Test und Verlässlichkeit 2 Test, Fehlerbeseitigung und Fehlervermeidung

Prof. G. Kemnitz

6. Oktober 2025

2.1 Inhalt Foliensatz 2

Inhaltsverzeichnis

1 Test

2.2 Ursachen für Fehlverhalten (Wiederholung)

- Fehler,
- Störungen (z.B. ein zufällig invertiertes Bit),
- Ausfälle.

In unserer Modellwelt sind Fehler die beseitigbaren Ursachen für Fehlverhalten (MF, Abstürze). Ein Fehler ist praktisch das, was nicht getan wurde, um ihn abzustellen. Wenn z.B. zur Beseitigung eines beobachtbaren Problems eine Fallunterscheidung ergänzt werden musste, ist (oder war) der Fehler eine fehlende Fallunterscheidung.

Störungen sind die nicht abstellbaren Ursachen. Dafür sind verursachte Fehlverhalten durch Wiederholung korrigierbar.

Ausfälle sind Fehler, die während des Betriebs entstehen. Gefährdungsabwendung durch Fehlfunktionsbehandlung, Wartungstest, Redundanzen, ... (Abschn. 6.5).

2.3 Testen

Verfahren zum Aufspühren von Fehlern. Grundeinteilung:

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

Mit statischen Tests kontrollierbare Merkmale:

- Dokumentationen: Verständlichkeit, Vollständigkeit, ...
- Software: Syntax, statische Code-Analyse (Entwurfsregeln, Typenverträglichkeit, API-Benutzerregeln, ...).
- Leiterplatten: Widerstand entlang und zwischen Leitungen zum Ausschluss von Kurzschlüssen und Unterbrechungen.

Dynamische Tests erst am funktionierenden (Teil-) System möglich, statische Tests bereits nach einzelnen Entwurfs- und Fertigungsschritten.

IT-Systeme werden vor dem Einsatz in der Regel einer Vielzahl statischer und dynamischer Tests unterzogen.

1.1 Kenngrößen

2.4 Kenngrößen von Tests

Kein Test ist vollkommen. Jeder Test

- erkennt nur einen Teil der möglichen Fehler und
- ist selbst ein System mit begrenzter Zuverlässigkeit.

Kenngrößen zur Beschreibung der Güte von Tests:

- Fehlerabdeckung:
$$FC = \left.\frac{\#DF}{\#F}\right|_{\text{ACR}} \tag{2.1}$$
- Phantomfehlerate, Anteil der korrekten Testerausgaben, die der Test als falsch klassifiziert:
$$\zeta_{\text{PF}} = \left.\frac{\#PM}{N}\right|_{\text{ACR}} \tag{2.2}$$

FC	Fehlerabdeckung (fault coverage), Anteil der nachweisbaren Fehler.
$\#F$, $\#DF$	Fehleranzahl, Anzahl der davon nachweisbaren Fehler.
ζ_{PF}	Phantomfehlerrate des Tests.
N, $\#PM$	Testanzahl, Anzahl der Phantomfehler.
ACR	Brauchbare Schätzwerte nur bei geeigneten Zählwertgrößen.

2.5 Umgang mit Phantomfehlern

Phantomfehler, z.B. durch falsche Sollwerte bei der Kontrolle von Testausgaben,

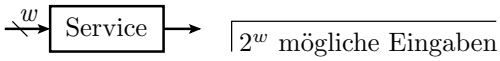
- starten überflüssige Beseitigungsiterationen,
- in denen neue nicht nachweisbare Fehler entstehen können.

Unsere idealisierte Fehlerkultur unterstellt, dass

- neu entwickelte Tests auf Phantomfehler getestet und
- erkannte Phantomfehler beseitigt werden.
- Wenn Tests einen Fehler signalisieren, zuerst Untersuchung, ob echter oder Phantomfehler.

Bei vernünftigem Umgang mit Phantomfehlern ist deren Einfluss auf die Verlässlichkeit vernachlässigbar.

2.7 Dynamische Tests



Dynamische Tests kontrollieren die Funktion nur für eine winzige Stichprobe der möglichen Eingaben.

	w	2^w	t_T
Gatter, 4 Eingänge	4	16	16 µs
ALU, 68 Eingänge	68	$3 \cdot 10^{20}$	10^7 Jahre
vier Eingabevariablen vom Typ int32_t	128	$3 \cdot 10^{38}$	10^{25} Jahre*

t_T – Testzeit, wenn jeder Einzeltest 1μs dauert.

- Die meisten Systeme verarbeiten $w \gg 100$ Eingabebits.
- Hinzu kommen oft tausende oder mehr gespeicherte Bits.

Vollständige Kontrolle mit allen Eingaben und Zuständen unmöglich!

---

w Anzahl der Eingabebits.

* Geschätzte Zeit seit dem Urknall nur $4 \cdot 10^9$ Jahre.

2.8 Testauswahl und Fehlerabdeckung

Strategien der Testauswahl:

- fehlerorientiert,
- zufällig hinsichtlich der zu erwartenden Fehler oder
- Mischformen.

Zum Zeitpunkt der Testauswahl sind die zu findenden und nach dem Test die nicht gefundenen Fehler nicht bekannt.

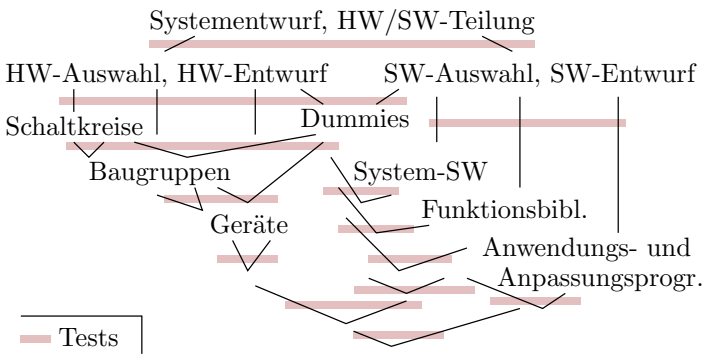
Eine nachträgliche Kontrolle der Fehlerabdeckung kann auch nur die im späteren Einsatz gefundenen und beseitigten Fehler, aber nicht die dauerhaft unerkannte geblieben zählen.

Die fehlerorientierte Auswahl und Bewertung von Tests erfolgt über Fehlerannahmen, (kleine) Änderungen (Mutationen) der Testobjektbeschreibung.

Für Fehlerannahmen ist Fehlerabdeckung exakt bestimmbar, für tatsächliche Fehler nur schätzbar. Interessanter Weise erlaubt zufällige Testauswahl genauere Vorhersagen als fehlerorientierte.

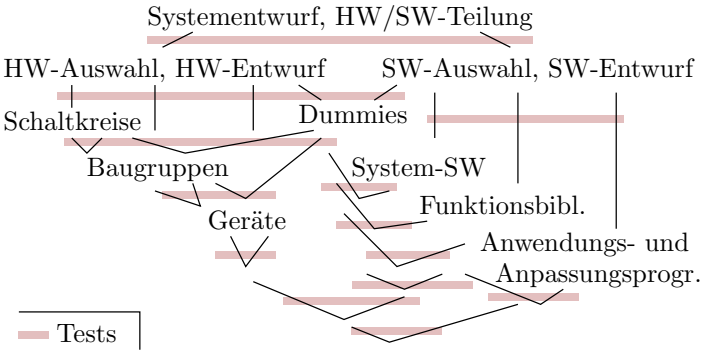
1.2 Vielfalt der Test

2.9 Entwurf und Test



Es gibt nicht den Test, sondern, ...

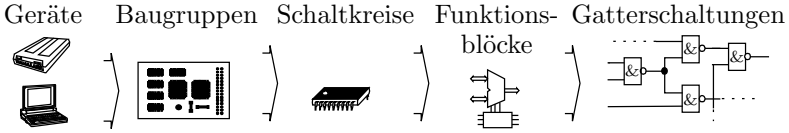
Der Entwurf eines IT-Systems ist ein komplexer Prozess, in dem ein modulares System aus HW- und SW-Bausteinen entsteht. Zwischen den Entwurfsschritten erfolgen **vielfältige statische und dynamische Tests** der entstandenen Beschreibungen.



Ein Entwurfsablauf ist idealerweise testgetrieben und strebt in jeder Entwurfsphase eine kontrollierbare Zwischenbeschreibung an. Die Entwurfsergebnisse der ersten Phasen (Sammlungen von Anforderungen, Lösungsideen, Entscheidungen) werden auf Machbarkeit, Verständlichkeit, Konsistenz, ... getestet, in der Regel statisch durch Inspektion.

Dynamische Tests sind erst möglich, wenn die Entwurfsbeschreibungen ein ausführ- oder simulierbares Ein-Ausgabeverhalten beschreibt.

2.11 Hierarchie und Test



- Rechner-Systeme bestehen aus Rechnern, EA-Geräten, Druckern, Netzwerkkomponenten, diese aus ...
- Die Hardware stellt der Software Grundfunktionen (Maschinenbefehle, Ein- und Ausgabeschnittstellen, ...) bereit.
- Software gliedert sich in Teilsysteme, Module, Bibliotheken, ...

Die durchgeführten Tests folgen der Hierarchie.

- Bauteil-, Schaltkreis-, Baugruppen- und Gerätetest.
- Modul-, Teilsystem-, Systemtest.

Da separate Tests mit weniger Aufwand höhere Fehlerabdeckungen versprechen (siehe später Abschn. 2.3.4), verwenden übergeordnete Systeme in der Regel nur gründlich getestete Bausteine und die übergeordneten Tests zielen nur noch auf Fehler im Zusammenwirken.

2.12 Ausfälle und Wartungstests

Ein **Hardware-Ausfall** in der Nutzungsphase verursacht einen neuen Fehler, der wie auch die bei der Fertigung und Reparatur entstehenden Fehler unterschiedliche Wirkung haben kann:

- komplette Funktionsuntüchtigkeit,
- ein anderes unübersehbares Fehlverhalten, z.B. gehäufte Abstürze, oder
- nur ein wenig offenkundiges Absinken der Zuverlässigkeit.

Zur zeitnahen Beseitigung der Zuverlässigkeitsminderungen durch Ausfälle wird Hardware regelmäßigen **Wartungstests** unterzogen, z.B. in Form von Einschalttests (siehe später Abschn. 6.123).

1.3 Fehlermodell, Haftfehler

2.13 Fehlermodellierung

Die zu findenen Fehler sind zum Zeitpunkt der Testauswahl unbekannt. Testauswahl und Bewertung mit Hilfe von Fehlermodellen.

Fehlermodell: Algorithmus zur Berechnung einer Menge von Modellfehlern aus einer Beschreibung des Testobjekts.

Modellfehler: geringfügige Beschreibungsverfälschung (**Mutation**).

Fehlerorientierte Testbewertung:

```
Wiederhole für jeden Test:
    Wiederhole für jeden Modellfehler:
        Bestimme, ob nachweisbar.
```

Fehlerorientierte Testsuche:

```
Wiederhole für alle Modellfehler:
    Suche Eingaben für den Nachweis.
```

Für Hardware haben sich **Haftfehler** etabliert. Für Software statt Modellfehler Abdeckungskriterien (Folgeabschnitt).

2.14 Das Haftfehlermodell

Seit Jahrzehnten das verbreitetste Fehlermodell für digitale Schaltkreise. In der Vorlesung das Beispielfehlermodell.

Das Haftfehlermodell generiert für eine Schaltung aus Logikgattern für jeden Gatteranschluss zwei Modellfehler (Beschreibungsmutationen):

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

Die initiale Fehlermenge wird von identisch oder implizit nachweisbaren und redundanten (nicht nachweisbaren) Modellfehlern bereinigt.

Entscheidente Merkmale für die Brauchbarkeit von Fehlermodellen allg. und von Haftfehlern insbesondere:

- Aufwand der Fehlersimulation und Testberechung (Abschn. 6.2),
- Größenordnung der Wahrscheinlichkeit p_{ij}, dass Tests für Modellfehler ähliche wirkliche Fehler finden (Abschn. 3.2.3).

In deisem Abschnitt beispielorientierte qualitative Einführung.

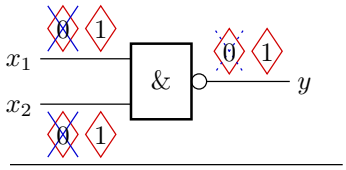
---

p_{ij} Wahrscheinlichkeit, dass ein Test, der Modellfehler j nachweist, auch Fehler i findet.

2.15 Haftfehler für ein Logikgatter

Für jeden Gatteranschluss wird unterstellt:

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



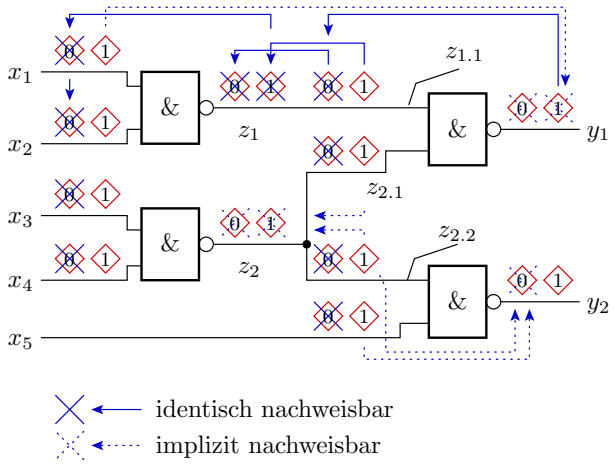
x_2	x_1	$\overline{x_2 \wedge x_1}$	sa0(x_1)	sa1(x_1)	sa0(x_2)	sa1(x_2)	sa0(y)	sa1(y)
0	0	1	1	1	1	1	0	1
0	1	1	1	1	1	0	0	1
1	0	1	1	0	1	1	0	1
1	1	0	1	0	1	0	0	1

Nachweisidentität (gleiche Nachweismenge)

0 sa0-Modellfehler
1 sa1-Modellfehler
× identisch nachweisbar
⋰ implizit nachweisbar
⋯→ Nachweisimplikation
zugehörige Eingabe ist Element der Nachweismenge

Dabei entstehen automatisch identisch und implizit nachweisbare Fehlerannahmen an jedem Gatter, die für die Testsuche und Bewertung nicht oder nur bedingt hilfreich sind und gestrichen werden können.

2.16 Schaltungsverbund



Größe der Anfangsfehlermenge:	24
Anzahl der nicht identisch nachweisbaren Fehler:	14
ohne implizit nachgewiesene Fehler:	9

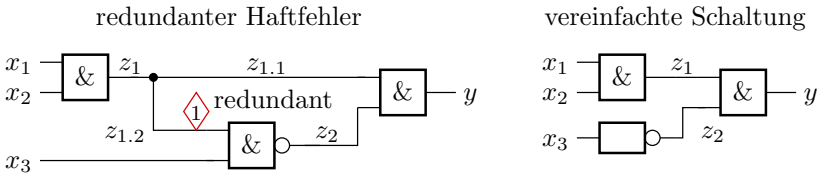
	Mengen von identisch nachweisbaren Fehlern	Nachweis impliziert durch
1	sa0(x_1), sa0(x_2), sa1(z_1), sa1($z_{1.1}$)	
2	sa1(x_1)	
3	sa1(x_2)	
4	sa0(x_3), sa0(x_4), sa1(z_2)	9, 12
5	sa1(x_3)	
6	sa1(x_4)	
7	sa0(z_2)	5, 6, 8, 11
8	sa0(z_1), sa0($z_{1.1}$), sa0($z_{2.1}$), sa1(y_1)	2, 3
9	sa1($z_{2.1}$)	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

Identischer Nachweis bei verzweigungsfreien Leitungen und Nachweisimplikation an Verzweigungen.

