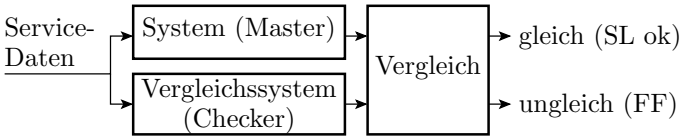
$$FFC \gtrapprox 1 - 2^{-r}$$

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

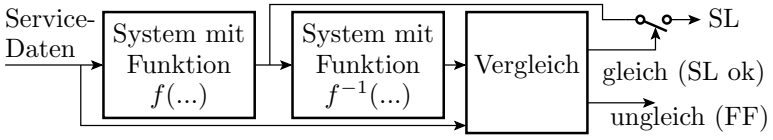
Bei angenommen $w_{\min} = 10^3$ kein nennenswerte Zusatzaufwand.

Verfahren zur Kontrolle auf Richtigkeit

1. Verdopplung und Vergleich:

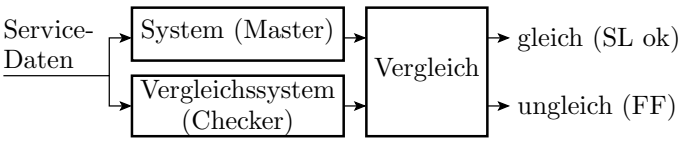


2. Eingaberückgewinnung aus der SL und Vergleich, z.B. Überwachung Versenden durch Empfang und Vergleich der empfangenen mit den Sendedaten:



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

Eigenschaften »Verdopplung und Vergleich«



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

$$Div = \frac{\#DFF}{\#FF} \tag{12}$$

($\#DFF$ – Anzahl der Master-FF, bei denen eine Wiederholung zum korrekten Ergebnis oder zu einer geänderten FF führt). FF-Überdeckung:

$$FFC \approx Div$$

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

$$\zeta_{\text{Phan}} \approx \zeta_{\text{VS}} \cdot (1 - Div)$$

Diversität von Software-Versionen

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

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

Ursprüngliche euphorische Meinung, dass so Diversität gegenüber allen Fehlern, außer denen in der Spezifikation erzielbar sei, nicht bestätigt. Die direkte oder indirekte Kommunikation der Entwicklungsteams über die Interpretation der Spezifikation, während des Test etc. trägt Gemeinsamkeiten in die Entwürfe. Neigung von Menschen, gewisse Fehler zu wiederholen, ... Erzielbare Divisität laut[6]

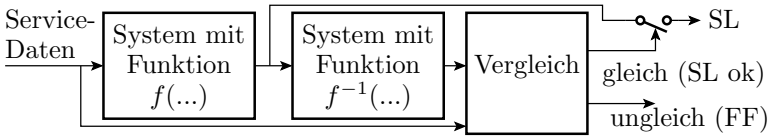
$$Div \approx FFC \leq 90\%$$

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

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

Eigenschaften »Eingaberückgewinnung«

- Eingaberückgewinnung und Vergleich:



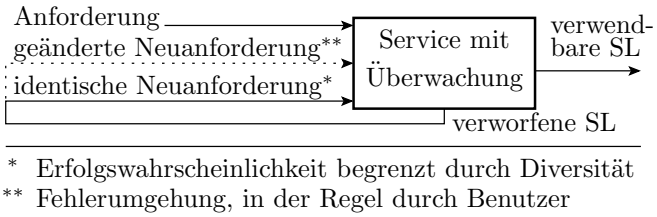
  - Höhere natürliche Diversität als Mehrfachberechung und Vergleich, weil $f(\ldots)$ und $f^{-1}(\ldots)$ in der Regel unterschiedliche, getrennt zu entwerfende Algorithmen sind.
  - Nur einsetzbar, wenn, $f(\ldots)$ eine umkehrbar eindeutige Abbildung ist. Besonders geeignet, wenn $f^{-1}(\ldots)$ viel einfacher als $f(\ldots)$ ist, z.B Wurzel $\Leftrightarrow$ Quadrat.

- Überwachung Korrektheitskriterium:
  - FF-Überdeckung und Phantom-FF-Rate ergeben sich aus der Erfolgsrate der Suche sowie der FF-Überdeckung und Phantom-FF-Rate der Kontrollfunktion.

Wenn einsetzbar, gute Überwachungsverfahren, aber ...

3.3 Korrekturverfahren

Wiederholung nach erkannter FF

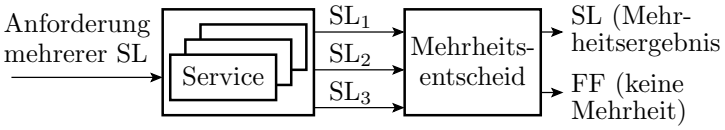


Fehlertoleranz (Erfolgshäufigkeit der Korrektur) bei identischer Neuanforderung:

$$FT \approx Div$$

(Div – Diversität), bei Fehlern als Hauptursache für FF gering. Geänderte Service-Anforderung (Fehlerumgehung[7]) erhöht die Diversität, ist aber schwer zu automatisieren.

Mehrfachberechung und Mehrheitsentscheid



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

Drei getrennte Rechner[8]:

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

Unterschiedliche Rechnertypen, unterschiedliche Betriebssysteme:

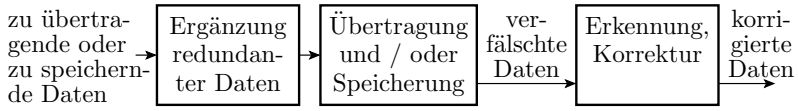
- auch Diversität gegenüber Hardware-Entwurfsfehlern und Fehlern im Betriebssystem.

Unterschiedliche ... (siehe später Foliensatz 4)

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

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

Fehlerkorrigierende Codes



- Einsatz für die Korrektur von Einzelbit- und Burst-Fehlern nach Datenübertragung und Speicherung, auch für RAIDs (siehe später Foliensatz 4).
- Höhere Datenredundanz als fehlererkennde Codes.
- Gute Lösung für die Korrektur gespeicherter oder empfangener Daten. Für andere SL ungeeignet.
- Der Anteil *FT* der korrigierbaren FF hängt vom Code, der Anzahl der redundaten Datenbits und den zu erwartenden Verfälschungen ab.

4 Fehlerbeseitigung

4.1 Ursachen von FF

Ursachen von Fehlfunktionen

Fehler:

- Entstehen mit dem System oder bei der Fehlerbeseitigung.
- Oft deterministisches, d.h. bei Wiederholung mit denselben Eingaben gleiches Fehlverhalten.
- Beseitungungserfolg kontrollierbar durch Testwiederholung.
- Fehlerbehandlung schwierig, da FF in der Regel nicht durch Wiederholung derselben SL mit demselben System korrigierbar.

Störungen:

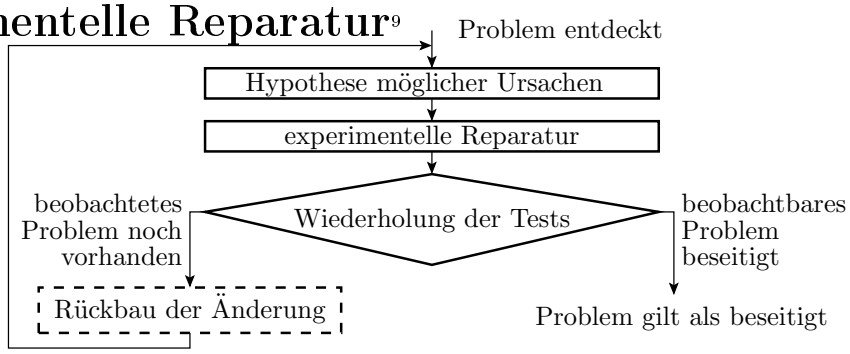
- Zufällige, nicht reproduzierbare Ursache-Wirkungs-Beziehungen.
- Fehlerbehandlung einfach, da FF durch Wiederholung derselben SL mit demselben System korrigierbar.
- Störquellen sind schwerer als Fehler zu beseitigen, allein weil sich der Beseitigungserfolg nicht durch Testwiederholung kontrollieren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

Ausfälle:

- während des Betriebs entstehende Fehler.

4.2 Experimentelle Reparatur

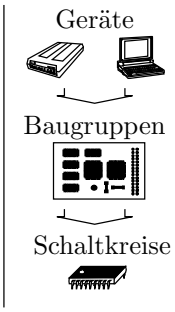
Experimentelle Reparatur[9]



[9] Grundprinzip der Problembeseitigung für reproduzierbare Probleme.

- Iteration aus Beseitigungsversuchen für hypothetische Fehler und Erfolgskontrolle durch Testwiederholung.
- Beseitigt alle vom Test nachweisbaren Fehler. Bei Reparaturversuchen können aber neue Fehler entstehen. Deshalb nach jedem erfolglosem Reparaturversuch Rückbau.
- Im Idealfall, wenn der Test keine Phantomfehler[10] ausweist und bei der Reparatur keine neuen Fehler entstehen, ist die Fehlerbeseitigungswahrscheinlichkeit gleich der Nachweiswahrscheinlichkeit des Tests.
- Abschätzungen unter Einbeziehung der bei Reparaturversuchen entstehenden Fehler und der Wirkung von Phantomfehlern später nach der themenspezifischen Einführung in die Stochastik.

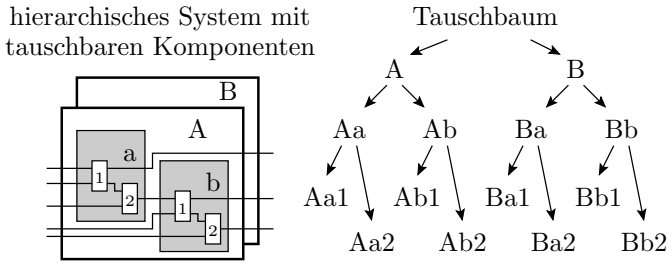
Reparatur bei wenig tauschbaren Komponenten



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:



Typisches Mechanikervorgehen:

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

Voraussetzungen:

- Wiederholbare Tests, die den Fehler nachweisen.
- Ausreichend Ersatzteile. Allgemeine Mechnikerkenntnisse[11].

Wichtig ist der Rückbau nach jedem erfolglosen Reparaturversuch. Warum?

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

[10] Phantomfehler: Vermeindlicher Fehler, die kein Fehler ist.
[11] Verständnis der Funktion des zu reparierenden Systems nicht zwingend.

4.3 Fehlerdiagnose

Fehlerdiagnose

Bestimmung von Ort- und Ursache eines Fehlers.

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

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

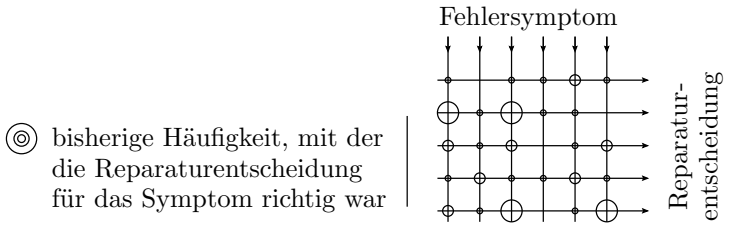
Die wichtigsten Fehlerlokalisierungstechniken:

- Ausnutzung des Parato-Prinzips.
- Rückverfolgung.

Pareto-Prinzip[12]

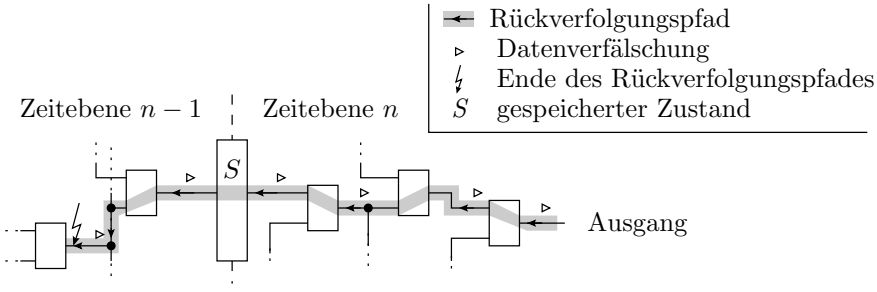
Produkte haben Schwachstellen. Die meisten Probleme geht auf einen kleinen Anteil der möglichen Ursachen zurück:

- Zählen Versuche und Erfolge unterschiedl. Reparaturalternativen.
- Bei Reparatur, Beginn mit den erfolgsversprechendsten Möglichkeit.



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

Rückverfolgung



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

[12] 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.

4.4 Test

Testen

Verfahren zum Aufspühren von Fehlern. Grundeinteilung:

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

Statisch kontrollierbare Merkmale

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

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

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

Kenngrößen von Tests

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

- Nichterkennen von Fehlern. Kenngröße Fehlerüberdeckung (Fault Coverage, Anteil der nachweisbaren Fehler):
$$FC = \frac{\#EF}{\#F} \tag{13}$$
($\#EF$ – Anzahl der nachweisbaren Fehler; $\#F$ – Anzahl der vorhandenen Fehler).
- Phantomfehler. Klassifizierung fehlerfreier Testergebnisse als fehlerhaft. Kenngröße Phantomfehlerrate[13]:
$$\varphi_{\text{Phan}} = \frac{\#PF}{\#T} \tag{14}$$
($\#PF$ – Anzahl der Phantomfehler, $\#T$ – Anzahl der durchgeführten Tests).

Testauswertung, Auswahl der Testbeispielen

Die von Tests kontrollierten Merkmal werden in der Regel durch Vergleich mit Sollwerten kontrolliert:

- Maskierungen von Fehlern durch Vergleichs-FF und
- Phantomfehler durch Tests mit falschen Sollwerten, ...

werden im Weiteren vernachlässigt.

Tests kontrollieren immer nur eine winzige Stichprobe der zugesicherndernden Merkmale und SL. Die FC hängt vom Umfang und der Auswahl der Testbeipiele ab.

Strategien der Testauswahl:

- Statische Tests: in der Regel fehlerorientiert,
- Dynamische Tests: fehlerorientierte, zufällige oder Mischformen.

Fehlerorientierte Auswahl und Bewertung erfolgt in der Regel[14], mit Fehlerannahmen (Modellfehlern oder Mutationen).

[13] Definition in Analogie zur Phantom-FF-Rate.
[14] Zum Zeitpunkt der Testauswahl sind die zu findenden und nach dem Test die nicht gefundenen Fehler unbekannt.