2.18 Redundante Fehler

Definition: Fehlerredundanz

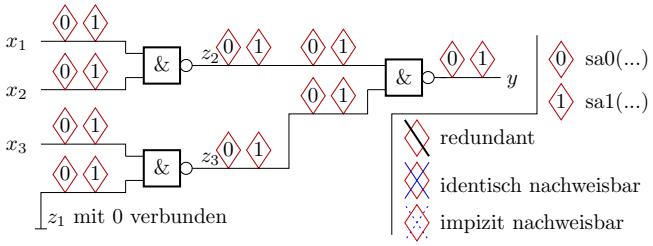
Redundanter Fehler sind Verfälschungen der Systembeschreibung, die die Funktion nicht verändern und damit auch nicht durch dynamische Tests nachweisbar ist.



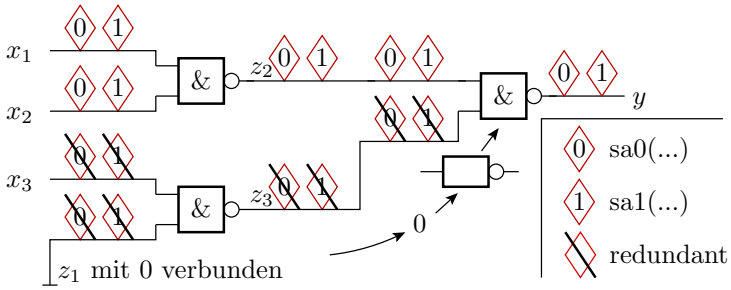
- Die Fehlerinfektion verlangt $z_1 = 0$ und die Ausbreitung von z_2 bis y verlangt $z_2 = 1$. Keine Eingabe $x_3x_2x_1$ kann den Fehler nachweisen.
- Die Beseitigung redundanter Fehler dient auch zur Vereinfachung der Systembeschreibung.

Beispiel 2.1 Haftfehlermenge

Schaltung mit 14 eingezeichneten Haftfehlern:



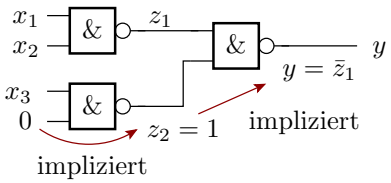
a) *Welche der Haftfehler sind redundant (nicht anregbar und/oder nicht beobachtbar).*



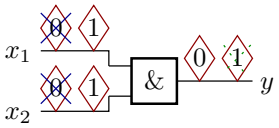
b) *Zeichen der vereinfachten Schaltung ohne redundante Haftfehler mit der Initalfehlermenge. Streichen der identisch nachweisbaren Fehler bis auf einen und Kennzeichnen des implizit nachweisbaren Haftfehlers.*

Die Funktion hängt nicht von x_3 ab und ist: $y = x_1 \wedge x_2$

Vereinfachungsmöglichkeiten

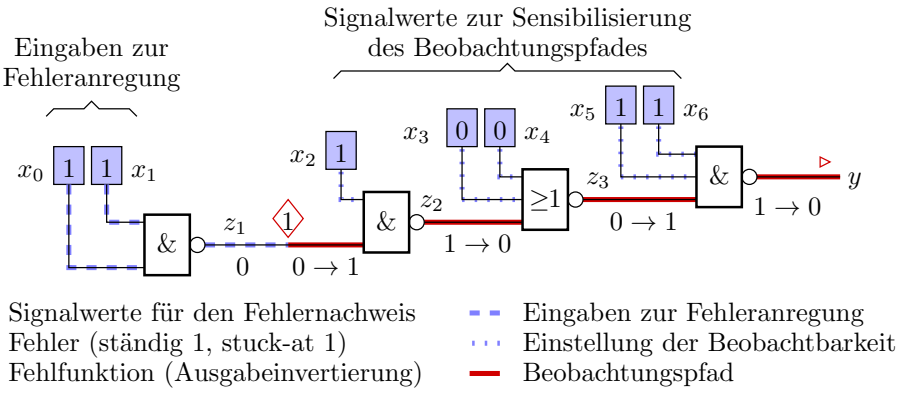


Redizierung der Fehlermenge für die vereinfachte Schaltung



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).

2.20 Testsuche



Suche durch Pfadsensibilisierung (Abschn. 6.2.2 *D-Algorithmus*):

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

2.21 Nachweisbedingungen und -mengen

Der Fehlnachweis lässt sich in Bedingungen aufspalten, im Beispiel in

- Infektion (Verursachung einer lokalen Verfälschung):

$$c_\mathrm{I} = x_1 x_0$$

- Ausbreitung (Verfälschungsausbreitung bis Ausgabe):

$$c_\mathrm{P} = x_6 x_5 \bar{x}_4 \bar{x}_3 x_2$$

Jede Bedingung repräsentiert eine Menge von Eingabewerten. Nachweismenge ist die Schnittmenge der Infektions- und Ausbreitungsmenge. Die Fehlernachweisbedingung ist die UND-Verknüpfung:

$$c_\mathrm{D} = c_\mathrm{I} \wedge c_\mathrm{P} = x_6 x_5 \bar{x}_4 \bar{x}_3 x_2 x_1 x_0$$

---

c_I, c_P, c_D Infektions-, Ausbreitungs- und Nachweisbedingung.

Infektion: Erfüllte Zusatzbedingung für eine lokale Werteverfälschung bei Ausführung.

2.22 Fehlernachweiswahrscheinlichkeit

Mit zufälligen Eingaben sind Infektion, Ausbreitung und Nachweis zufällige Ereignisse. Wenn alle 2^7 möglichen Eingaben gleichhäufig auftreten, betragen die Eintrittswahrscheinlichkeiten im Beispiel:

Infektionswahrscheinlichkeit;	$p_\mathrm{I} = 2^{-2}$
Ausbreitungswahrscheinlichkeit:	$p_\mathrm{P} = 2^{-5}$
Nachweiswahrscheinlichkeit:	$p_\mathrm{D} = p_\mathrm{I} \cdot p_\mathrm{P} = 2^{-7}$

Die tatsächlichen Fehler sind unbekannt, teilen sich aber in der Regel mit einigen der Modellfehler Nachweisbedingungen. Tests, die Modellfehler j nachweisen, erkennen wirkliche Fehler i mit ähnlichen Infektions- und Ausbreitungsbedingungen mit erhöhter Wahrscheinlichkeit $p_{ij} \gg p_i$.

---

p_I, p_P, p_D Infektions-, Ausbreitungs- und Nachweiswahrscheinlichkeit.

p_i Nachweiswahrscheinlichkeit für einen Fehler i mit einem zufällig gewählten Test.

1.4 Kriterienabdeckung

2.23 Fehlermodellierung für Software

Fehlermodellierung für Software erfolgt durch Mutation:

- Verfälschung eines arithmetischen Ausdrucks (x=a+b ⇒ x=a*b)
- Verfälschung eines booleschen Ausdrucks (if(a>b){} ⇒ if(a<b){})
- Verfälschung einer Adresszuweisung (ref=obj1 ⇒ ref=obj2)
- Entfernen eines Schlüsselworts (static int x=5 ⇒ int x=5), ...
- Off-by-One-Fehler, z.B. Wertezuweisung (y=a+x ⇒ y=a+x+1)

Vielfältiger als bei digitalen Schaltungen. Erfordert Beschränkung:

1. zufällige Stichprobe typischer Fehler,
2. etwas ähnliches wie Haftfehler (Off-by-One-Fehlermodell),
3. **Abdeckungskriterien** statt Modellfehler.

(1) genauste Vorhersage der tatsächlichen Fehlerabdeckung, (2) Übernahme der Haftfehler-Idee für Software, (3) bestes Aufwand/Nutzenverhältnis für Software-Testauswahl, warum Bedarf einer Erklärung.

Weiterführende Literatur: [4, 3]

2.24 Fehlerinjektion

Injektion einer Stichprobe typischer Fehler in das System

1. zur Untersuchung, welche Fehler ein Test nachweist,
2. theoretisch auch Testsuche für den Nachweis der Fehler

aber auch

- Kontrolle der Fehlfunktionsbehandlung,
- Kontrolle der Verfügbarkeit, Zuverlässigkeit und Sicherheit, ...

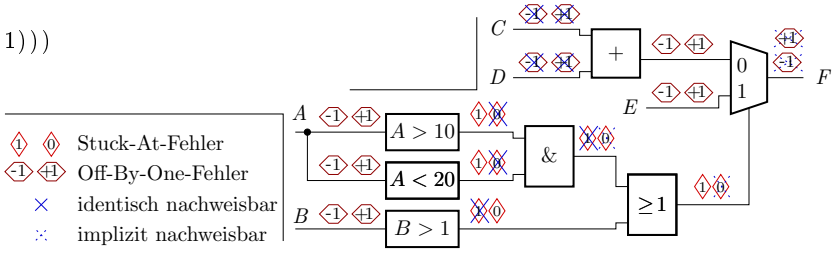
(1) **Statistische gesicherte** Modelle, insbesondere auch zur Festlegung der Stichprobengrößen in Abhängigkeit der gewünschen Vorhersagegenauigkeit (Abschn. 4.2.7), (2) statistisch problematisch.

Anwendbar auf Hardware, Software, Mechanik, ... zur Kontrolle von Verhersagemodellen der Fehlerabdeckungen, Fehlerfunktionsraten, ... auch zur **Untersuchung** der **Beziehung** zwischen **Kriterienabdeckung** und **Fehlerabdeckung**, die uns noch interessieren wird.

Bei **Software** problematisch, dass für **fehlende Aspekte** häufig **keine ähnlich nachweisbare Fehler** injizierbar (Folie 2.28).

2.25 Off-By-One-Fehler

```
if (((A>10)&&(A<20))||(B>1)))
  F=C+D;
else F=E;
```



Ein mit Haftfehlern vergleichbares Modell für komplett durchmusterbare Modellfehler- (Mutations-) Mengen. Generiert für alle Ein- und Ausgaben aller Operation oder Anweisung:

- ⟨+1⟩ Wert geringfügig zu groß und
- ⟨−1⟩ Wert geringfügig zu klein.

Bereinigung der initialen Fehlermenge von identisch oder implizit nachweisbaren und redundanten (nicht nachweisbaren) Modellfehlern.

Auch Fehlende-Aspekte-Problem, Folie 2.28.

2.26 Abdeckungskriterien statt Modellfehler

Aufspaltung Software-Fehlernachweis in eine Konjunktion [1]

$$\text{Nachweis} = \text{Erreichbarkeit} \wedge \text{Infektion} \wedge \text{Ausbreitung} \tag{2.3}$$

Erreichbarkeit ist mit **Abdeckungszähler**n im Code kontrollierbar (Folie 7.110). Infektion, z.B. Division durch null, verlangt eine Bedingungsabfrage vor dem Abdeckungszähler, im Beispiel »Divisor = 0?« vor der der Divisionsausführung (siehe auch Folie 7.113).

Während die Kontrolle der Erreichbarkeit und Infektion nur eine instrumentierte Programmversion mit allen Zählern benötigen, verlangt **Ausbreitung**skontroll genau wie Fehlerinjektion je **eine instrumentierte Programmversion je Fehler**.

Instrumentierung: Ergänzung von Programmen um Code, um das Verhalten zu untersuchen.
Erreichbarkeit: Ausführung des fehlerhaften Programmteils.
Infektion: Erfüllte Zusatzbedingung für eine lokale Werteverfälschung bei Ausführung.
Ausbreitung: Abbildung der lokalen Verfälschung auf ein beobachtbares Ergebnis.
Weiterführende Literatur: [2]

(2.3) Nachweis = Erreichbarkeit ∧ Infektion ∧ Ausbreitung

$$p_D = p_A \cdot p_I \cdot p_P$$

Die Fehlernachweiswahrscheinlichkeit von Tests, die sich mit den zu findenen Fehlern nur Erreichbarkeit (und Infektion) teilen, ist viel geringer als von Tests, die sich zusätzlich Infektions- und Ausbreitungskriterien teilen. Dieselbe Fehlerabdeckung verlangt eine Abdeckungsanzahl je Kriterium proportional

$$w \sim \frac{1}{\min([p_I \cdot] p_P)}$$

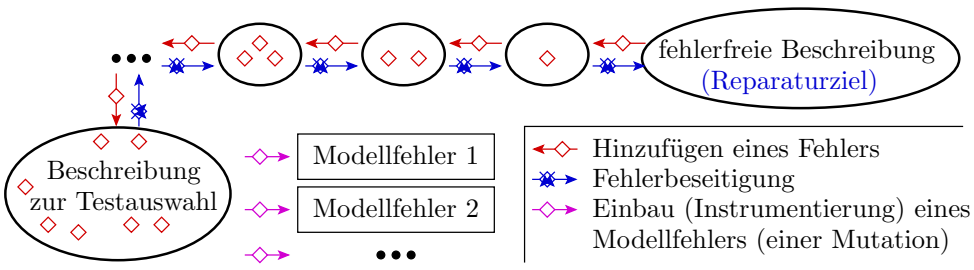
(siehe später Folie 3.65).

Nur Kontrolle Erreichbarkeit (und Infektion) Zehnerpotenzen weniger Rechenaufwand als mit Ausbreitungskontrolle (oder Fehlerinjektion).

Dafür je Erreichbarkeitskriterium $w \gg 1$ zufällige Tests erforderlich.

p_A, p_I, p_P Erreichbarkeits-, Infektions- und Ausbreitungswahrscheinlichkeit.
w Erforderliche Abdeckungsanzahl je Kriterium.

2.28 Fehlende Aspekte



- Das zu testende Programm enthält die zu findenen Fehler d.h. injizierte Modellfehler sind Mutationen fehlerhafter Programme.
- Für fehlende Programmzweige (Fallunterscheidungen, Ausnahmebehandlungen, ...) nur Orte der potentiell fehlender Verzweigungen, d.h. Teilerreichbarkeitsbedingungen angebbar.

Fehlende Aspekte verlangen für jeden erreichbaren Programmpunkt hohe Abdeckungsanzahl. Dadurch auch für die anderen Fehler die Berücksichtigung der Infektions- und Ausbreitungskriterien unwichtig.

2.29 Abdeckungsklassen

1. Grobtest: Einige wohlplatzierte Tests für den Nachweis der überwiegenden Mehrheit der Fehler (Pareto-Prinzip).
2. Gründlicher Test: hohe anteilige Kriterienabdeckung,
3. W1: Abdeckung aller (Erreichbarkeits-) Kriterien mit $w \geq 1$.
4. Vielffachabdeckung $w \gg 1$, Vision für die Zukunft.

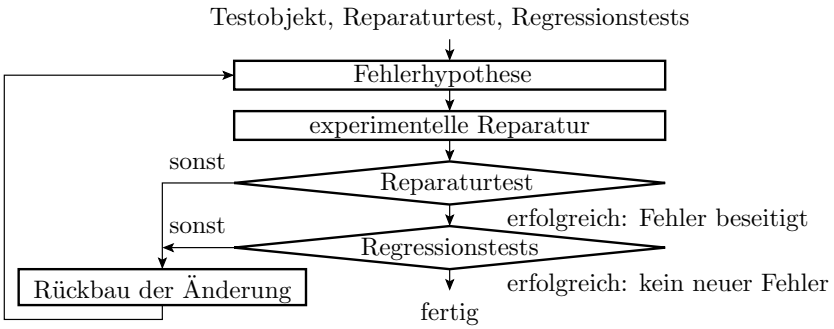
Aufwand nimmt von (1) bis (4) stark zu und der Anteil der nicht nachweisbaren Fehler stark ab. (1) zweckmäßig innerhalb der Codierungsphase, weil viele Nachbesserungen. (2) gilt heute als gute Praxis für viele Anwendungen. (3) heute beste Praxis sogar in sicherheitskritischen Bereichen wie Flugwesen. Befreit Hersteller von der Produkthaftung (Folie 7.117). (4) mit w-fache Aufwand von (3) zufriedenstellende Zuverlässigkeit und Sicherheit ab der ersten Einsatzversion. Mit aktueller Testautomatisierung unwirtschaftlich (Abschn. 7.5).