4.5 Haftfehler

Modellfehler und Fehlermodell

Ein Modellfehler ist ein angenommener Fehler, in der Regel eine kleine Änderung in der Systembeschreibung.

Ein Fehlermodell ist ein Algorithmus zur Berechnung einer Modellfehlermenge (Menge von Änderungen oder geänderten Beschreibungen).

Bestimmung der FC (Fehlersimulation):

- Wiederhole für jeden Test
  - Bestimmung der Sollausgaben
  - Wiederhole für alle Modellfehler
    - Bestimmung der Ausgaben mit Fehler
    - Abhakten der nachweisbaren Fehler

Fehlerorientierte Testsuche:

- Wiederhole für alle Modellfehler
  - Suche Eingaben für den Nachweis

Beide Aufgaben erfordern einen großen Rechenaufwand. Lösungen für HW siehe Foliensatz F5, erste Ansätze für SW 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 Modellfehlern:

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

als initiale Modellfehlermenge und reduziert um identisch und implizit nachweisbare und redundate (nicht nachweisbare) Modellfehler.

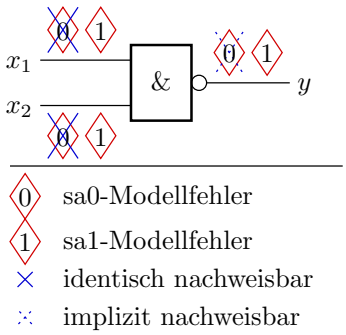
Seit 4 bis 5 Jahrzehnten am weitesten verbreitete Fehlermodell für digitale Schaltkreise.

Bei den sich aktuell entwickelnden Testauswahltechniken für Software lassen sich Parallelitäten zu den Hardware-Fehlermodellen, insbesondere dem Haftfehlermodell, aufzeigen. Es ist deshalb auch für Informatiker nützlich, das Haftfehlermodell zu kennen.

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

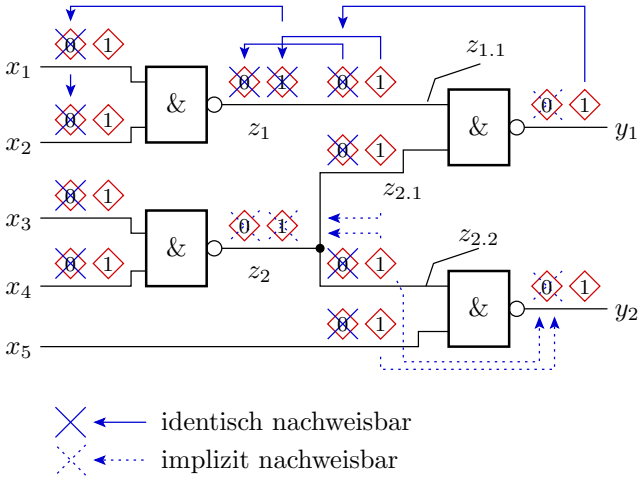
Nachweisidentität (gleiche Nachweismenge)

Nachweisimplikation

zugehörige Eingabe ist Element der Nachweismenge

Zusammenfassung identisch nachweisbarer Fehler. Optionale Streichung redundanter und implizit nachweisbarer Modellfehler. Modellierte Fehler sind ähnlich wie Transistorfehler in Gattern nachweisbar[15].

Identische und implizit nachweisbarer Fehler im Schaltungsverbund



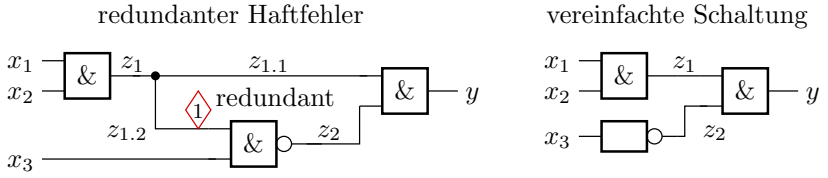
Größe der Anfangsfehlermenge:	24
Anzahl der nicht identisch nachweisbaren Fehler:	14
ohne implizit nachgewiesene Fehler:	10

	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

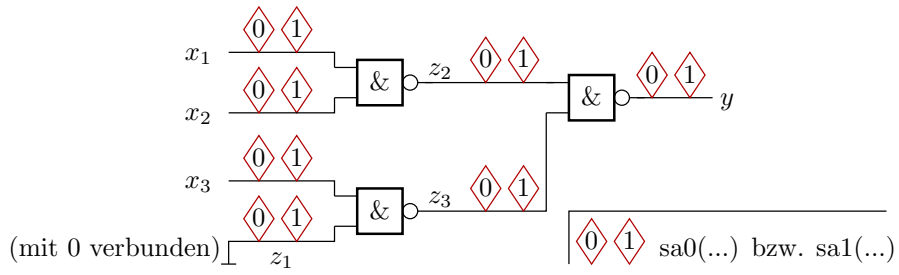
Fehler, der die Funktion des Gesamtsystems nicht beeinträchtigt.



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

Beispielaufgabe

Schaltung mit 12 eingezeichneten Haftfehlern:

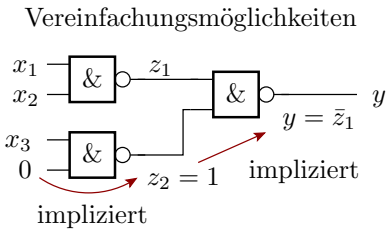


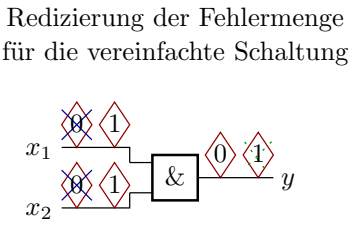
[15] Software-Mutationen wie Off-By-One, Branch-Fehler, ... lassen sich ähnlich beschreiben (siehe später Foliensatz 6).

Gesucht:

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

Lösung



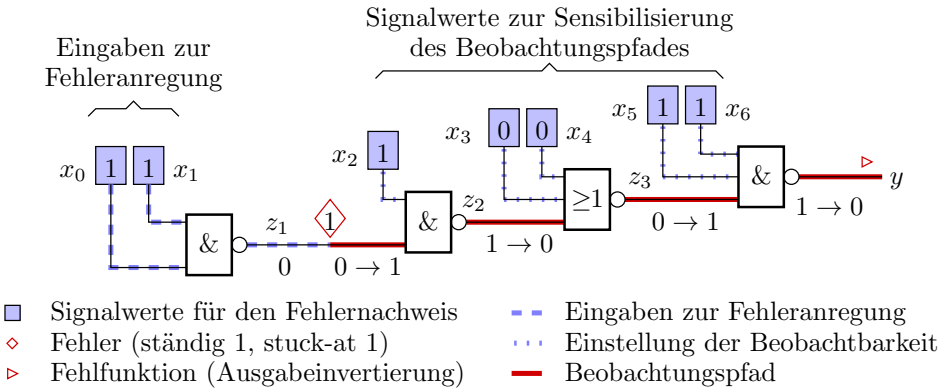


1. Die Funktion hängt nicht von x_3 ab und ist:

$$y = x_1 \wedge x_2$$

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



Suche durch »Pfadsensibilisierung« (siehe später Foliensatz 5):

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

- 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 $2^5 = 32$ möglichen Eingaben: $p_A = 2^{-2}$,
  - beobachtbar mit $2^2 = 4$ 128 möglichen Eingaben: $p_B = 2^{-5}$,
  - nachweisbar mit einer möglichen Eingaben: $p_N = p_A \cdot p_B = 2^{-7}$

4.6 Test und Zuverlässlichkeit

Fehleranzahl und FF-Rate von SW

Beispiel 4. 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?

Dieser Abschnitt wird zeigen, dass sich Zuverlässigkeit und Sicherheit proportional zur Testanzahl n und umgekeht proportional zur Anzahl der nicht beseitigten Fehler $\#F \cdot (1 - FC)$ verhalten:

$$Z \sim \frac{n}{\#F \cdot (1 - FC)};\ S = \frac{Z}{\eta_\mathrm{g}}$$

Fehlfunktionsrate durch Fehler

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

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

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

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

Unter den Annahmen, dass:

- die Fehler beim Test dieselbe FF-Rate wie in der Anwendung haben und
- alle nachweisbaren Fehler beseitigt werden

ist ζ_max der nicht beseitigten Fehler:

$$\zeta_\mathrm{max} \approx \frac{1}{n}$$

(n – Testsatzlänge). Abschätzbare Obergrenze der FF-Rate durch alle nicht beseitigten Fehler

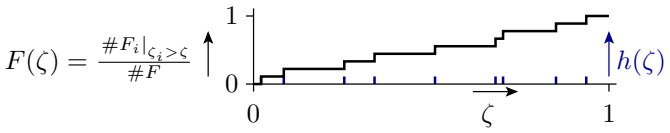
$$\zeta_\mathrm{F} \leq \#F \cdot (1 - FC) \cdot \zeta_\mathrm{max} \approx \frac{\#F \cdot (1 - FC)}{n}$$

$\#F$ – Anzahl entstandene Fehler; FC – Fehlerüberdeckung.

Mindestzuverlässigkeit und Sicherheit, wenn FF durch Störungen vernachlässigbar sind:

$$Z \geq \frac{n}{\#F \cdot (1 - FC)}; \qquad S = \frac{Z}{\eta_\mathrm{g}}$$

Verteilungsfunktion und Dichte der FF-Rate



$F(\zeta)$: Verteilungsfunktion FF-Rate, Anteil Fehler mit FF-Rate $\leq \zeta$.
$h(\zeta)$: Dichte FF-Rate: Anteil Fehler mit FF-Rate ζ.

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

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

Mit der Dichte der FF-Rate statt des Maximums den Mittelwert der FF-Rate je Fehler und damit der gesamten FF-Rate abschätzen:

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

Abschätzung der Verteilung der FF-Rate

Unter der vereinfachten Annahme, dass ein Zufallstest der Länge n alle Fehler mit einer FF-Rate $\zeta \geq \frac{1}{n}$ nachweist[16], ist der Anteil der nicht nachweisbaren Fehler:

$$1 - FC(n) = F\left(\zeta = \frac{1}{n}\right) = \int_0^{\frac{1}{n}} h(\zeta) \cdot \mathrm{d}\zeta$$

Bei einem Zufallstest erfordert eine Verringerung von $1 - FC(n)$ um eine Dekade einen mehr als eine bis mehrere Dekaden längeren Testsatz. Potenzfunktion:

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

k	1	0,5	0,33	0,25
$\frac{n}{n_0}$ für $1 - FC(n) = 0{,}1$	10	100	10^3	10^4

n_0 – Bezugstestsatzlänge für $FC = 0$; n – Testsatzlänge incl. n_0.

Aus $1 - FC(n) \approx (n/n_0)^{-k}$ und $1 - FC(n) = F\left(\zeta = \frac{1}{n}\right)$ folgt:

$$F(\zeta) \approx (\zeta \cdot n_0)^k \quad \text{mit } 0 \leq \zeta < \frac{1}{n_0} \text{ und } 0 < k < 1$$

Dichte der FF-Rate, wenn die mit n_0 Tests erkennbaren Fehler beseitigt werden:

$$h(\zeta) = \frac{\mathrm{d}F(\zeta)}{\mathrm{d}\zeta} \approx k \cdot n_0^k \cdot \begin{cases} \zeta^{k-1} & 0 \leq \zeta < \frac{1}{n_0} \text{ und } 0 < k < 1 \\ 0 & \text{sonst} \end{cases}$$

Wenn die mit allen n Tests erkennbaren Fehler beseitigt werden:

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

[16] Diese Annahme ist eine grobe Näherung, führt aber auf fast dasselbe Ergebnis wie die spätere genaue Modellierung der Nachweislänge auf Foliensatz 3.

Abnahme der mittleren FF-Rate:

$$\mathrm{E}\left[\zeta,\mathrm{n}\right] \approx \int_0^{\frac{1}{n}} \zeta \cdot \underbrace{k \cdot n^k \cdot \zeta^{k-1}}_{h(\zeta,n)} \cdot \mathrm{d}\zeta = k \cdot n^k \cdot \int_0^{\frac{1}{n}} \zeta^k \cdot \mathrm{d}\zeta$$
$$= \frac{k \cdot n^k}{k+1} \cdot \left(\left(\frac{1}{n}\right)^{k+1} - 0^{k+1} \right) = \frac{k}{(k+1) \cdot n}$$

Abnahme der Fehleranzahl:

$$\#F\left(n\right) = \#F \cdot \left(1 - FC\left(n\right)\right) \approx \#F\left(n_0\right) \cdot \left(\frac{n}{n_0}\right)^{-k}$$

Abnahme der FF-Rate durch alle Fehler:

$$\begin{aligned}\zeta_{\mathrm{F}}\left(n\right) &= \#F\left(n\right) \cdot \mathrm{E}\left[\zeta,\mathrm{n}\right] \\ &\approx \frac{\#F\left(n\right) \cdot k}{(k+1) \cdot n} = \#F\left(n_0\right) \cdot \left(\frac{n}{n_0}\right)^{-k} \cdot \frac{k}{(k+1) \cdot n} \\ &= \underbrace{\frac{\#F\left(n_0\right) \cdot k}{(k+1) \cdot n_0}}_{\zeta_{\mathrm{F}}(n_0)} \cdot \left(\frac{n}{n_0}\right)^{-(k+1)} = \zeta_{\mathrm{F}}\left(n_0\right) \cdot \left(\frac{n}{n_0}\right)^{-(k+1)}\end{aligned}$$

Abnahme der FF-Rate

- proportional zur Anzahl der nicht beseitigten Fehler und umgekehrt proportional zur Anzahl der Tests n (incl. n_0),
- mit einem um eins größeren Exponenten k mit n.