Pareto-Prinzip: Ein kleiner Teil der Ursachen ist für den größten Teil der Wirkungen verantwortlich. Wenige Tests decken die überwiegende Mehrheit der Fehler auf. Ein kleiner Teil der Fehler verursacht die überwiegende Mehrheit der Fehlfunktionen. ....

2 Fehlerbeseitigung

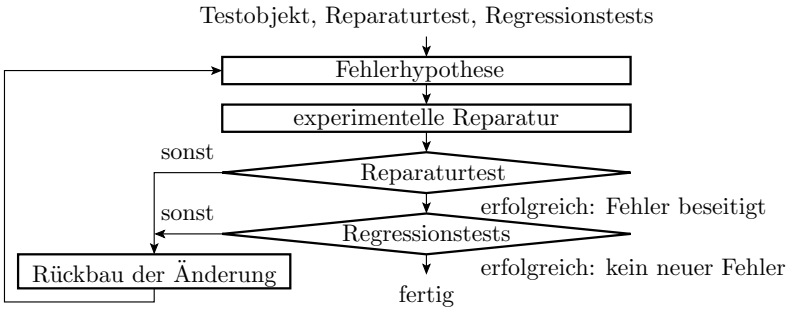
2.1 Beseitigungsiteration

2.31 Experimentelle Reparatur



Iteration aus Beseitigungsversuchen für hypothetische Fehler und Erfolgskontrolle durch Testwiederholung.

Rückbau nach erfolglosen Reparaturversuchen zur Vermeidung neuer Fehler durch Reparatur.

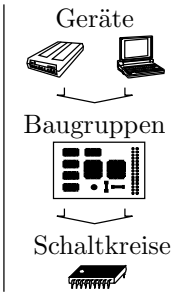


Reparaturtests sind fehlgeschlagene Tests für den Fehlernachweis. Idealerweise **deterministisch** zum Ausschluss von Fehlklassifikationen (Erfolg als Versagen und umgekehrt). Warum?

Regressionstests sind erfolgreichen Tests, die vor der Reperatur keine vorhandenen Fehler nachweisen.

Geringe **Rate neu entstehender Fehler je beseitigter Fehler** verlangt hohe Erfolgswahrscheinlichkeit Rückbau und hohe Fehlerabdeckung Regressionstest (siehe später Abschn. 3.3).

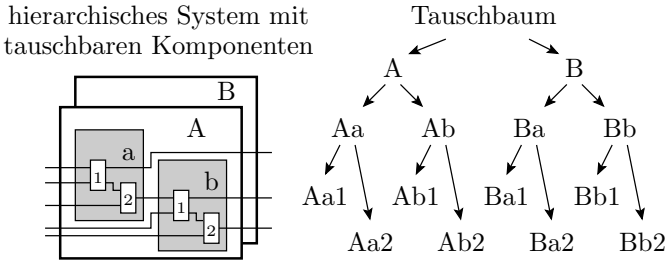
2.33 Tauschbare Komponenten



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:



2.34 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. Verständnis der Funktion des zu reparierenden Systems nicht zwingend

Fragen:

- Günstig ist der Tausch der Hälfte, von der fehlerhaften Hälfte auch die Hälfte, ... Warum?
- Kann man so auch Fehler in SW suchen, wenn ja, was für Fehler?

2.2 Fehlerdiagnose & -isolation

2.35 Fehlerdiagnose

Abschätzung von Ort-, Ursache und Beseitigungsmöglichkeiten von Fehlern aus Testergebnissen zur Minderung:

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

Allgemeine Diagnosetechniken:

- erfolgsorientiertes Tauschen und
- Rückverfolgung von Verfälschungen entgegen dem Daten- oder Berechnungfluss.

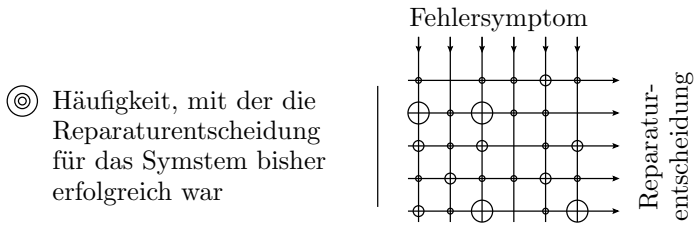
Voraussetzung ist ein reparaturgerechter Entwurf.

Reparaturgerechter Entwurf: Entwurfsvorkehrungen zur Ermöglichung ein wirtschaftlichen Fehlerlokalisierung und Reparatur.

2.36 Erfolgsorientiertes Tauschen

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

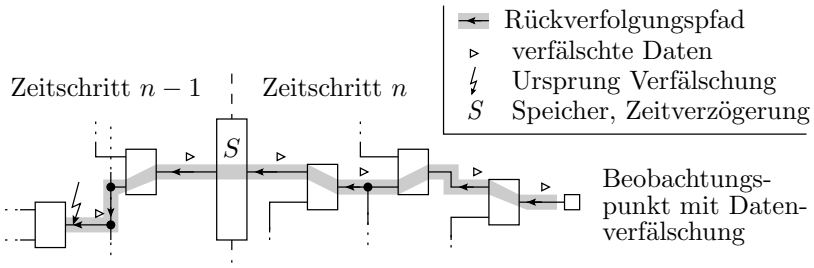
- Zählen der Erfolge unterschiedlicher Reparaturalternativen.
- Bei Reparatur, Beginn mit der erfolgsversprechendsten Möglichkeit.



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.

2.37 Rückverfolgung von Datenverfälschungen

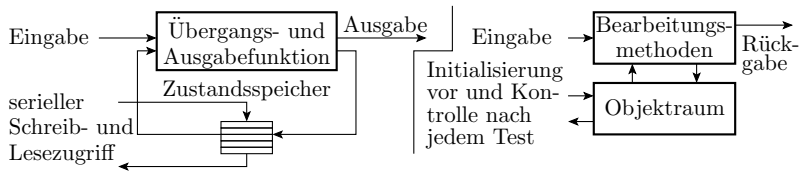


Von erkannter Verfälschung (Ausgabe oder anderer kontrollierter Wert) Rückverfolgung entgegen Signalfluss bis zur Komponente mit verfälschter Ausgabe und richtiger Eingabe, gegebenenfalls über Zeitschritte.

Bei Software Datenfluss statt Signalfluss (Abschn. 7.4.2).

Potentielle Verfälschungsquelle außer gefundenen Komponente bei Hardware z.B. auch Kurzschluss oder bei Software ein fehlgeleiteter Schreibzugriff. Letzter Lokalisierungsschritt Reparaturversuch.

2.38 Hirarchischer Test, isolierter Test, ...



Hierarchischer Test, Fehlersuche mit den Tests der kleinsten Bausteine, die ihn nachweisen (Folie 2.11 *Hierarchie und Test*).

Isolierter Test der Übergangsfunktion von Automaten und durch Neuinitialisierung des Zustands vor und Auslesen nach jedem Testschritt.

Isolierter Test der Bearbeitungsmethoden von Software durch Neuinitialisierung des Objektraums vor und Auslesen nach jedem Testschritt.

Bei Software einfügen von Testausgaben von Zwischenwerten (printf-Debugging), Trace-Aufzeichnung, ...

2.39 Reparatur- und prüfgerechter Entwurf

Sammlungen von

- Regeln »of good practise«, zur Ermöglichung / Vereinfachung von Test, Fehlerlokalisierung und Reparatur und
- Antipattern (typ. Aspekte, die Probleme verursachen).

Einige Regeln »of good practise«:

- Modulares System aus tauschbaren / seperat testbaren Funktionsblöcken.
- Deterministisches Verhalten mit gerichtetem Berechnungsfluss.
- Fehlfunktionsisolation (siehe Folgefolie).

Antipattern für Software:

- Weiterentwicklung alter Code mit schlechten Regressionstests. Mangelnder Nachweis der bei der Reparatur entstehenden Fehler.
- Unübersichtlichkeit erhöht die Anzahl der Reparaturversuche und darüber die Raten der neuen Fehler je beseitigter Fehler.
- Fehlende Dokumentationen erzwingen Blindfehlersuche, siehe übernächste Folie.

2.40 Fehlerisolation

Ausschluss der Beeinträchtigung des Restsystems durch nicht korrekt arbeitende Teilsysteme:

- Zwischenergebnisskontrolle an Teilsystemgrenzen. Weitergabe nur von als korrekt eingestuften Werten.
- Verhinderung unzulässiger Ressourcen-Blockierung (Rechenzeit, Speicher, IO-Geräte, ...) fehlerhafte Teilsysteme.
- Keine Zugriffmöglichkeit auf Daten und Resourcen anderer Funktionseinheiten außer über definierte Schnittstellen.
- Bei extremen Verlässlichkeitsanforderungen sogar physikalische und räumliche Trennung zur Risikominderung durch ursächlich gleiche Probleme (Stromausfall, Hacker-Angriff, ...).

Grundlegendes Architekturmuster für

- Betriebssysteme,
- eingebettete Systeme,
- verteilte Systeme,
- sicherheitskritische Systeme, ...

2.41 Blindfehlersuche

Die Alternative zum systematischen Tauschen mit oder ohne Fehlerdiagnose ist ein »Blindfehlersuche«, d.h. ein intuitives Probieren.

Aufwändig, oft nicht erfolgreich, frustrierend aber:

- bei fehlenden Tauschmöglichkeiten,
- keiner Möglichkeit zur Rückverfolgung,
- fehlender Quallifikation oder fehlenden Dokumentationen

die einzige Möglichkeit der Fehlerbeseitigung.

Reparaturgerechte Systemgestaltung, Testgestaltung, Diagnose, ... sind große eigenständige Forschungs- und Arbeitsgebiete. Weiterführende Literatur [Buch scintific debugging].

2.3 Ausbeute, Defektanteil

2.42 Ausbeute und Defektanteil

Bei nicht reparierbaren Systemen und Komponenten interessiert nicht die Fehleranzahl, sondern der Anteil der verwendbaren bzw. der defekten Produkte.

Die Ausbeute ist der Anteil der als gut befundenen Produkte:

$$Y = 1 - \left.\frac{\#DD}{\#P}\right|_{\text{ACR}} \tag{2.4}$$

Der Defektanteil ist der Anteil der tatsächlich defekten Produkte:

$$DL = \left.\frac{\#D}{\#P}\right|_{\text{ACR}} \tag{2.5}$$

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

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

Y, DL	Ausbeute, Defektanteil.
$\#D$, $\#DD$	Anzahl aller defekten Produkte, Anzahl der davon erkannten defekten Produkte.
$\#P$	Anzahl aller getesteten Produkte.
ACR	Brauchbare Schätzwerte nur bei geeigneten Zählwertgrößen.

2.43 Defektabdeckung

Die Defektabdeckung ist der Anteil der erkannten defekten Produkte:

$$DC = \left.\frac{\#DD}{\#D}\right|_{\text{ACR}} \tag{2.6}$$

Ausbeute und Defektanteil ungetesteter Produkte:

$$Y = 1 - DL_{\text{M}} \cdot DC \tag{2.7}$$

Ohne Test ($DC = 0$) ist die Ausbeute immer $Y = 1$.

Aussortieren erkannter defekter Produkte verringert Zähler und Nenner in (Gl. 2.7) um die Anzahl der erkannten defekten Produkte:

$$DL = \frac{\#P \cdot DL_{\text{M}} - \#P \cdot DL_{\text{M}} \cdot DC}{\#P - \#P \cdot DL_{\text{M}} \cdot DC}$$

$$DL = \frac{DL_{\text{M}} \cdot (1 - DC)}{1 - DL_{\text{M}} \cdot DC} \tag{2.8}$$

DC	Defektabdeckung (defect coverage), Anteil der erkennbar defekten Produkte.
$\#D$, $\#DD$	Anzahl aller defekten Produkte, Anzahl der davon erkannten defekten Produkte.
Y, DL	Ausbeute, Defektanteil.
DL_{M}	Defektanteil der Fertigung vor Aussortieren der erkannten defekten Produkte.

2.44 Erforderliche Defektabdeckung

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

eingesetzt in

(2.7) $$Y = 1 - DL_{\mathrm{M}} \cdot DC$$

ergibt einen Defektanteil nach Aussortieren in Abhängigkeit von Defektabdeckung und Ausbeute:

$$DL = \frac{(1-Y)\cdot(1-DC)}{Y \cdot DC} \tag{2.9}$$

Erforderliche Defektabdeckung zur Erzielung eines Defektanteil DL bei einer Ausbeute Y:

$$DC = \frac{1-Y}{1+(DL-1)\cdot Y} \tag{2.10}$$

Y, DL Ausbeute, Defektanteil.

DC Defektabdeckung (defect coverage), Anteil der erkennbar defekten Produkte.

2.45 Defektanteil digitaler Schaltkreise

Für Schaltkreise findet man in der Literatur als typische Angaben:

- Ausbaute: $Y = 10\% \ldots 90\%$
- Defektanteil: $DL = 200\,\mathrm{dpm} \ldots 1000\,\mathrm{dpm}$
- Haftfehlerabdeckung: $FC_{\mathrm{SA}} = 80\% \ldots 99\%$..

Erforderliche Defektabdeckung nach

(2.10) $$DC = \frac{1-Y}{1+(DL-1)\cdot Y}$$

	$Y = 10\%$	$Y = 50\%$	$Y = 90\%$
$DL = 200\,\mathrm{dpm}$	$1 - 2{,}2 \cdot 10^{-5}$	$1 - 2 \cdot 10^{-4}$	$1 - 1{,}7 \cdot 10^{-3}$
$DL = 1000\,\mathrm{dpm}$	$1 - 1{,}1 \cdot 10^{-4}$	$1 - 2 \cdot 10^{-3}$	$1 - 8{,}9 \cdot 10^{-3}$

Der Anteil der Schaltkreise, die der Test nicht erkennt, ist laut Abschätzung eine bis zwei Zehnerpotenzen kleiner als der Anteil der nicht nachweisbaren Haftfehler. Daraus resultierende Fragen:

- Gilt für Schaltkreise tatsächlich $1 - DC \ll 1 - FC_{\mathrm{sa}}$ oder
- ist die Dunkelziffen der defekten Schaltkreise so viel größer?

Weitere Frage, wie oft enthalten Rechnern defekte Schaltkreise?

2.46 Systeme aus getesteten Teilsystemen

Für Systeme aus Teilsystemen gelten die Grundregeln:

- gründlicher Test der Teilsysteme vor dem Einbau,
- Testfokusierung nach Einbau auf die Verbindungen.