Beispiel

Um welchen Faktor verringern sich Fehleranzahl und FF-Rate durch Fehler, wenn die Anzahl der dynamischen Tests verdreifacht wird, für $0{,}3 \leq k \leq 0{,}5$?

$$\frac{\#F\left(3 \cdot n_0\right)}{\#F\left(n_0\right)} \approx 3^{-k}; \quad \frac{\zeta_{\mathrm{F}}\left(3 \cdot n_0\right)}{\zeta_{\mathrm{F}}\left(n_0\right)} \approx 3^{-(k+1)}$$

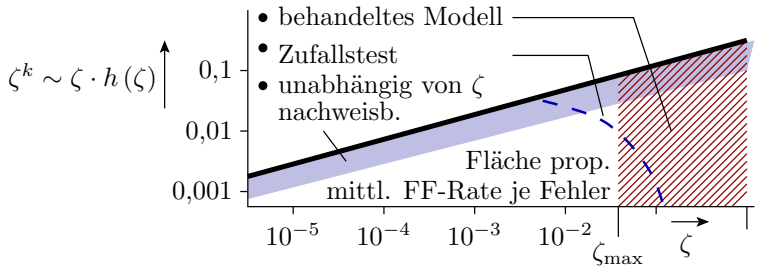
	$k = 0{,}3$	$k = 0{,}5$
$\frac{\#F(3 \cdot n_0)}{\#F(n_0)}$	0,72	0,56
$\frac{\zeta_{\mathrm{F}}(3 \cdot n_0)}{\zeta_{\mathrm{F}}(n_0)}$	0,24	0,19

Eine Ergebnissinterpretation hierzu:

- Welcher Zuverlässigkeitsgewinn ist zu erwarten, wenn das Personal der Testabteilung verdreifacht wird?
- Dreifacher Testaufwand verringert die FF-Rate auf 19% bis 24%, also 4 bis 5-fache Zuverlässigkeit.

Zusammenhang $\zeta_{\mathrm{F}}\left(n\right)$als Graphik

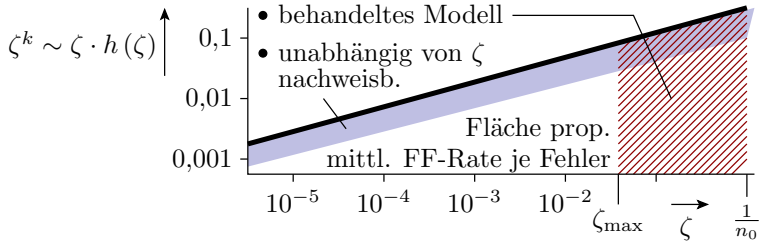
$$\zeta_{\mathrm{F}}\left(n\right) \approx \underbrace{\#F\left(n_0\right) \cdot \left(\frac{n}{n_0}\right)^{-k}}_{\#F(n)} \cdot \underbrace{\int_0^{\frac{1}{n}} \zeta \cdot \underbrace{k \cdot n^k \cdot \zeta^{k-1}}_{h(\zeta,n)} \cdot \mathrm{d}\zeta}_{\text{mittl. FF}-\text{Rate je Fehler}} = \underbrace{\#F\left(n_0\right) \cdot k \cdot n_0^k}_{\text{konstant}} \cdot \int_0^{\frac{1}{n}} \zeta^k \cdot \mathrm{d}\zeta$$



Beseitigung

- alle Fehler ab $\zeta \geq \zeta_{max}$: bisher behandelt für $\zeta_{max} = \frac{1}{n}$
- unabhängig von ζ: typ. für statische Tests
- mit einer Wahrscheinlichkeit $p(\zeta)$: Zufallstest, siehe Foliensatz F3

Statische und fehlerorientiert gesuchte Tests



- Reduzierung der Fehleranzahl durch statische Tests:

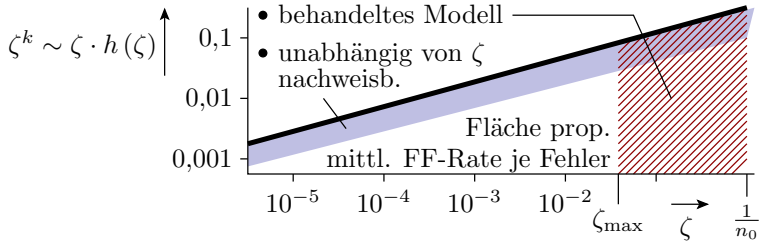
$$\#F_{ST} = \#F \cdot (1 - FC_{ST})$$

- Fehlerorientiert gesuchte Tests haben eine viele höhere Fehlerüberdeckung als gleichlange Zufallstests:

$$\#F(n_0) = \#F_{ST} \cdot (1 - FC_{FO}) \quad \text{mit } (1 - FC_{FO}) \ll \left(\frac{n_{FO}}{n_0}\right)^{-k}$$

  n_{FO} – Anzahl fehlerorientiert gesuchte Tests.

Effektive Testsatzlänge



Die Testsatzlänge n, ab der (im Mittel) alle Fehler mit $\zeta_i \geq \zeta_{max}$ beseitigt werden, kann um einen Faktor c von der tatsächlichen Testanzahl abweichen:

$$\zeta_{max} \approx \frac{1}{c \cdot n}$$

- FF-Rate für Fehler im Mittel c-mal so groß wie für Modellfehler,
- Bei FF Beseitigung nur mit Wahrscheinlichkeit c, ...

Erweitertes Modell incl. statische Tests, ...

Fehleranzahl nach statischen und fehlerorientiert gewählten Tests:

$$\#F\left(n_0\right) = \#F \cdot \left(1 - FC_{\mathrm{ST}}\right) \cdot \left(1 - FC_{\mathrm{FO}}\right)$$

$\#F$ – Fehleranzahl vor allen Tests; FC_{ST} – Fehlerüberdeckung des statischen Tests; FC_{FO} – Fehlerüberdeckung der fehlerorientiert gesuchten Tests. Fehleranzahl nach weiteren n_{RT} Zufallstetst:

$$\#F\left(n\right) \approx \#F\left(n_0\right) \cdot \left(\frac{n}{n_0}\right)^{-k} \qquad n_0 = c \cdot n_{\mathrm{FO}},\; n = n_0 + c \cdot n_{\mathrm{RT}}$$

n – effektive Testsatzlänge; n_0 – Bezugstestsatzlänge; n_{RT} – Anzahl der Zufallstests. FF-Rate vor dem Zufallstests:

$$\zeta_{\mathrm{F}}\left(n_0\right) \approx \frac{k \cdot \#F\left(n_0\right)}{(k+1) \cdot n_0}$$

FF-Rate nach weiteren n_{RT} Zufallstetst:

$$\zeta_{\mathrm{F}}\left(n\right) \approx \frac{k \cdot \#F\left(n\right)}{(k+1) \cdot n} = \zeta_{\mathrm{F}}\left(n_0\right) \cdot \left(\frac{n}{n_0}\right)^{-(k+1)}$$

Zuverlässigkeit und Sicherheit Produktfreigabe

- In komplexen IT-Systemen sind meist Entwurfsfehler die Hauptursache für FF.
- Bei signifikannter FF-Rate durch Störungen, wird diese mit Mitteln der FF-Toleranz auf akzeptable kleine Werte abgesenkt (z.B. Übertragung mit PKZ und Wiederholung nach erkannten FF) .
- Die in der Regel genutzten Mittel der FF-Toleranz, wie Überwachung + Wiederholung nach FF, bieten Robustheit, aber kaum Fehlertoleranz für FF durch Fehler.

Abschätzung für die Zuverlässigkeit:

$$Z \approx \frac{1}{\zeta_{\mathrm{F}}\left(n\right)} \approx \frac{(k+1) \cdot n}{k \cdot \#F\left(n\right)} = \frac{(k+1)}{k \cdot \#F\left(n_0\right)} \cdot \frac{n}{n_0^k}^{k+1} = Z\left(n_0\right) \cdot \left(\frac{n}{n_0}\right)^{k+1}$$

Abschätzung für die Sicherheit:

$$S = \frac{Z}{\eta_{\mathrm{g}} \cdot \left(1 - ROB\right)}$$

η_{g} – Anteil gefährdende FF; ROB – Anteil der FF, die das System erkennt und robust darauf reagiert.

4.7 Reifeprozesse

Das Problem immer größerer IT-Systeme

Anzahl der entstehenden nicht durch statische oder fehlerorientiert gewählte Tests nachweisbare Fehler:

$$\#F\left(n_0\right) \approx \#N \cdot \eta_{\mathrm{E}} \cdot \left(1 - FC_{\mathrm{ST}}\right) \cdot \left(1 - FC_{\mathrm{FO}}\right) \sim \#N$$

N– Anzahl der Netto-Codezeilen; $\eta_{\mathrm{E}} \approx 10 \ldots 100$ – entstehende Fehler je 1000 NLOC (Netto Lines of Code); FC_{ST}, FC_{FO} – Anteil der von den statischen bzw. fehlerorientiert gesuchten Tests nachweisbaren Fehler. Die Systemgröße $\#N$ verdoppelt sich alle paar Jahre, $\eta_{\mathrm{E}} \cdot \left(1 - FC_{\mathrm{ST}}\right) \cdot \left(1 - FC_{\mathrm{FO}}\right)$ bleiben etwa gleich. Damit die Zuverlässigkeit nach Gl.

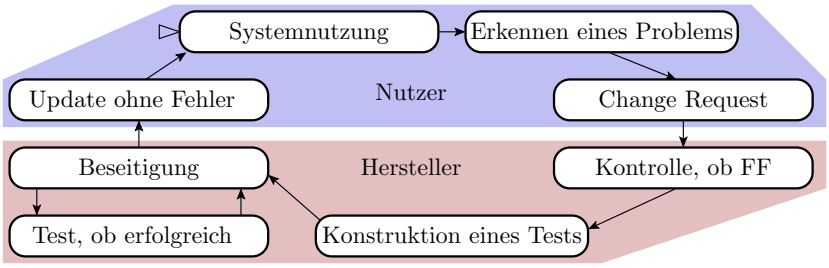
$$Z \approx \frac{(k+1)}{k \cdot \#F\left(n_0\right)} \cdot \frac{n}{n_0^k}^{k+1} \sim \frac{n^{k+1}}{\#N} \quad \text{mit } 0 < k < 1$$

nicht sinkt, muss die effektive Testsatzlänge mit den wachsenden Programmgrößen $\#N$ mitwachsen.

Alternative Reifeprozess: Fortsetzung der Iteration aus Test und Fehlerbeseitigung mit den Nutzern als Tester.

Reifeprozesse

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



Fehlerbeseitigungsiteration für FF bei den Anwendern:

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

1. Bei einer vermuteten Fehlfunktion stellt der Nutzer einen Änderungsanforderung (Change Request). Alternativ sendet das System einer FF-Report. FF-Reports werden in Schubladen vermutlich gleicher Ursache gesammelt.
2. Der Hersteller bevorzugt für die Beseitigung Schubladen mit schwiegenden und vielen FF.
3. Suche von Tests, die die FFs reproduzierbar anregen. Testbeispiele dienen zur Fehlerlokalisierung und Erfolgskontrolle.
4. Experimentelle Reperatur, Einspielen der Änderung über Updates.

Die Schritte 1 bis 3 haben Erfolgswahrscheinlichkeiten deutlich unter 1. Ein Fehler wird im Mittel erst, wenn er viele FF verursacht hat, beseitigt.

Erhöhung der effektiven Testanzahl n um $c \cdot n_\mathrm{R}$ auf:

$$n = n_\mathrm{T} + c \cdot n_\mathrm{R} \quad \text{mit } c \ll 1$$

c – Anteil der FF, für der verursachende Fehler erfolgreich beseitigt wird; n_R – Summe der SL bei allen Nutzern. Für Systemen mit vielen Nutzern nach längerer Nutzungsdauer:

$$c \cdot n_\mathrm{R} \gg n_\mathrm{T}$$

Sehr hohe Zuverlässigkeiten, die auf anderem Weg nicht erzielbar sind.

Zuverlässigkeit gereifter Systeme

In einem Reifeprozess sinken Fehleranzahl und FF-Rate durch Fehler

$$\#F\,(n) \approx \#F\,(n_0) \cdot \left(\frac{n}{n_0}\right)^{-k}$$

$$\zeta_\mathrm{F}\,(n) \approx \zeta_\mathrm{F}\,(n_0) \cdot \left(\frac{n}{n_0}\right)^{-(k+1)}$$

mit der effektiven Testsatzlänge:

$$n = n_\mathrm{T} + c \cdot n_\mathrm{R}$$

und einer Bezugstestsatzlänge $n_0 = n_\mathrm{T} + c \cdot n_\mathrm{R0}$ für eine Reifedauer n_R0 (n_T – Anzahl Herstellertests; c – Anteil FF, für der verursachende Fehler beseitigt wird).

Für Überschläge mit $c \cdot n_\mathrm{R} \gg n_\mathrm{T}$, $c \cdot n_\mathrm{R0} \gg n_\mathrm{T}$ und $\frac{n_\mathrm{R}}{n_\mathrm{R0}} \approx \frac{t_\mathrm{R}}{t_\mathrm{R0}}$:

$$\frac{n}{n_0} \approx \frac{n_\mathrm{R}}{n_\mathrm{R0}} \approx \frac{t_\mathrm{R}}{t_\mathrm{R0}}$$

($t_\ldots$ – Reifedauer in Zeiteinheiten).

$$Z(n) \approx Z(n_0) \cdot \left(\frac{n}{n_0}\right)^{k+1}$$
$$Z(n_R) \approx Z(n_{R0}) \cdot \left(\frac{n_R}{n_{R0}}\right)^{k+1}$$
$$Z(t_R) \approx Z(t_{R0}) \cdot \left(\frac{t_R}{t_{R0}}\right)^{k+1}$$

Hohe Zuverlässigkeit verlangt viele Nutzer, lange Reifezeit und einen hohen Anteil der FF bei Nutzern, für die die verursachenden Fehler beseitigt werden.

Systeme, die viele Jahre gereift sind, haben hohe, auf anderem Wege unerreichbare Zuverlässigkeiten. Schwer ersetzbar durch neue Systeme (siehe Jahr2000-Problem).

Neue / alternative Systeme sind in den ersten Nutzungsjahren vielfach viel unzuverlässer als die Systeme, die sie ersetzen. Wenn das die Akzeptanz beeinträchtigt, reifen sie auch nicht ...

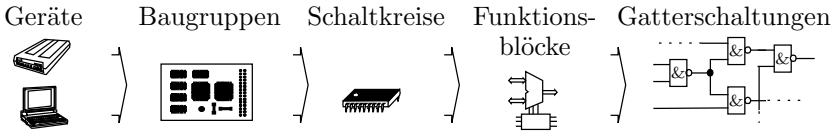
Lernprozesse der Benutzer

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

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

4.8 Modularer Test

IT-Systeme sind modular aufgebaut



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

Modularität ist wichtig für ...