Vor Einbau in Teilsysteme nicht erkannte Fehler bleiben auch im Gesamtsystem unerkannt. Zu erwartende Fehleranzahl Gesamtsystem:

$$\mu_{\mathrm{F}} = \mu_{\mathrm{F.Con}} \cdot (1 - FC_{\mathrm{Con}}) + \sum_{i=1}^{\#Prt} \mu_{\mathrm{F}.i} \tag{2.11}$$

Für den zu erwartenden Fehleranteil folgt später die Abschätzung:

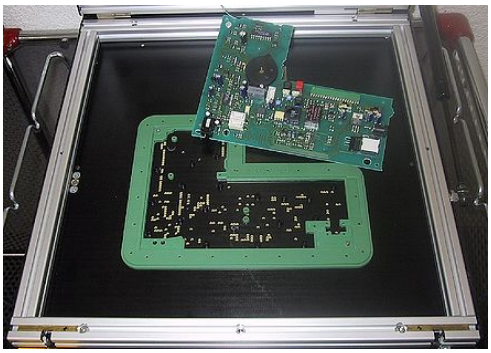
(4.89) $$\mu_{\mathrm{DL}} = 1 - \mathrm{e}^{-\mu_{\mathrm{F}}}$$

Für sehr wenige zu erwartende Fehler $\mu_{\mathrm{F}} \ll 1$ gilt:

(4.90) $$\mu_{\mathrm{DL}} = 1 - \left(1 + (-\mu_{\mathrm{F}}) + \frac{(-\mu_{\mathrm{F}})^2}{2!} + \ldots\right) \overset{(\mu_{\mathrm{F}} \ll 1)^*}{=} \mu_{\mathrm{F}}$$

μ_{F}, $\mu_{\mathrm{F.Con}}$ Zu erwartende Gesamtfehleranzahl, zu erwartende Anzahl der Verbindungsfehler.
$\mu_{\mathrm{F}.i}$ Zu erwartendender Fehleranzahl Teilsystem i.
FC_{Con} Fehlerübedeckung für Verbindungsfehler (Fault coverange for connection faults).
$\#Prt$ Anzahl der Bauteile.
μ_{DL} Zu erwartender Defektanteil.

2.47 Leiterplattentest



Bestückte Leiterplatten 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 und Fehleranteil der Bauteile sehr klein ($\mu_{\mathrm{F}.i} \ll 1 \Rightarrow \mu_{\mathrm{F}.i} = \mu_{\mathrm{DL}.i}$):

$$\mu_{\mathrm{F}} = \sum_{i=1}^{\#Prt} \mu_{\mathrm{DL}.i} \tag{2.12}$$

Für $\mu_{\mathrm{F}} \ll 1$ ist die abgeschätzte erwartete Fehleranzahl der zu erwartende Fehleranteil der Baugruppe.

μ_{F} Zu erwartende Fehleranzahl des Gesamtsystems.
$\#Prt$ Anzahl der Bauteile.
$\mu_{\mathrm{DL}.i}$ Zu erwartendender Defektanteil von Bauteil i.

2.48 Beispielabschätzung

Leiterplatte mit nachfolgenden Komponenten:

Typ	Anzahl	$\mu_{\mathrm{DL}.i}$
Leiterplatte	1	20 dpm
Schaltkreise	20	200 dpm
diskrete Bauteile	35	10 dpm
Lötstellen	560	1 dpm

$$\begin{aligned}\mu_{\mathrm{DL.Sys}} = \mu_{\mathrm{F}} &= 10\,\mathrm{dpm} + 20 \cdot 200\,\mathrm{dpm} + 35 \cdot 10\,\mathrm{dpm} + 560 \cdot 1\,\mathrm{dpm} \\ &= 5000\,\mathrm{dpm} = 0{,}005\,\mathrm{dpu}\end{aligned}$$

Etwa jedes 200ste Gerät enthält ein nicht erkanntes defektes Bauteil, natürlich nur mit einem kaum nachweisbaren Defekt, der die Zuverlässigkeit nur wenig mindert.

Wenn Defektanteils der Schaltkreisen tatsächlich um Zehnerpotenzen größer

$\mu_{\mathrm{DL.Sys}}$ Defektanteil des Systems.

Zusammenfassung

2.49 Fehlerbeseitigung

Fehlerbeseitigung: Iteration aus Beseitigungsversuchen für hypothetische Fehler und Erfolgskontrolle durch Testwiederholung:

- Beseitigung aller erkennbaren Fehler.
- Rückbau nach erfolglosen Reparaturversuchen.
- Bei wenigen Bausteinen durch systematisches Tauschen.

Fehlerdiagnose: Abschätzung von Ort-, Ursache und Beseitigungsmöglichkeiten von Fehlern aus Testergebnissen:

- Pareto-Prinzip, Bevorzugungung erfolgsversprechender Fehlerbeseitigungsversuche.
- Rückverfolgung entgegen den Berechnungs- bzw. Signalfluss.

Reparaturgerechter Entwurf:

- Tauschbare Module, deterministische Verhalten,
- gerichteter Berechnungfluss, Fehlerisolation, ...

Bei einer vernünftigen Reparaturtechnologie werden alle erkannten Fehler beseitigt und es entsteht nur eine vernachlässigbar kleine Anzahl neuer nicht nachweisbarer Fehler.

2.50 Test und Testvielfalt

IT-Systeme werden einer Vielzahl Tests unterzogen:

- dem Entwurfsfluss folgend nach jeder Entwurfsphase,
- dem Fertigungsfluss folgend bausteinweise und danach das Zusammenwirken der Bausteine im übergeordneten System,
- zur Fehlerbeseitigung vor dem Einsatz und als Wartungstest,
- jeweils statisch (direkte Merkmalskontrolle) und dymamisch (Ausprobieren mit Beispieleingaben).

Kenngrößen:

- Fehlerabdeckung

  (2.1) $$FC = \left.\frac{\#DF}{\#F}\right|_{\text{ACR}}$$

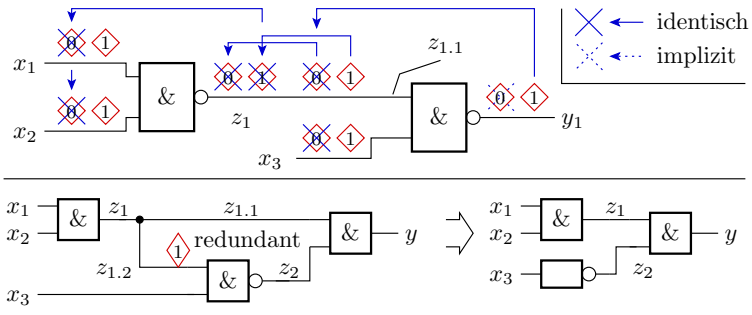
- Phantomfehlerrate

  (2.2) $$\zeta_{\text{PF}} = \left.\frac{\#PM}{N}\right|_{\text{ACR}}$$

Die Testauswahl und Bewertung erfolgt mit Hilfe von Modellfehlern:

- zielgerichtet (Testsuche für jeden Modellfehler) oder
- zufällig (nur modellfehlerorientierte Bewertung).

2.51 Haftfehler



Seit Jahrzehnten wichtigstes Fehlermodell für digitale Schaltungen:

- Initialfehlermenge: je Gatteranschluss sa0 und sa1.
- Zusammenfassen identisch nachweisbarer Fehler, streichen redundanter und implizit nachweisbarer Fehler.

Im Weiteren oft Beispiel, wenn konkrete Modellfehler und Modellfehlermengen zur Veranschaulichung von Sachverhalten benötigt werden.

2.52 Abdeckungskriterien

Software enthält die zu findenen Fehler. Bedeutet für Testauswahl u.a.

1. die die Abdeckungskontrolle für einen Mutation kostet ähnlich viel Rechenaufwand wie die Durchführung eines Tests.
2. Für vergessene Aspekte in Form fehlender Programmzweige lassen sich nur Erreichbarkeitskriterien für die existierenden Pfade, aber nicht die nicht abzweigenden fehlenden Pfade modellieren.

Wegen (1) ist es zielführender, bei gleichem Rechenaufwand mehr Tests, die nur mit Erreichbarkeit- und eventuell auch Infektionskriterien instrumentiert sind, abzuarbeiten, als je Test viel mutierte Programmversionen. Erfordert hohe Abdeckungsanzahl je Kriterium.

(2) bedeutet, dass es für vergessene Aspekte, die einen erheblich Anteil der möglichen Fehler darstellen, nur Pfade als Teilereichbarkeitbedingungen modellierbar sind. Das verlangt ohnehin für alle Pfade eine hohe Abdeckungsanzahl.

2.53 Defektanteil, Ausbeute

Bei nicht reparierbaren Systemen und tauschbaren Komponenten interessieren statt der Fehleranzahl, Ausbeute und Fehleranteil:

(2.4) $$Y = 1 - \left.\frac{\#DD}{\#P}\right|_{\mathrm{ACR}}$$

(2.5) $$DL = \left.\frac{\#D}{\#P}\right|_{\mathrm{ACR}}$$

Bindeglied ist die Defektabdeckung:

(2.6) $$DC = \left.\frac{\#DD}{\#D}\right|_{\mathrm{ACR}}$$

(2.7) $$Y = 1 - DL_{\mathrm{M}} \cdot DC$$

Defektanteil nach Ersatz der erkannten defekten Produkte:

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

(2.9) $$DL = \frac{(1-Y) \cdot (1-DC)}{Y \cdot DC}$$

2.54 Modulare Systeme aus getesteten Bauteilen

Fehleranzahl:

(2.11) $$\mu_{\mathrm{F}} = \mu_{\mathrm{F.Con}} \cdot (1 - FC_{\mathrm{Con}}) + \sum_{i=1}^{\#Prt} \mu_{\mathrm{F}.i}$$

Beziehung Fehleranteil und Fehleranzahl (Vorgriff):

(4.89) $$\mu_{\mathrm{DL}} = 1 - \mathrm{e}^{-\mu_{\mathrm{F}}}$$

Für $\mu_{\mathrm{F}} \ll 1$:

(4.90) $$\mu_{\mathrm{DL}} = 1 - \left(1 + (-\mu_{\mathrm{F}}) + \frac{(-\mu_{\mathrm{F}})^2}{2!} + \ldots\right) \overset{(\mu_{\mathrm{F}} \ll 1)^*}{=} \mu_{\mathrm{F}}$$

Für getestete Leiterplatten gilt in der Regel $FC_{\mathrm{Con}} = 1$ und Fehleranzahl gleich Summe der Defektanteile aller Bauteile:

(2.12) $$\mu_{\mathrm{F}} = \sum_{i=1}^{\#Prt} \mu_{\mathrm{DL}.i}$$

Für $\mu_{\mathrm{F}} \ll 1$ gilt auch hier Gl. 4.90.

3 Zuverlässigkeit & Test

3.1 Einfache Abschätzung

Beispiel 2.2 Fehleranzahl und Zuverlässigkeit

Programmgröße 10.000 NLOC. 30 ... 100 Fehler je 1000 NLOC. Fehlerabdeckung der Tests $FC = 70\%$.

a) Wie groß ist die Fehleranzahl nach Beseitigung aller erkennbaren Fehler?

$$10.000\,\text{NLOC} \cdot \frac{30\ [\text{F}] \ldots 100\,[\text{F}]}{1000\,\text{NLOC}} \cdot (1 - 70\%) = 90\,[\text{F}] \ldots 300\ [\text{F}]$$

b) Wie zuverlässig ist ein System mit ca. 90 bis 300 Fehlern?

Hängt von der Fehlfunktionsrate der nicht erkannten Fehler ab.

Je höher die Fehlfunktionsrate je Fehler, je weniger Tests genügen.

Mit der Testanzahl nimmt nicht nur der Anteil der nicht nachweisbaren Fehler, sondern auch die zu erwartende Fehlfunktionsrate je Fehler ab.

Das interessiert uns genauer!

[F] Zählwert in Fehlern.
FC Fehlerabdeckung (fault coverage), Anteil der nachweisbaren Fehler.
NLOC Netto Lines of Code, Anzahl der Code-Zeilen ohne Kommentar und Leerzeilen.

2.56 Fehlfunktionsrate durch Fehler

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

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

ist eine Obergrenze $\zeta_\text{F} \leq \zeta_{\sum}$ und, wenn fast alle MF nur einen Fehler als Ursache haben, praktisch gleich der MF-Rate durch alle Fehler:

$$\zeta_\text{F} \overset{(\leq 1)}{=} \sum_{i=1}^{\#F} \zeta_i \quad \text{für} \quad \zeta_\text{F} \ll 1$$

Im weiteren betrachten wir meist gründlich vorgetestete Systeme mit $\zeta_\text{F} \ll 1$, in denen die schlimmsten Fehler schon beseitigt sind.

MF, HW Fehlfunktion, erbrachte Service-Leistung.
$\#F$ Anzahl der vorhandenen Fehler.
ζ_i MF-Rate verursacht durch Fehler i.
ζ_F Fehlfunktionsrate durch Fehler.
(≥ 1) Der errechnete Wert ist eine Obergrenze. Der tatsächliche Wert ist max. eins.

Unter den Annahmen:

- Beseitigung aller nachweisbaren Fehler,
- mittlere MF-Rate je nicht beseitigten Fehler $\bar{\zeta} < 1/N$,
- je Fehlfunktion nur ein Fehler als Ursache,
- Test mit Nutzungsprofil.

beträgt die MF-Rate für alle nicht beseitigten Fehler zusammen:

$$\zeta_{\mathrm{F}}(N) = \mu_{\mathrm{F}}(N) \cdot \bar{\zeta} \qquad (2.13)$$
$$\zeta_{\mathrm{F}}(N) < \frac{\mu_{\mathrm{F}}(N)}{N}$$

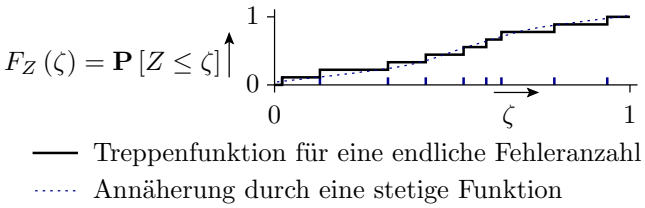
Die fehlerbezogene Teilzuverlässigkeit beträgt mindestens:

$$R_{\mathrm{F}} > \frac{N}{\mu_{\mathrm{F}}(N)}$$

$\zeta_{\mathrm{F}}(N)$ Fehlfunktionsrate durch Fehler in Abhängigkeit von der Testanzahl.
$\mu_{\mathrm{F}}(N)$ Zu erwartende Anzahl der Fehler, die nach N Tests nicht erkannt und beseitigt sind.
$\bar{\zeta}(N)$ Mittlere Fehlfunktionsrate je Fehler als Funktion der Testanzahl N.
N Anzahl der Tests, für die alle erkannten Fehler beseitigt sind.
$R_{\mathrm{F}}(N)$ Fehlerbezogene Teilzuverl. nach Beseitigung aller mit N Tests nachweisbaren Fehler.
Eingabeprofil: Nutzungshäufigkeit der Äquivalenzklassen ähnlich verarbeiteter Eingaben.
Nutzungsprofil: Eingabeprofil bei der Nutzung.

3.2 Verbessertes Modell

2.58 Verteilung der Fehlfunktionsrate



Die Verteilung $F_Z(\zeta)$ der Zufallsgröße Z beschreibt die Wahrscheinlichkeit, dass die Fehlfunktionsrate eines (zufällig ausgewählten) Fehlers nicht größer als ζ ist:

$$F_Z(\zeta) = P(Z \leq \zeta)$$

Dichte der Fehlfunktionsrate (siehe später Foliensatz 4):

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

Z Fehlfunktionsrate als Zufallsvariable.
$F_Z(\zeta)$ Verteilungsfunktion der Fehlfunktionsrate, Z – Zufallsvariable, ζ – Wert.
$h(\zeta)$ Dichtefunktion der Fehlfunktionsrate.

2.59 Fehlerabdeckung und Fehlfunktionsrate

Zu erwartende Anzahl der mit N Tests nicht beseitigten Fehler, wenn alle nachweisbaren Fehler beseitigt werden:

$$\mu_{\mathrm{F}}(N) = \mu_{\mathrm{F}} \cdot \int_0^1 p_{\mathrm{FNE}}(\zeta, N) \cdot h(\zeta) \cdot \mathrm{d}\zeta \qquad (2.14)$$

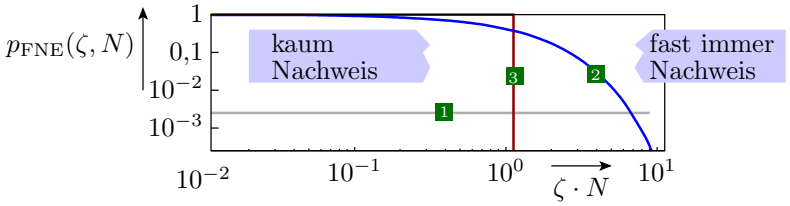
Zu erwartende Fehlfunktionsrate durch die nicht beseitigten Fehler:

$$\zeta_{\mathrm{F}}(N) \overset{(\leq 1)}{=} \mu_{\mathrm{F}} \cdot \underbrace{\int_0^1 p_{\mathrm{FNE}}(\zeta, N) \cdot h(\zeta) \cdot \zeta \cdot \mathrm{d}\zeta}_{\text{mittlere Fehlfunktionsrate je Fehler}} \qquad (2.15)$$

(Integration über die Produkte aus Häufigkeit des Vorhandenseins und Fehlfunktionsrate).

$\mu_F(N)$	Zu erwartende Anzahl der Fehler, die nach N Tests nicht erkannt und beseitigt sind.
μ_F	Zu erwartende Fehleranzahl vor der Iteration aus Test und Fehlerbeseitigung.
$p_{FNE}(\zeta, N)$	Wahrscheinlichkeit, dass Fehler mit MF-Rate ζ nach N Tests nicht beseitigt sind.
$h(\zeta)$	Dichtefunktion der Fehlfunktionsrate vor der Fehlerbeseitigung.
N	Anzahl der Tests, für die alle erkannten Fehler beseitigt sind.
$\zeta_F(N)$	Fehlfunktionsrate durch Fehler in Abhängigkeit von der Testanzahl.
(≥ 1)	Der errechnete Wert ist eine Obergrenze. Der tatsächliche Wert ist max. eins.

2.60 Fehlernachweiswahrscheinlichkeit



Die Nichtbeseitigungswahrscheinlichkeit $p_{FNE}(\zeta, N)$ eines Fehlers hängt auch von der Art der Testauswahl ab:

1. statische Tests: Keine Abhängigkeit vom ζ der Fehler.
2. Zufallstests: Fehler mit $\zeta \ll N^{-1}$ werde nicht und mit $\zeta \gg N^{-1}$ sicher nachgewiesen. Dazwischen, siehe später

   (3.11) $$p_{FNE}(\zeta, N) = 1 - p_{FD}(\zeta, N) = e^{-\zeta \cdot N}$$

3. Vorab verwendete Vereinfachung (einfachere Integrale):

$$p_{FNE}(\zeta, N) = \begin{cases} 1 & \zeta \leq \frac{1}{N} \\ 0 & \text{sonst} \end{cases} \tag{2.16}$$

2.61 Typische Fehlerabdeckung von Zufallstests

Bei einem Zufallstest erfordert eine Verringerung des zu erwartenden Anteils der nicht nachweisbaren Fehler $1 - \mu_{FC}(N)$ um eine Dekade eine Erhöhung der Testanzahl N um mehr als eine Dekade.

Das ist die Eigenschaft einer Potenzfunktion:

$$\mu_{FC} = 1 - \frac{\mu_F(N_2)}{\mu_F(N_1)} = 1 - \left(\frac{N_2}{N_1}\right)^{-K} \quad \text{mit} \quad 0 < K < 1 \tag{2.17}$$

K	1	0,5	0,33	0,25
$\frac{N_2}{N_1}$ für $\mu_{FC} = 0{,}1$	10	100	10^3	10^4

Formfaktor (Rechengröße, Testobjektmerkmal):

$$K = -\log\left(\frac{\mu_F(N_2)}{\mu_F(N_1)}\right) / \log\left(\frac{N_2}{N_1}\right) \tag{2.18}$$

μ_{FC}	Zu erwartende Fehlerabdeckung.
$\mu_F(N)$	Zu erwartende Anzahl der Fehler, die nach N Tests nicht erkannt und beseitigt sind.
K	Formfaktor der Dichte der Fehlfunktionsrate ($0 < K < 1$).
N_1, N_2	Testanzahl mit bekannter oder gesuchter zu erwartender Fehleranzahl.

2.62 Dichte der MF-Rate

Mit der vereinfachten Nichtbeseitigungswahrscheinlichkeit

$$p_{\mathrm{FNE}}(\zeta, N) = \begin{cases} 1 & \zeta \le \frac{1}{N} \\ 0 & \text{sonst} \end{cases} \tag{2.16}$$

der zu erwartenden Fehleranzahl

$$\mu_{\mathrm{F}}(N) = \mu_{\mathrm{F}} \cdot \int_0^1 p_{\mathrm{FNE}}(\zeta, N) \cdot h(\zeta) \cdot \mathrm{d}\zeta \tag{2.14}$$

und der empirischen Abschätzung der Abnahme der Fehleranzahl:

$$\mu_{\mathrm{FC}} = 1 - \frac{\mu_{\mathrm{F}}(N_2)}{\mu_{\mathrm{F}}(N_1)} = 1 - \left(\frac{N_2}{N_1}\right)^{-K} \quad \text{mit} \quad 0 < K < 1 \tag{2.17}$$

$$\frac{\mu_{\mathrm{F}}(N_2)}{\mu_{\mathrm{F}}(N_1)} = \left(\frac{N_2}{N_1}\right)^{-K} \tag{2.19}$$

Muss die Dichte der MF-Rate $h(\zeta)$ folgende Bedingung erfüllen:

$$\left(\frac{N_2}{N_1}\right)^{-K} = \frac{\int_0^{\frac{1}{N_2}} h(\zeta) \cdot \mathrm{d}\zeta}{\int_0^{\frac{1}{N_1}} h(\zeta) \cdot \mathrm{d}\zeta} \tag{2.20}$$

$$\left(\frac{N_2}{N_1}\right)^{-K} = \frac{\int_0^{\frac{1}{N_2}} h(\zeta) \cdot \mathrm{d}\zeta}{\int_0^{\frac{1}{N_1}} h(\zeta) \cdot \mathrm{d}\zeta} \tag{2.20}$$

Die passende Dichtefunktion für $\zeta \in (0,1)$ ist die Potenzfunktion:

$$h(\zeta) = K \cdot \zeta^{K-1} \quad \text{mit} \quad 0 < K < 1 \quad \text{und} \; 0 < \zeta \le 1 \tag{2.21}$$

Kontrolle:

$$\int_0^1 K \cdot \zeta^{K-1} = 1 \surd$$

$$\int_0^{\frac{1}{N}} K \cdot \zeta^{K-1} \cdot \mathrm{d}\zeta = N^{-K} \surd$$

2.64 Fehlfunktionsrate und Testanzahl

$$h(\zeta) = K \cdot \zeta^{K-1} \quad \text{mit} \quad 0 < K < 1 \quad \text{und} \; 0 < \zeta \le 1 \tag{2.21}$$

eingesetzt in

$$\zeta_{\mathrm{F}}(N) \overset{(\le 1)}{=} \mu_{\mathrm{F}} \cdot \underbrace{\int_0^1 p_{\mathrm{FNE}}(\zeta, N) \cdot h(\zeta) \cdot \zeta \cdot \mathrm{d}\zeta}_{\text{mittlere Fehlfunktionsrate je Fehler}} \tag{2.15}$$

mit $p_{\mathrm{FNE}}(\zeta, N) = 1$ für $\zeta \le 1/N$ sonst 0 ergibt sich nach Beseitigung aller erkennbaren Fehler eine Fehlfunktionsrate von:

$$\zeta_{\mathrm{F}}(N) \overset{(\le 1)}{=} \mu_{\mathrm{F}} \cdot \int_0^{\frac{1}{N}} K \cdot \zeta^{K-1} \cdot \zeta \cdot \mathrm{d}\zeta$$

$$\zeta_{\mathrm{F}}(N) \overset{(\le 1)}{=} \mu_{\mathrm{F}} \cdot \frac{K}{K+1} \cdot N^{-(K+1)} \tag{2.22}$$

Abnahme mit Exponent $K+1$ mit der Testanzahl:

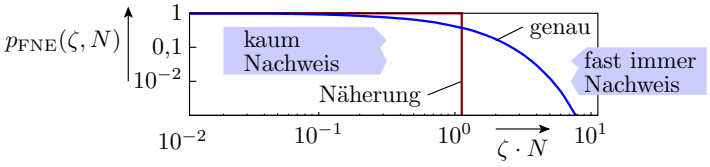
$$\zeta_{\mathrm{F}}(N_2) \overset{(\le 1)}{=} \zeta_{\mathrm{F}}(N_1) \cdot \left(\frac{N_2}{N_1}\right)^{-(K+1)} \tag{2.23}$$

$h(\zeta, N)$ Dichte der Fehlfunktionsrate nach Beseitigung der mit N Tests nachweisbaren Fehler.

$\zeta_{\mathrm{F}}(N)$ Fehlfunktionsrate durch Fehler in Abhängigkeit von der Testanzahl.

(≥ 1) Der errechnete Wert ist eine Obergrenze. Der tatsächliche Wert ist max. eins.

2.65 Verhältnis Fehlfunktionrate, Fehleranzahl



(2.22) $$\zeta_{\mathrm{F}}(N) \overset{(\leq 1)}{=} \mu_{\mathrm{F}} \cdot \frac{K}{K+1} \cdot N^{-(K+1)}$$

(2.19) $$\frac{\mu_{\mathrm{F}}(N_2)}{\mu_{\mathrm{F}}(N_1)} = \left(\frac{N_2}{N_1}\right)^{-K}$$

$$\zeta_{\mathrm{F}}(N) \overset{(\leq 1)}{=} \frac{K}{K+1} \cdot \frac{\mu_{\mathrm{F}}(N)}{N} \tag{2.24}$$

Bei genauerer Modellierung entfällt der Term $K+1$ (Folie 3.48):

(3.18) $$\zeta_{\mathrm{F}}(N) \overset{(\leq 1)}{=} K \cdot \frac{\mu_{\mathrm{F}}}{N^{K+1}}$$

$$\zeta_{\mathrm{F}}(N) \overset{(\leq 1)}{=} K \cdot \frac{\mu_{\mathrm{F}}(N)}{N} \tag{2.25}$$

Wir lassen den Term $K+1$ in Beispielabschätzungen schon jetzt weg.

2.67 Fehlfunktionsrate, Fehleranzahl, Formfaktor

(2.23) $$\zeta_{\mathrm{F}}(N_2) \overset{(\leq 1)}{=} \zeta_{\mathrm{F}}(N_1) \cdot \left(\frac{N_2}{N_1}\right)^{-(K+1)}$$

(2.25) $$\zeta_{\mathrm{F}}(N) \overset{(\leq 1)}{=} K \cdot \frac{\mu_{\mathrm{F}}(N)}{N}$$

Die Fehlfunktionsrate

- nimmt mit Exponet $K+1$ mit der Testanzahl ab und
- und ist proportionale zum Verhältnis aus Fehler- und Testanzahl.

Voraussetzung:

- Test mit Nutzungsprofil, Beseitigung aller nachweisbaren Fehler,
- Abnahme der Anzahl der nicht nachweisbaren Fehler etwa mit

(2.19) $$\frac{\mu_{\mathrm{F}}(N_2)}{\mu_{\mathrm{F}}(N_1)} = \left(\frac{N_2}{N_1}\right)^{-K}$$

Abschätzung Formfaktor aus der Abnahme der der Fehlfunktionsrate:

$$K = \log\left(\frac{\zeta_{\mathrm{F}}(N_1)}{\zeta_{\mathrm{F}}(N_2)}\right) / \log\left(\frac{N_2}{N_1}\right) - 1 \tag{2.26}$$

(≥ 1) Der errechnete Wert ist eine Obergrenze. Der tatsächliche Wert ist max. eins.
$\zeta_{\mathrm{F}}(N)$ Fehlfunktionsrate durch Fehler in Abhängigkeit von der Testanzahl.
N_1, N_2 Testanzahl mit bekannter / gesuchter Fehlfunktionsrate oder Fehleranzahl.
K Formfaktor der Dichte der Fehlfunktionsrate ($0 < K < 1$).

2.68 Aufteilung in Vortest und Zufallstest

Vor einem gründlichen Zufallstest erfolgen Vortests:

- statische Tests: Reviews, Syntax, ...
- Grobtests, ob überhaupt etwas funktioniert und
- gezielt gesuchte Tests für Grenz- und Sonderfälle.

Bei statischen und fehlerorientiert gesuchten Tests hängt $p_{\mathrm{FNE}}(\zeta, N)$ weniger von ζ als beim Zufallstest ab. Pauschalannahme, dass alle Vortests zusammen einen Anteil von FC_{PT} Fehler erkennen, die alle beseitigt werden und $N_0 \geq 1$ dynamische Tests enthalten:

$$\mu_{\mathrm{F}}(N_0) = \mu_{\mathrm{FCR}} \cdot (1 - FC_{\mathrm{PT}}) \tag{2.27}$$

$$\zeta_{\mathrm{F}}(N_0) \overset{(\leq 1)}{=} \frac{K \cdot \mu_{\mathrm{F}}(N_0)}{N_0} \tag{2.28}$$

$p_{\mathrm{FNE}}(\zeta, N)$	Wahrscheinlichkeit, dass Fehler mit MF-Rate ζ nach N Tests nicht beseitigt sind.
$\mu_{\mathrm{F}}(N_0)$	Zu erwartende Anzahl der Fehler, die nach N_0 Tests nicht erkannt und beseitigt sind.
N_0	Anzahl der dynamischen Tests aller Vortests zusammen.
μ_{FCR}	Zu erwartende Fehleranzahl aus den Entstehungs- und Reparaturprozessen insgesamt.
FC_{PT}	Fehlerabdeckung aller Vortests zusammen.
$\zeta_{\mathrm{F}}(N_0)$	Fehlfunktionsrate nach Beseitigung der von Vortests erkannten Fehler.

2.69 Abnahme bei weiterer Testanzahlerhöhung

Die Vortests finden die schwerwiegensten und auch die meisten Fehler. Die sich anschließenden Zuverlässigkeitstests erhöhen die Testsanzahl anschließend von N_0 auf insgesamt N Tests (Ersatz $N_1 \rightarrow N_0$ und $N_2 \rightarrow N$ in Gl. 2.19 und 2.23) und mindern die zu erwartende Fehleranzahl und Fehlfunktionsrate auf:

$$\mu_{\mathrm{F}}(N) = \mu_{\mathrm{F}}(N_0) \cdot \left(\frac{N}{N_0}\right)^{-K} \quad \text{mit} \quad 0 < K < 1 \tag{2.29}$$

$$\zeta_{\mathrm{F}}(N) \overset{(\leq 1)}{=} \zeta_{\mathrm{F}}(N_0) \cdot \left(\frac{N}{N_0}\right)^{-(K+1)} \tag{2.30}$$

Zu erwartende Fehlerabdeckung:

$$\mu_{\mathrm{FC}}(N) = 1 - \left(\frac{N}{N_0}\right)^{-K} \tag{2.31}$$

$\mu_{\mathrm{F}}(N)$	Zu erwartende Anzahl der Fehler, die nach N Tests nicht erkannt und beseitigt sind.
$\mu_{\mathrm{F}}(N_0)$	Zu erwartende Anzahl der Fehler, die nach N_0 Tests nicht erkannt und beseitigt sind.
N, N_0	Gesamttestanzahl, Testanzahl der dynamischen Vortests.
K	Formfaktor der Dichte der Fehlfunktionsrate ($0 < K < 1$).
$\zeta_{\mathrm{F}}(N)$	Fehlfunktionsrate durch Fehler in Abhängigkeit von der Testanzahl.
(≥ 1)	Der errechnete Wert ist eine Untergrenze. Der tatsächliche Wert ist mindestens eins.