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

Effektive Testsatzlänge von Zufallstests

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

- nur ein kleiner Teil der Gesamt-SL nutzt sie als Teil-SL.
- Nur ein kleiner Teil der lokalen FF verursacht Gesamt-FF.
- Die FF-Raten durch diese Fehler sind beim isolierten Modultest $c \gg 1$ mal größer als im Gesamtsystem.
- Bei Beseitigung der vom Modultest nachweisbaren Fehler sinkt die FF-Rate schlecht nachweisbarer Fehler in den Modulen etwa wie mit der c–fachen Anzahl von Gesamttests.

Für Interaktionsfehler zwischen den Modulen werden zusätzlich Gesamttests gebraucht.

Ein guter Mix aus Modultests und Gesamttests hat eine um einen Faktor $c \gg 1$ höhere effektive Testsatzlänge als nur Gesamttests.

5 Fehlervermeidung

Fehler als FF des Entstehungsprozesses

Ein Entstehungsprozess ist auch ein Service

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

Er erbt damit auch die Kenngrößen zur Beschreibung der Verlässlichkeit:

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

und die Maßnahmen zur Sicherung der Verlässlichkeit.

Im Weiteren werden wir davon nur betrachten:

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

Fehlerentstehungsraten und -metriken

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

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

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

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

Beispiel 5. 30 Fehler / 1000 NLOC, Programm mit 2000 NLOC. Zu erwartende Anzahl der entstehenden Programmfehler: 60

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

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

5.1 Fehleranteil, Ausbeute

Fehleranteil und Ausbeute

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

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

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

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

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

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

  $$DL = \mathbb{E}[\#F] \ll 1$$

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

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

  FC_{Obj} – Anteil der erkennbaren fehlerhaften Objekte.

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

Beispiel 7. Ausbeute $Y = 95\%$, abgeschätzt mit einem Test, der $FC_{\text{Obj}} = 50\%$ der fehlerhaften Objekte erkennt. Fehleranteil der Objekte:

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

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

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

Beispiel 8. Schaltkreisausbeute $Y = 80\%$, Fehleranteil nach Test und Fehlerbeseitigung $DL_{\text{T}} = 1000\,\text{dpm}$. Gesucht FC_{Obj}.

Eine Verringerung von DL von ≈ 1 auf 10^{-3} verlangt $FC_{\text{Obj}} \approx 1$:

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

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

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

Das ist auch die typische Größenordnung der Fehlerüberdeckung von Schaltkreistests.

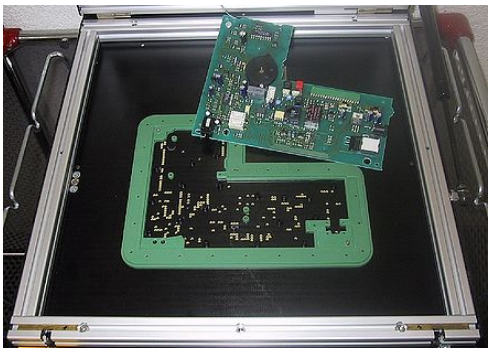
Fehleranzahl komplexer Systeme

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

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

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

Baugruppentest



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

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

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

$$DL_{\mathrm{BG.T}} = \#F_{\mathrm{BG.T}}$$

Beispiel Fehleranteil einer Baugruppe

Beispiel 9. Anzahl und Fehleranteil der Bauteile:

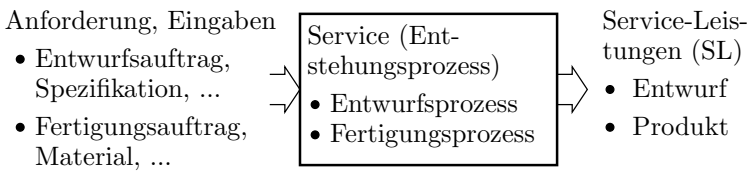
Typ	Anzahl	DL_{BT}
Leiterplatte	1	20 dpm
Schaltkreise	20	200 dpm
diskrete Bauteile	35	10 dpm
Lötstellen	560	1 dpm

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

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

5.2 Determinismus und Zufall

Fehlerentstehung

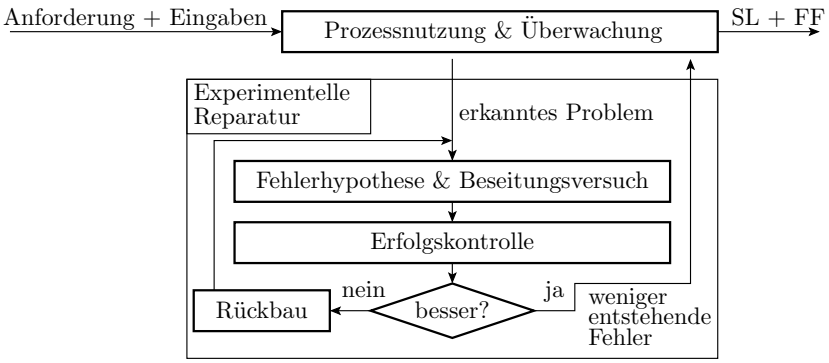


Ursachen für die Fehlerentstehung:

- Fehler: deterministische Ursache-Wirkungsbeziehung
  - beseitigbare Ursachen,
  - Erfolgskontroll durch Testwiederholung, ...
- Störungen: zufällige Ursache-Wirkungsbeziehung
  - FF durch Wiederholung beseitigbar,
  - Erfolgskontrolle 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, versuchsweise Beseitigung,
- Erfolgkontrolle durch Wiederholung der Prozessabläufe, die das Problem erkannt haben.

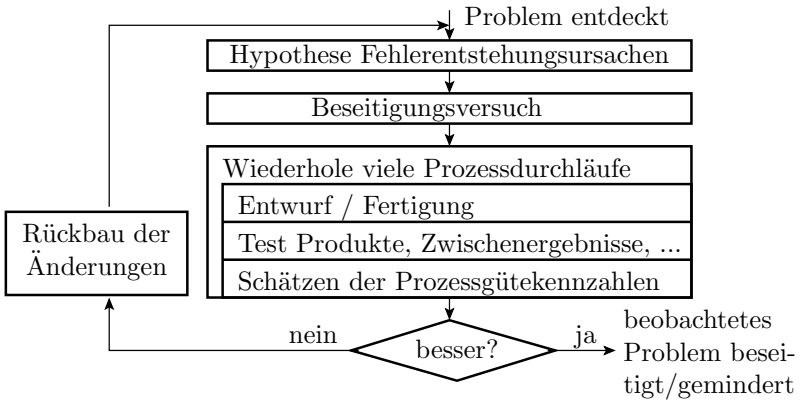
Experimentelle Reparatur und Determinismus

Determinismus bedeutet, dass das fehlerfreie System für denselben Entwurfs- oder Fertigungsauftrag (nach derselben 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 des Erfolgs eines Problembeseitigungsversuchs in der Regel eine statistisch signifikannte Stichprobe von Prozessdurchläufen zur Bestimmung von Prozessgütekennzahlen und Entscheidungen mit Irrtumswahrscheinlichkeiten.