3.3 Zuverl. & Sicherheit

2.70 Fehlerbezogene Teilzuverlässigkeit

(2.30) $\zeta_{\mathrm{F}}(N) \overset{(\leq 1)}{=} \zeta_{\mathrm{F}}(N_0) \cdot \left(\frac{N}{N_0}\right)^{-(K+1)}$

(2.25) $\zeta_{\mathrm{F}}(N) \overset{(\leq 1)}{=} K \cdot \frac{\mu_{\mathrm{F}}(N)}{N}$

Fehlerbezogenen Teilzuverlässigkeit als Kehrwert der MF-Rate:

$$R_{\mathrm{F}}(N) \overset{(\geq 1)}{=} = \frac{1}{\zeta_{\mathrm{F}}(N)} = R_{\mathrm{F}}(N_0) \cdot \left(\frac{N}{N_0}\right)^{K+1} \tag{2.32}$$

$$R_{\mathrm{F}}(N) \overset{(\geq 1)}{=} \frac{N}{K \cdot \mu_{\mathrm{F}}(N)} \tag{2.33}$$

Voraussetzung:

- Test mit Nutzungsprofil,

- Beseitigung aller nachweisbaren Fehler, ...

$\zeta_{\mathrm{F}}(N)$	Fehlfunktionsrate durch Fehler in Abhängigkeit von der Testanzahl.
$\mu_{\mathrm{F}}(N)$	Zu erwartende Anzahl der Fehler, die nach N Tests nicht erkannt und beseitigt sind.
K	Formfaktor der Dichte der Fehlfunktionsrate ($0 < K < 1$).
N	Effektive Testanzahl, für die alle erkannten Fehler beseitigt werden.
$R_{\mathrm{F}}(N)$	Fehlerbezogene Teilzuverl. nach Beseitigung aller mit N Tests nachweisbaren Fehler.
(≥ 1)	Der errechnete Wert ist eine Obergrenze. Der tatsächliche Wert ist max. eins.

2.71 Zuverl. mit Fehlfunktionsbehandlung

Bei einem System mit Fehlfunktionsbehandlung, das keine erkannte Fehlfunktion »weitergibt«, ist die Zuverlässigkeit der Kehrwert der Rate der nicht erkannten Fehlfunktionen, einschließlich der durch Störungen:

$$R_{\mathrm{MT}}(N) \overset{(\geq 1)}{=} \frac{1}{(\zeta_{\mathrm{F}}(N) + \zeta_{\mathrm{D}}) \cdot (1 - MC)} \tag{2.34}$$

Mit der Testanzahl nimmt nur die Fehlfunktionsrate durch Fehler ab:

$$R_{\mathrm{MT}}(N) \overset{(\geq 1)}{=} \frac{N}{(K \cdot \mu_{\mathrm{F}}(N) + N \cdot \zeta_{\mathrm{D}}) \cdot (1 - \mu_{\mathrm{MC}})}$$

Wenn Fehlfunktionen durch Störungen vernachlässigbar sind:

$$R_{\mathrm{MT}}(N) \overset{(\geq 1)}{=} \frac{N}{K \cdot \mu_{\mathrm{F}}(N) \cdot (1 - \mu_{\mathrm{MC}})} \tag{2.35}$$

R_{MT}	Zuverlässigkeit mit Fehlfunktionsbehandlung.
ζ_{F}	Fehlfunktionsrate durch Fehler.
N	Anzahl der Tests.
ζ_{D}	Fehlfunktionsrate durch Störungen (Malfunction rate due to disturbance).
μ_{MC}	Zu erwartende Fehlfunktionsabdeckung der Überwachung im Betrieb.
K	Formfaktor der Dichte der Fehlfunktionsrate ($0 < K < 1$).

2.72 Teilzuverlässigkeiten und Sicherheit

Fehler durch Ausfälle, sicherheitsgefährdene Fehler, ... unterschiedlicher Umgang, Berücksichtigung in getrennten Teilzuverlässigkeiten. Tendentiell gilt für alle fehlerbezogene Teilzuverlässigkeiten:

$$R_{\mathrm{x}} \sim \frac{N^{K+1}}{\mu_{\mathrm{Fx}}} \tag{2.36}$$

Unter Vernachlässigung von Fehlfunktionen durch Störungen und Ausfälle, bei Beseitigung alle erkannten Fehler und Übergang des Systems bei jedem erkannten Problem in einen sicheren Zustand, wachsen Zuverlässigkeit und Sicherheit mit der $K+1$-ten Potenz der Testanzahl:

$$\frac{R_{\mathrm{MT}}(N_2)}{R_{\mathrm{MT}}(N_1)} = \frac{S(N_2)}{S(N_1)} = \left(\frac{N_2}{N_1}\right)^{K+1} \tag{2.37}$$

R_{x}, μ_{Fx}	Teilzuverlässigkeit x, Fehleranzahl vor dem Test, die R_{x} zugeordnet sind.
N, K	Effektive Testanzahl, Formfaktor $0 < K < 1$.
R_{MT}	Zuverlässigkeit mit Fehlfunktionsbehandlung.
S	Sicherheit (Safety or security).
N_1, N_2	Testanzahl mit bekannter oder gesuchter Zuverlässigkeit.

2.73 Weitere nützliche Abschätzungen

Fehlerabdeckung und Zuverlässigkeit unter Vernachlässigigung von Störungen als Funktion der Fehlerabdeckung:

(2.17) $$\mu_{\mathrm{FC}} = 1 - \frac{\mu_{\mathrm{F}}(N_2)}{\mu_{\mathrm{F}}(N_1)} = 1 - \left(\frac{N_2}{N_1}\right)^{-K} \quad \text{mit} \quad 0 < K < 1$$

(2.35) $$R_{\mathrm{MT}}(N) \overset{(\geq 1)}{=} \frac{N}{K \cdot \mu_{\mathrm{F}}(N) \cdot (1 - \mu_{\mathrm{MC}})}$$

$$R \overset{(\geq 1)}{=} \frac{N}{K \cdot \mu_{\mathrm{F}} \cdot (1 - \mu_{\mathrm{FC}}) \cdot (1 - \mu_{\mathrm{MC}})} \tag{2.38}$$

R	Zuverlässigkeit mit Fehlfunktionsbehandlung unter Vernachlässigung von Störungen.
N, K	Effektive Testanzahl, Formfaktor $0 < K < 1$.
μ_{MC}	Zu erwartende Fehlfunktionsabdeckung der Überwachung im Betrieb.
μ_{F}	Zu erwartende Fehleranzahl vor der Iteration aus Test und Fehlerbeseitigung.
μ_{FC}	Zu erwartende Fehlerabdeckung.

Beispiel 2.3 Zuverlässigkeit dreifacher Testaufwand

a) *Um welche Faktoren nehmen MF-Rate $\zeta_{\mathrm{F}}(N)$ und Fehleranzahl $\mu_{\mathrm{F}}(N)$ ab, wenn die Anzahl der dynamischen Tests verdreifacht wird? Formfaktoren der Verteilung der MF-Rate $K \in \{0{,}3,\ 0{,}5\}$.*

Geschätzte Reduzierung der MF-Rate und der Fehlerzahl sowie die Erhöhung der Zuverlässigkeit als Kehrwert der MF-Rate:

$$\frac{\mu_{\mathrm{F}}(3\cdot N)}{\mu_{\mathrm{F}}(N)} = 3^{-K};\quad \frac{\zeta_{\mathrm{F}}(3\cdot N)}{\zeta_{\mathrm{F}}(N)} = 3^{-(K+1)};\quad \frac{R_{\mathrm{F}}(3\cdot N)}{R_{\mathrm{F}}(N)} = 3^{K+1}$$

	$\frac{\mu_{\mathrm{F}}(3\cdot N)}{\mu_{\mathrm{F}}(N)}$	$\frac{\zeta_{\mathrm{F}}(3\cdot N)}{\zeta_{\mathrm{F}}(N)}$	$\frac{R_{\mathrm{F}}(3\cdot N)}{R_{\mathrm{F}}(N)}$
$K = 0{,}3$	0,72	0,24	4,17
$K = 0{,}5$	0,56	0,19	5,19

Die Fehleranzahl verringert sich auf 56% bis 72% und die Fehlfunktionsrate durch nicht beseitigte Fehler auf 19% bis 24%.

Die Rechengröße K kennen wir in der Regel nicht so genau. Für die Abnahme der Fehleranzahl hat K einen großen, aber für die Abnahme die Zuverlässigkeitsverbesserung nur moderaten Einfluss.

b) *Welche Erhöhung der Zuverlässigkeit ist unter Vernachlässigung der Fehlfunktionen durch Störungen zu erwarten, wenn das Personal der Testabteilung verdreifacht wird?*

Wenn 3-facher Personaleinsatz den dreifachen Testaufwand impliziert, zu erwartende Erhöhung der Zuverlässigkeit auf etwa das 4- bis 5-fache.

$\zeta_{\mathrm{F}}(N)$	Fehlfunktionsrate durch Fehler in Abhängigkeit von der Testanzahl.
$\mu_{\mathrm{F}}(N)$	Zu erwartende Anzahl der Fehler, die nach N Tests nicht erkannt und beseitigt sind.
K	Formfaktor der Dichte der Fehlfunktionsrate ($0 < K < 1$).
$R(N)$	Zuverlässigkeit nach Beseitigung aller mit den N Tests nachweisbaren Fehler.

3.4 Effektive Testanzahl

2.75 Effektive Testanzahl

In den bisherigen Abschätzungen ist N die Anzahl der Tests, für die alle erkennbaren Fehler beseitigt werden. Es gibt Situationen, in denen das ein Bruchteil oder ein Vielfaches der tatsächlichen Testsanzahl ist:

- Modularer Test: Modultests erkennen dieselben Fehler tendentiell nur einen Bruchteil der Testanzahl für einen ganzheitlichen Zufallstests (Abschn. 2.3.5).
- Die Fehlfunktionsrate von Modellfehlern kann im Mittel größer oder kleiner als die tatsächlichen Fehler sein (Abschn. 2.3.6).
- Reifeprozess: Fehler in genutzter Software müssen viel Fehlfunktionen verursachen, bevor er beseitigt werden (Abschn. 2.3.7).

Berücksichtigung durch den Skalierungsfaktor C in Gl. 2.39.

$$N = C \cdot N_{\mathrm{T}} \tag{2.39}$$

N	Effektive Testanzahl, für die alle erkannten Fehler beseitigt werden.
N_T	Tatsächliche Testanzahl.
C	Testskalierung, Verhältnis von effektiver und tatsächlicher Testanzahl.

Der Skalierungsfaktor hat keinen Einfluss auf die relative Erhöhung der Testanzahl in

(2.17) $$\mu_\mathrm{FC} = 1 - \frac{\mu_\mathrm{F}(N_2)}{\mu_\mathrm{F}(N_1)} = 1 - \left(\frac{N_2}{N_1}\right)^{-K} \quad \text{mit} \quad 0 < K < 1$$

(2.23) $$\zeta_\mathrm{F}(N_2) \overset{(\leq 1)}{=} \zeta_\mathrm{F}(N_1) \cdot \left(\frac{N_2}{N_1}\right)^{-(K+1)}$$

(2.37) $$\frac{R_\mathrm{MT}(N_2)}{R_\mathrm{MT}(N_1)} = \frac{S(N_2)}{S(N_1)} = \left(\frac{N_2}{N_1}\right)^{K+1}$$

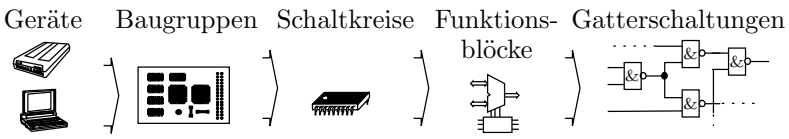
Nur in (Gl. 2.25) hat die Testskalierung Einfluss:

$$\zeta_\mathrm{F}(N_\mathrm{T}) = \frac{K \cdot \mu_\mathrm{F}(N)}{N} = \frac{K \cdot \mu_\mathrm{F}(c \cdot N_\mathrm{T})}{C \cdot N_\mathrm{T}} \tag{2.40}$$

$\mu_\mathrm{F}(N)$	Zu erwartende Anzahl der Fehler, die nach N Tests nicht erkannt und beseitigt sind.
N_1, N_2	(Effektive) Testanzahl mit bekannter / gesuchter Fehlfunktionsrate oder Fehleranzahl.
R_MT	Zuverlässigkeit mit Fehlfunktionsbehandlung.
S	Sicherheit (Safety or security).
N, N_T	Effektive Testanzahl, tatsächliche Testanzahl.
C	Testskalierung, Verhältnis von effektiver und tatsächlicher Testanzahl.

3.5 Modularer Test

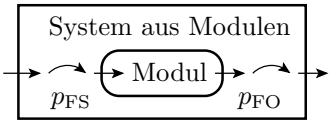
2.77 Modularer Test



- Rechner-Systeme bestehen aus Rechnern, EA-Geräten, Druckern, Netzwerkkomponenten, diese aus ...
- Die Hardware stellt der SW Grundfunktionen (Maschinenbefehle, Ein- und Ausgabe, ...).
- Software gliedert sich in Teilsysteme, Module, Bibliotheken, ...

Die durchgeführten Tests folgen der Hierarchie. Wenn möglich, werden die Bausteine vor Übernahme in das übergeordnete System gründlich getestet (Abschn. 2.1.2 *Vielfalt der Test*).

Der übergeordnete Test zielt hauptsächlich auf Fehler im Zusammenwirken (vergl. auch Leiterplattentest, Folie 2.47). Ein Grund die deutlich größere effektive Testanzahl von Modultests $C \gg 1$ (Abschn. 2.3.4).



Modulinterne Fehler werden bei Einbettung in ein übergeordnetes System im Mittel um die Anregungswahrscheinlichkeit p_{FS} seltener angeregt und fehlerverursachte Verfälschungen am Modulausgang verfälschen nur mit einer Beobachtbarkeit $p_{\mathrm{FO}} \leq 1$ die Systemausgabe. Fehlfunktionsrate eines modulinterenen Fehlers

- wenn das Modul isoliert getestet wird: ζ_{Mod}
- beim eingebetteten Test des Moduls im System: $p_{\mathrm{FS}} \cdot \zeta_{\mathrm{Mod}} \cdot p_{\mathrm{FO}}$

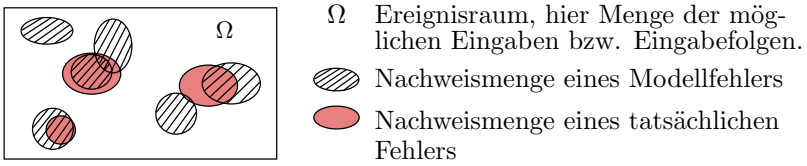
Für einen genauso häufigen Nachweis ist die $\frac{1}{p_{\mathrm{FS}} \cdot p_{\mathrm{FO}}}$-fache Anzahl von Systemtests erforderlich:

$$N = C \cdot N_{\mathrm{M}} \quad \text{mit} \quad C = \frac{1}{p_{\mathrm{FS}} \cdot p_{\mathrm{FO}}} \gg 1 \tag{2.41}$$

p_{FS}	Fehleranregungswahrscheinlichkeit (Probability of fault stimulation).
p_{FO}	Fehlerbeobachtbarkeitswahrscheinlichkeit (Probability of fault observation).
N	Effektive Testanzahl, für die alle erkannten Fehler beseitigt werden.
C, N_{M}	Testskalierung, Anzahl der Modultests.

3.6 Fehlermodellskalierung

2.79 Fehler und Modellfehler



Ein gutes Fehlermodell generiert für (fast) alle zu erwartenden Fehler Mengen ähnlich nachweisbare Modellfehler, die sich mit den tatsächlichen Infektions- und Ausbreitungsbedingungen teilen. Je nach Wahl des Fehlermodells können die Modellfehler tendentiell

- schlechter oder
- besser

als die tatsächlichen Fehler nachweisbar sein.

2.80 Effektive Testanzahl und Fehlerabdeckung

Skalierung der effektiven Testanzahl:

$$N = C \cdot N_{\mathrm{MF}}$$

- Modellfehler schlechter nachweisbar: $C < 1$
- Modellfehler besser nachweisbar: $C > 1$

Bei Skalierung ist die zu erwartende Fehlerabdeckung ungefähr die zu erwartenden Modellfehlerabdeckung der C-fachen Testanzahl:

$$N = C \cdot N_{\mathrm{MF}} \quad \text{für } \mu_{\mathrm{FC}}(N) = \mu_{\mathrm{FCM}}(N_{\mathrm{MF}}) \tag{2.42}$$

In Abschn. 6.1.4 wird später gezeigt, dass für zu erwartende Schaltkreisfehler (Kurzschlüsse, Unterbrechungen, Transistorfehler) und Haftfehler grob Testskalierung $C \approx 0{,}5 \ldots 1$ zu erwarten ist. Haftfehlertests sind für die gleiche tatsächliche Fehlerabdeckung tendentiell mit der ein- bis zweifachen Testanzahl zu simulieren.

N, C	Effektive Testanzahl, Testskalierung.
N_{MF}	Testanzahl, mit der die Modellfehlerabdeckung bestimmt wird.
μ_{FC}	Zu erwartende Fehlerabdeckung.
μ_{FCM}	Zu erwartende Modellfehlerabdeckung.

3.7 Reifeprozesse

2.81 Das Problem immer größerer IT-Systeme

Die zu erwartende Fehleranzahl wächst proportional zur Systemgröße bzw. zum Entstehungsaufwand (siehe später Abschn. 2.4.1):

(2.56) $$\mu_{\mathrm{CF}} = \xi_{<\mathrm{C}>} \cdot M_{\mathrm{C}}$$

Nach Beseitigung der von den Vor- und Zufallstest gefundenen Fehler:

(2.27) $$\mu_{\mathrm{F}}(N_0) = \mu_{\mathrm{FCR}} \cdot (1 - FC_{\mathrm{PT}})$$

(2.29) $$\mu_{\mathrm{F}}(N) = \mu_{\mathrm{F}}(N_0) \cdot \left(\frac{N}{N_0}\right)^{-K} \quad \text{mit} \quad 0 < K < 1$$

(2.25) $$\zeta_{\mathrm{F}}(N) \overset{(\leq 1)}{=} K \cdot \frac{\mu_{\mathrm{F}}(N)}{N}$$

(2.32) $$R_{\mathrm{F}}(N) \overset{(\geq 1)}{=} = \frac{1}{\zeta_{\mathrm{F}}(N)} = R_{\mathrm{F}}(N_0) \cdot \left(\frac{N}{N_0}\right)^{K+1}$$

$\xi_{<\mathrm{C}>}$ Fehlerentstehungsrate in Fehlern je Bezugsgröße der Metrik M_{C}.
M_{C} Metrik für den Entstehungsaufwand oder die Größe des Produkts.
μ_{CF} Zu erwartende Anzahl der Fehler aus den Entstehungsprozessen.
μ_{FCR} Zu erwartende Fehleranzahl aus den Entstehungs- und Reparaturprozessen insgesamt.
N_0 Anzahl der dynamischen Tests aller Vortests zusammen.
Aussprache: ξ: xi, μ: my.

Zuverlässigkeitsabnahme mit der Systemgröße M_{C}:

$$R_{\mathrm{F}}(N) = \frac{N}{K \cdot \mu_{\mathrm{F}}(N_0)} \cdot \left(\frac{N}{N_0}\right)^{K} = \ldots \sim \frac{N^{K+1}}{M_{\mathrm{C}}}$$

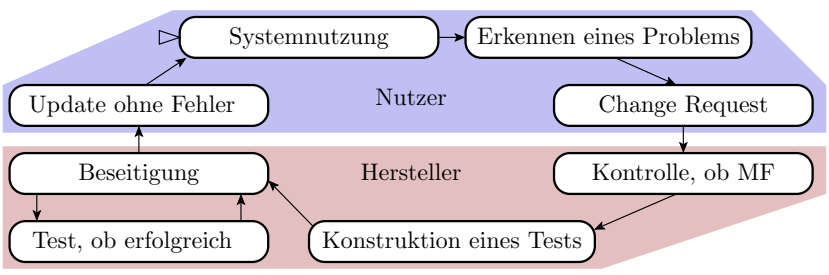
Die Kompensation des Zuverlässigkeitsverlust durch den immer größeren Entstehungsaufwand bzw. die wachsende Systemgröße, beschrieben durch die Metrik M_{C}, verlangt eine größere effektive Testanzahl N.

Die Größe M_{C} der IT-Systeme nimmt über die Jahre exponentiell zu, der erbringbare Testaufwand N ist durch Zeit und Personal begrenzt.

Was tun gegen den drohenden Zuverlässigkeitsverlust?

$\zeta_{\mathrm{F}}(N)$ Fehlfunktionsrate durch Fehler in Abhängigkeit von der Testanzahl.
$R_{\mathrm{F}}(N)$ Fehlerbezogene Teilzuverl. nach Beseitigung aller mit N Tests nachweisbaren Fehler.
$\mu_{\mathrm{F}}(N)$ Zu erwartende Anzahl der Fehler, die nach N Tests nicht erkannt und beseitigt sind.
N_0 Anzahl der dynamischen Tests aller Vortests zusammen.
N Effektive Testanzahl, für die alle erkannten Fehler beseitigt werden.
K Formfaktor der Dichte der Fehlfunktionsrate ($0 < K < 1$).
M_{C} Metrik für den Entstehungsaufwand oder die Größe des Produkts.

2.83 Reifen der Produkte in der Einsatzphase



Alternative zu immer längeren Testzeiten vor dem Einsatz ist die Fortsetzung der Fehlerbeseitigung im Einsatz mit den Nutzern als Tester.

- Erfassen Problemen (Abstürze, Fehlfunktionen, ...) im Einsatz.
- Zusammenstellen der Daten, um das Problem nachzustellen.
- Übermittlung an den Hersteller.
- Suche von Tests für einen reproduzierbaren Fehlernachweis.
- Beseitigung durch experimentelle Reparatur.
- Herausgabe und Installieren von Updates.

2.84 Fehlerbeseitigungswahrscheinlichkeit

Die Fehlerbeseitigungswahrscheinlichkeit p_{FE} ist die bedingte Wahrscheinlichkeit, dass, wenn ein Problem auftritt,

1. Nutzer oder System dieses erkennen,
2. an den Hersteller einen Problembericht senden,
3. das vermeindliche Problem vom Hersteller als solche bestätigt und für die Beseitigung priorisiert wird,
4. der Hersteller Tests für den Fehlernachweis findet,
5. den verursachenden Fehler findet und beseitigt und
6. der Anwender das Update nach Fehlerbeseitigung übernimmt.

(3) Sammeln von Problemberichte in Schubladen vermuteter gleicher Ursache. Bevorzugte Beseitigung häufiger schwerer Probleme.

Die Wahrscheinlichkeit p_{FE}, dass ein Fehler beseitigt wird, wenn er ein Problem verursacht, ist gering.

---

* Generieren und Versenden vorzugsweise automatisch. Manuelle Problembericht in der Regel Änderungsanfragen, die auch Änderungswünsche des Sollverhaltens enthalten.

2.85 Effektive Testanzahl

Reifende Produkte werden von vielen Nutzern über lange Zeit mit unzähligen Beispieleingaben genutzt. Geschätze effektive Testanzahl:

$$N = p_{\mathrm{FE}} \cdot \mu_{\mathrm{NU}} \cdot \eta_{\mathrm{SU}} \cdot (t_{\mathrm{M}} + t_{\mathrm{V0}}) \quad \text{mit} \quad t_{\mathrm{V0}} = \frac{N_{\mathrm{V0}}}{p_{\mathrm{FE}} \cdot \mu_{\mathrm{NU}} \cdot \eta_{\mathrm{SU}}} \tag{2.43}$$

Genau genommen nimmt die effektive Testanzahl nicht kontinuierlich mit der Reifedauer zu, sondern zeitdiskret mit den Versionsfreigaben. Zunahme der effektiven Testanzahl mit der Versions-Anzahl bei gleich langen Release-Intervallen:

$$N = \underbrace{p_{\mathrm{FE}} \cdot \mu_{\mathrm{NU}} \cdot \eta_{\mathrm{SU}} \cdot t_{\mathrm{VR}}}_{N_{\mathrm{MV}}} \cdot (u + u_{\mathrm{V0}}) \quad \text{mit} \quad u_{\mathrm{V0}} = \frac{N_{\mathrm{V0}}}{N_{\mathrm{VM}}} \tag{2.44}$$

N	Effektive Testanzahl, für die alle erkannten Fehler beseitigt werden.
p_{FE}	Wahrscheinlichkeit, dass ein Fehler beseitigt wird, wenn er eine MF verursacht.
μ_{NU}	Zu erwartende Nutzeranzahl (Expected **n**umber of **u**ser).
η_{SU}	Mittlere Anzahl der **S**ervice-Leistungen pro Nutzer (**u**ser) und Nutzungszeit.
t_{M}	Reifedauer (Maturing time).
N_{V0}	Effektive Testanzahl von Version 0, d.h. der Fehlerbeseitigungsiteration vor dem Einsatz.
t_{VR}	Versionsintervall, Zeit zwischen der Freigabe aufeinanderfolgender Version.
N_{MV}	Erhöhung der effektive Testanzahl mit jeder Version.
u	Versionsnummer des reifenden Objekts, Zählweis 0, 1, 2, ...

Aussprache: μ: my, η: eta.

2.86 Fehleranzahlabnahme mit der Reifedauer

Der Abschnitt betrachtet nur den vereinfachten Fall, dass bei der Fehlerbeseitigung keine neuen Fehler entstehen bzw. neue entstandene Fehler vor Versionsfreigabe gefunden und beseitigt werden. Ausgehend von

(2.17) $$\mu_{\mathrm{FC}} = 1 - \frac{\mu_{\mathrm{F}}(N_2)}{\mu_{\mathrm{F}}(N_1)} = 1 - \left(\frac{N_2}{N_1}\right)^{-K} \quad \text{mit } 0 < K < 1$$

mit Gl. 2.43 bzw. 2.44 nimmt die zu erwartende Fehleranzahl mit der K-ten Potenz der Reifedauer bzw. bei gleich langen Release-Intervallen mit der Versionsanzahl ab:

$$\mu_{\mathrm{F}}(t_{\mathrm{M}}) = \mu_{\mathrm{F}}(t_{\mathrm{M0}}) \cdot \left(\frac{t_{\mathrm{M}}+t_{\mathrm{V0}}}{t_{\mathrm{M0}}+t_{\mathrm{V0}}}\right)^{-K} \tag{2.45}$$

$$\mu_{\mathrm{F}}(u) = \mu_{\mathrm{F}}(v) \cdot \left(\frac{u+u_{\mathrm{V0}}}{v+u_{\mathrm{V0}}}\right)^{-K} \tag{2.46}$$

μ_{F}	Zu erwartende Fehleranzahl.
t_{M}	Reifedauer (Maturing time).
t_{V0}	Equivalente Reifedauer vor Freigabe von Version null.
t_{M0}	Bezugsreifedauer.
K	Formfaktor der Dichte der Fehlfunktionsrate ($0 < K < 1$).
u, v	Versionnummern und Bezugsversionsnummer des reifenden Objekts.
u_{V0}	Verhältnis der Reifedauer vor Versionsfreigabe zur Reifedauererhöhung je Version.

2.87 Fehlfunktionsrate und Zuverlässigkeit

Die Fehlfunktionsrate durch Fehler nimmt mit der $K+1-$ten Potenz der Reifedauer ab:

$$\zeta_{\mathrm{F}}(t_{\mathrm{M}}) = \zeta_{\mathrm{F}}(t_{\mathrm{M0}}) \cdot \left(\frac{t_{\mathrm{M}}+t_{\mathrm{V0}}}{t_{\mathrm{M0}}+t_{\mathrm{V0}}}\right)^{-(K+1)} \tag{2.47}$$

$$\zeta_{\mathrm{F}}(u) = \zeta_{\mathrm{F}}(v) \cdot \left(\frac{u+u_{\mathrm{V0}}}{v+u_{\mathrm{V0}}}\right)^{-(K+1)} \tag{2.48}$$

Durch digitale Verarbeitung, elektromagnetische Verträglichkeit, Datenübertragung und Speicherung mit Prüfkennzeichen, ... sind Fehlfunktionen durch Störungen oft vernachlässigbar. Wenn das der Fall, ist Zuverlässigkeit der Kehrwert der Fehlfunktionsrate durch Fehler:

$$R_{\mathrm{MT}}(t_{\mathrm{M}}) = R_{\mathrm{MT}}(t_{\mathrm{M0}}) \cdot \left(\frac{t_{\mathrm{M}}+t_{\mathrm{V0}}}{t_{\mathrm{M0}}+t_{\mathrm{V0}}}\right)^{K+1} \tag{2.49}$$

$$R_{\mathrm{MT}}(u) = R_{\mathrm{MT}}(v) \cdot \left(\frac{u+u_{\mathrm{V0}}}{v+u_{\mathrm{V0}}}\right)^{K+1} \tag{2.50}$$

ζ_{F}	Fehlfunktionsrate durch Fehler.
R_{MT}	Zuverlässigkeit mit Fehlfunktionsbehandlung.
u, v	Versionnummern und Bezugsversionsnummer des reifenden Objekts.
u_{V0}	Verhältnis der Reifedauer vor Versionsfreigabe zur Reifedauererhöhung je Version.

2.88 Sicherheit

Wenn bei allen erkannten Problemen ein sicherer Zustand hergestellt wird, ist nur von den nicht erkannten Fehlfunktionen ein Anteil ρ sicherheitsgefährdent:

(1.24) $$S = \frac{R_{\mathrm{MT}}}{\rho}$$

Bei zusätzlicher Vernachlässigung der Fehlfunktionen durch Störungen wächst die Sicherheit genau wie die Zuverlässigkeit mit der $K+1$-ten Potenz der Reifedauer bzw. Versionsanzahl:

$$\frac{S(t_{\mathrm{M}})}{S(t_{\mathrm{M0}})} = \frac{R_{\mathrm{MT}}(t_{\mathrm{M}})}{R_{\mathrm{MT}}(t_{\mathrm{M0}})} = \left(\frac{t_{\mathrm{M}}+t_{\mathrm{V0}}}{t_{\mathrm{M0}}+t_{\mathrm{V0}}}\right)^{K+1} \tag{2.51}$$

$$\frac{S(u)}{S(v)} = \frac{R_{\mathrm{MT}}(u)}{R_{\mathrm{MT}}(v)} = \left(\frac{u+u_{\mathrm{V0}}}{v+u_{\mathrm{V0}}}\right)^{K+1} \tag{2.52}$$

S	Sicherheit (Safety or security).