... nicht deterministische Prozesse, Fehlerwirkungen, Störungen:

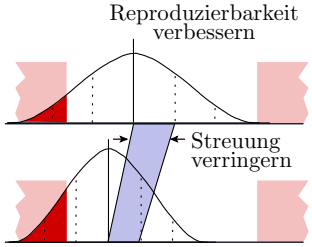


Die Beseitigung eines einzelnen Problems verlangt um Zehnerpotenzen mehr Prozessdurchläufe und Kontrollen, schlechtere Erfolgschancen, viel höheres Risiko, bei Beseitigungsversuchen neue Fehler einzubauen, die nicht durch Rückbau beseitigt werden, ...

Prozesszentrierung und -verbesserung

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

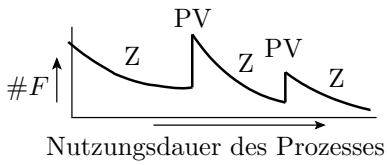
- Prozesszentrierung: Verschiebung der Verteilung mit Hilfe von Einstellgrößen in die Mitte des Toleranzbereichs.
- Prozessverbesserung: Verringerung der Streuung durch technologische Neuerungen neue Maschinen, Verfahren, ...



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

Prozessverbesserung und -zentrierung stehen stellvertretend für große Prozesseingriffe und kleine Nachbesserungen.

Sägezahnverlauf der Fehleranzahl



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

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

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

Auch bei anderen Fertigungsprozessen und Entwurfsprozessen

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

Fakt 10. *Am qualitativ hochwertigsten sind tendentiell die Produkte, die kurz vor technologischen Neuerungen entstehen. Maxima der Prozesszuverlässigkeit. (Am besten versuchen, immer solche Produkte zu bekommen.)*

Eine Schattenseite von Innovationen

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

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

Schattenseite:

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

Linux unterscheidet z.B. in seiner Versionsverwaltung:

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

5.3 Projekte, Vorgehensmodelle

Der Technologiegedanke

Technologie: Lehre von reproduzierbaren Abläufen zur Erzeugung von Produkten[17].

Technologiegedanke

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

Die technologische Entwicklung hin zur

- automatisierten menschenfreien Fertigung und
- rechnergestützten / automatisierten Entwurfsprozessen

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

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

Übertragung des Technologiegedanken auf Projekte

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

Wie verhält es sich mit Projekten:

- Manuelle kreative Teile der Entwurfsprozesse[18] und
- Fertigung von Prototypen, Demonstratoren, ... ?

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

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

Schließt das eine Fehlervermeidung aus?

Vorgehensmodelle

Vereinheitlichung des Vorgehens für große Klassen von Projekten

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

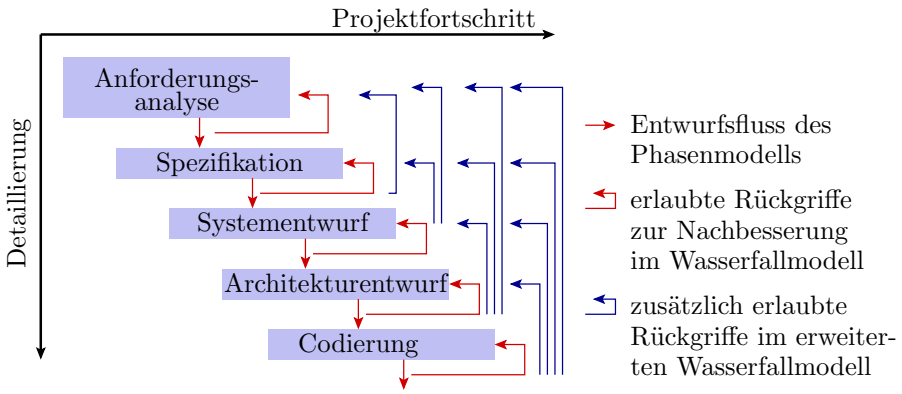
Typische Vorgehensmodelle für den Entwurf und die Fertigung von IT-Komponenten umfassen:

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

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

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

Stufenmodelle



Stufenmodelle variieren:

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

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

Gestaltbare Einflussfaktoren auf Qualität und Kosten:

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

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

Bewertung von Vorgehensmodellen

Jede Art der Fehlervermeidung benötigt eine Erfolgskontrolle:

Daraus resultierende Frage

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

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

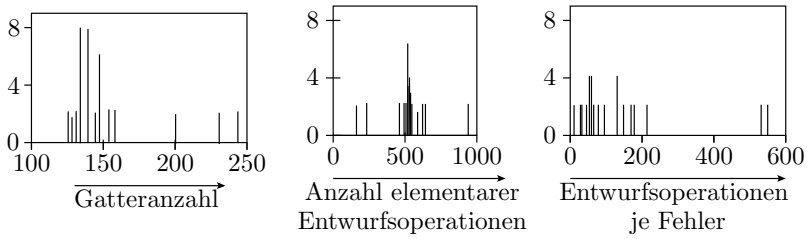
- Projektdauer, Projektkosten,
- Arbeitsschritte je entstehender Fehler, Umfrageergebnisse, ...

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

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

Ein Experiment [19]

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, ...



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

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

5.4 Qualität und Kreativität

Qualität und Kreativität

Qualität verlangt Fehlervermeidung. Fehlervermeidung verlangt Reproduzierbarkeit:

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

Kreativität verlangt »Einzigartigkeit«:

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

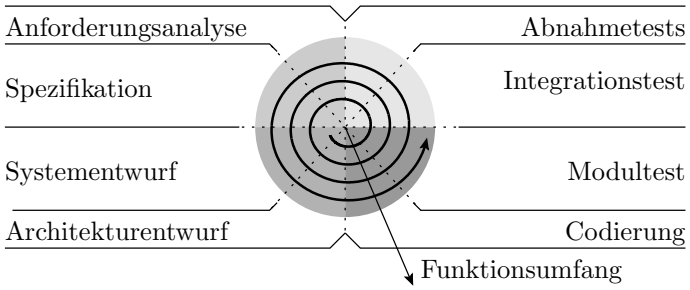
Schlussfolgerung

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

IT-Entwurf verlangt Qualität und Kreativität.

Spiralmodell als Beispiel evolutionärer Modelle

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



Aufteilung einer Entwicklung auf ein mehrmaliges Durchlaufen eines Stufenmodells.

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

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

Querverbindungen zum akademischen Alltag

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

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

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

Fehlervermeidung beschränkt die Kreativität. Sind Kreativitätsbeschränkungen für Universitäten akzeptabel?

Ein Abstecher zu Lernrozessen

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

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

Lernphasen:

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

An Universitäten:

- Phase 1: Vorlesung, Seminare, Selbststudium, ...
- Phase 2: Übung, Klausurvorbereitung[20], Praktika.
- Phase 3: Aus Zeitgründen erst in der Berufspraxis für den eigenen eingeschränkten Tätigkeitsbereich.

Querverbindung Drittmittelprojekte

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

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

Fehlerkultur

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

Positive Sichtweisen:

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

Ideale Fehlerkultur aus Sicht unserer Vorlesung:

- Erkannten Probleme beseitigen.
- Beseitigungserfolg durch Testwiederholung kontrollieren.

Die Praxis verlangt Kompromisse zwischen den Kosten zur Sicherung der Verlässlichkeit und der Bereitschaft der Kunden, dafür zu zahlen.