R_{MT} Zuverlässigkeit mit Fehlfunktionsbehandlung.
ρ Anteil sicherheitskritischer Fehlfunktionen an den nicht erkannten Fehlfunktionen.
K Formfaktor der Dichte der Fehlfunktionsrate ($0 < K < 1$).
u, v Versionnummern und Bezugsversionsnummer des reifenden Objekts.
u_{V0} Verhältnis der Reifedauer vor Versionsfreigabe zur Reifedauererhöhung je Version.

2.89 Hohe Zuverlässigkeit und Sicherheit

Hohe Zuverlässigkeit und Sicherheit verlangen:

- Entstehungsprozesse mit geringer Fehlerentstehungsrate,
- gründliche Tests vor dem Einsatz,
- hohe Fehlerabdeckung der Tests vor dem Einsatz,
- sicherer Zustand für alle erkennbaren Probleme, ...

Zusätzlich Reifeprozesse mit sehr großer effektiver Testanzahl:

- Systemfunktionen / organisatorischer Rahmen / Kontrollen, die bewirken, dass Fehlfunktionen bei Nutzern oft (z.B. jedes 10^{-3}-te mal) eine Fehlerbeseitigung zu Folge haben,
- viele Nutzer, lange Reifedauer, ...

Systeme, die viele Jahre gereift sind, haben hohe, auf anderem Wege unerreichbare Zuverlässigkeiten und Sicherheiten. 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.

2.90 Reifeprozess des Nutzerverhaltens

Ein ähnliches Zuverlässigkeits- und Sicherheitswachstum ist auch ohne Fehlerbeseitigung mit der Nutzungsdauer zu beobachten.

Komplexe IT-Systeme bieten oft viele Lösungswege für eine Aufgabe, die nicht alle funktionieren (vergl. Folie 1.104). Mit zunehmender Nutzung lernt der Nutzer problematische Eingaben zu vermeiden und seine Service-Anforderungen an die Möglichkeiten des Systems anzupassen. Zuverlässigszunahme mit der Nutzungdauer ähnlich (Gl. 2.49).

Die für das Zuverlässigkeitswachstum verantwortliche Zeit ist allerdings nur die des einzelnen Benutzers. Wechselt der Benutzer, bricht die Zuverlässigkeit ein, weil die Nutzungsdauer neu beginnt. Wenn Wissen über Fehlerumgehungsmöglichkeiten weitergegeben wird, z.B. über Foren oder FAQ-Seiten, lernt die gesamte Nutzergemeinschaft, so dass sich die Nutzungsdauern vieler Nutzer aufsummieren.

FAQ Web-Seiten mit Problembeschreibungen und Lösungen für die Problemumgehung.

2.91 Prosumenten

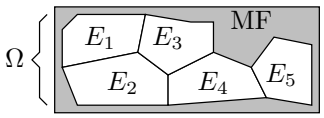
Prosument Nutzerbeteiligung an der Wertschöpfung.

Beta-Software sind Vorabversionen zur Einbeziehung der Nutzer in die Tests bis zur Anwendungsreife.

Bei Unterhaltungs-Software, Comuterspiele, neuartige Apps ist diese Konzept sogar zum Teil dahingehend erweitert, dass Nutzer nicht nur Testen, sonder auch bei der Anforderungsentwicklung mit helfen, z.B. neue Spiele-Ideen beitragen.

3.8 Eingabeprofile

2.92 Eingabe- und Nutzungsprofile



Ω Eingaberaum
E_i Äquivalenzklasse i
MF Fehlfunktion (unzulässige Eingaben)

Der **Eingaberaum** Ω eines IT-Systems umfasst viele Funktionen, die für unterschiedliche Aufgaben unterschiedlich häufig mit unterschiedlichen Daten und Datenbereichen genutzt werden.

Komplexe Systeme bieten sogar für viele Aufgaben mehrere Eingabe- und Lösungsmöglichkeiten (verg. Folie 1.104).

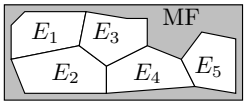
Äqivalenzklassen sind Eingabebereiche mit ähnlicher Verarbeitung und Nutzung.

Ein **Eingabeprofil** beschreibt die relative Nutzungshäufigkeit der Äqivalenzklassen und hat großen Einfluss auf die Nachweiswahrscheinlichkeiten und Fehlfunktionsraten vorhandener und modellierter Fehler.

Ein **Nutzungsprofil** ist das Eingabeprofil, wenn das System für einen seiner Bestimmungszwecke genutzt wird.

2.93 Messen von Nutzungsprofilen

Operationsprofil



mit $E_i = (c_i, f_i, h_i)$

Messprogramm

```
      if c1: f1; H1++;
[else] if c2: f2; H2++;
[else] if c3: f3; H3++;
...
[else ⟨MF-Behandlung⟩; H0++]
```

Eine Äquivalenzklasse hat eine Funktion f_i die unter Bedingung c_i mit Nutzungshäufigkeit h_i ausgeführt wird. Die Nutzungshäufigkeiten h_i lassen sich durch Instrumentierung von Ausführungszählern H_i messen:

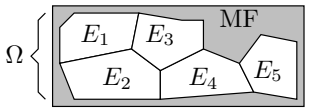
$$h_i = \frac{H_i}{N}$$

Das optionale »else« oben beschreibt gegenseitigen Ausschluss.

Für unzulässige Eingaben, also solche, die zu keiner genutzten Äquivalanzklasse gehören, ist das Sollverhalten Fehlfunktionsbehandlung.

Instrumentierung: Ergänzung von Programmen um Code, um das Verhalten zu untersuchen.

2.94 Erforderliche Nutzungsprofilabdeckung



Ω Eingaberaum
E_i Äquivalenzklasse i
MF Fehlfunktion (unzulässige Eingaben)

Die Zusicherung einer ausreichenden Zuverlässigkeit verlangt nach

(2.33)
$$R_\mathrm{F}(N) \overset{(\geq 1)}{=} \frac{N}{K \cdot \mu_\mathrm{F}(N)}$$

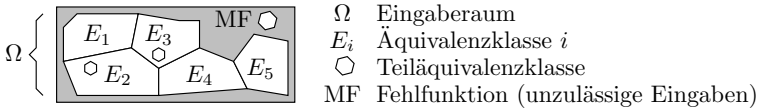
eine ausreichende Testanzahl N für alle potentiellen Nutzungsprofile. Erforderliche Testanzahl je Äquivalenzklasse:

$$N_i \geq h_{i.\,\max} \cdot N \tag{2.53}$$

E_i, h_i Äquivalenzklasse i, Auswahlhäufigkeit für Äquivalenzklasse i.
$\mu_\mathrm{F}(N)$ Zu erwartende Anzahl der Fehler, die nach N Tests nicht erkannt und beseitigt sind.

$R_F(N)$ Fehlerbezogene Teilzuverl. nach Beseitigung aller mit N Tests nachweisbaren Fehler.
K, N Formfaktor der Dichte der Fehlfunktionsrate, Testanzahl.
c_i, h_i Auswahlbedingung und Auswahlhäufigkeit für Äquivalenzklasse i.
$h_{i.\max}$ Maximale Nutzungshäufigkeit für alle potentiellen Operationsprofile.
N_i Ausreichende Testanzahl für Äquivalenzklasse i.

2.95 Testprofile



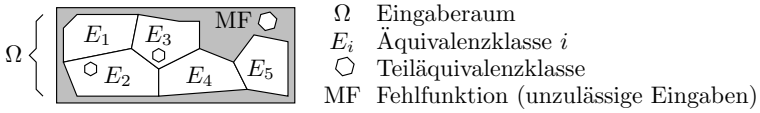
In einer Iteration aus Test und Beseitigung aller erkennbaren Fehler nimmt die Fehleranzahl mit der Testanzahl mit Exponent $0 < K < 1$ ab und die Fehlfunktionsraten der verbleibenden Fehler mit Exponent $K + 1$. Die mit zunehmender Testanzahl noch vorhandenen Fehler gehören zu immer kleineren Teiläquivalenzklassen (Abschn. 2.3.2).

Testprofile sind fehlerorientiert ausgewählte Eingabeprofile, die Teiläqivalenzklassen $E_{\mathrm{i.j}}$ mit Eingabebedingungen $c_{i.j}$, die schlecht nachweisbare Fehler extrem bevorzugen.

Idealerweise sind für alle potentiellen Fehler Testprofile zu suchen und wie die Nutzungsprofile mit genügend Tests abzudecken.

Eingabeprofil: Nutzungshäufigkeit der Äquivalenzklassen ähnlich verarbeiteter Eingaben.
Nutzungsprofil: Eingabeprofil bei der Nutzung.
.

2.96 Eingabebeschränkung und -wichtung

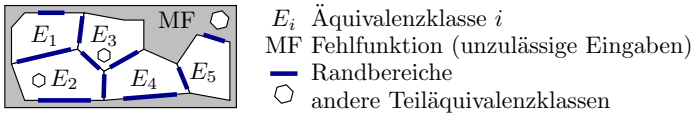


Angenommen, der Eingaberaum einer Äqivallenzklasse E_i umfasst 10 Zahlen. Teiläquivalenzklasse $E_{i.j}$ beschränkt den Auswahlbereich je Zahl auf ein Zehntel. Dann ist die Eingabemenge von $E_{i.j}$ um 10 Zehnerpotenzen kleiner und die Auswahlwahrscheinlichkeit je Wert ist um 10 Zehnerpotenenzen größer.

Für Fehler, die der Teiläqivalenzklasse $E_{\mathrm{i.j}}$ zugeordnet sind, erhöht sich proportional dazu die Nachweiswahrscheinlichkeit. Proportional zur Nachweiswahrscheinlichkeit erhöht sich nach (Gl. 2.39) die effektive Testanzahl, im Beispiel um den Faktor $C = 10^{10}$.

Die fehlerorientierte Bevorzugung von Eingabebereichen werden wir im Weiteren auch als Wichtung bezeichnet.

2.97 Randbereiche

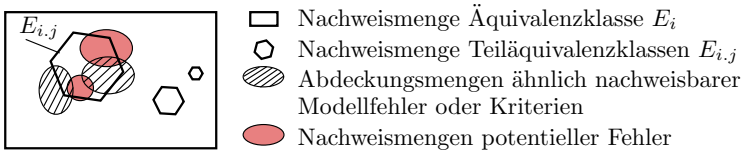


Eine Arte selten genutzter Eingabeteilbereiche mit zugeordneten schlecht nachweisbaren Fehlern sind Bereichsränder.

Bereichsränder sind bildlich gesehen $n-1$ dimensionale Flächen im n–dimensionalen Eingaberaum. Sie umfassen alle Eingabewerte mit geringem Abstand zu diesen Flächen. Anschaulich enthalten dünne Randflächen viel weniger Eingabewerte als die eingeschlossen Volumen. Umgekehrt proportional zur Bereichsverkleinerung erhöht sich die effektive Testanzahl der zugeordneten potentiellen Fehler.

Die Eingabebedingungen $c_{i.j}$ für Bereichsrandwerte leiten sich aus den Eingabebedingungen der umschließenden Äqivalenzklassen ab.

2.98 Andere Testabdeckungskriterien



Für andere Teiläqivalenzklassen mit schlecht nachweisbaren potentiellen Fehlern lassen sich nur Modellfehler oder Abdeckungkriterien angeben, die sich mit möglichen Fehlern Nachweisbedingungen für die Erreichbarkeit, Infektion und Ausbreitung teilen (Folie 2.26).

Bildlich gesehen werden für die straffierten bzw. secheckigen Flächen Eingabebedingen oder Tests gesucht, um zufällig die roten Flächen der unbekannten Nachweismengen zugeordneter Fehler zu treffen.

Zu Veranschaulichung sei die »Fläche« von $E_{\mathrm{i.j}}$ um 10 Zehnerpotenzen kleiner als die von E_i. Die Fehler werden angenommen von 1% der Eingaben aus $E_{i.j}$ nachgewiesen. Dann weist ein zufälliger Test aus E_i die Fehler mit Wahrscheinlichkeit 10^{-12} und einer aus $E_{i.j}$ mit 10^{-2} nach.

2.99 Modellrechnung für einen Idealfall

Mit den Nutzungs- und den unteschiedlichen Testprofilen sind andere Fehler gut nachweisbar. Im Idealfall erhöhen die N_i Tests für jedes Profil i die effektive Testanzahl gegenüber derselben kleinen Bezugstestanzahl N_0 um den Faktor ${}^{N_i}/_{N_0}$. Für unabhängige Nichterkennungsanteile $\left({}^{N_i}/_{N_0}\right)^{-K}$ je Äquivalenzklasse ist die Gesamtnichterkennung UND des Nichterkennungsanteils je Profil i (Folie 3.10):

$$1-\mu_{\mathrm{FC}}=\frac{\mu_{\mathrm{F}}}{\mu_{\mathrm{F}}(N_0)}=\prod_{i=1}^{\#OP}\left(\frac{N_i}{N_0}\right)^{-K}$$

Für übereinstimmende Testanzahl $N_i=N$ für alle Profile i:

$$\mu_{\mathrm{FC}}=1-\left(\frac{N}{N_0}\right)^{-\#OP\cdot K} \tag{2.54}$$

N_0, N	Bezugstestsanzahl, Anzahl der Tests je Operationsprofil incl. N_0.
K	Formfaktor der Dichte der Fehlfunktionsrate ($0<K<1$).
$\#OP$	Anzahl der unterschiedlichen Profile, mit denen getestet wird.
$\mu_{\mathrm{F}}(N_0)$	Zu erwartende Anzahl der nicht nachgewiesenen Fehler für die Bezugstestsanzahl N_0.
μ_{F}	Zu erwartende Anzahl der nicht nachgewiesenen Fehler nach allen Tests.
μ_{FC}	Zu erwartende Fehlerabdeckung.

2.100 Zuverlässigkeitsverbesserung

(2.54) $$\mu_{\mathrm{FC}}=1-\left(\frac{N}{N_0}\right)^{-\#OP\cdot K}$$

Für die Zuverlässigkeit in

(2.33) $$R_{\mathrm{F}}(N)\overset{(\geq 1)}{=}\frac{N}{K\cdot\mu_{\mathrm{F}}(N)}$$

ist $\mu_{\mathrm{F}}(N)=\mu_{\mathrm{F}}(N_0)\cdot(1-\mu_{\mathrm{FC}})$ und N die Testanzahl des Nutzungsprofils, also nicht die Gesamt- sondern nur die übereinstimmende Testanzahl N:

$$R_{\mathrm{F}}(\#OP\cdot N)=\frac{N}{K\cdot\mu_{\mathrm{F}}(N_0)}\cdot\left(\frac{N}{N_0}\right)^{\#OP\cdot K}\sim N^{1+\#OP\cdot K} \tag{2.55}$$

Unter der Voraussetzung, dass jedes Testprofil den Anteil der verbleibenden nicht nachweisbaren Fehler auch wieder um etwa $\left({}^{N}/_{N_0}\right)^{-K}$ verringert, nimmt die Zuverlässigkeit mit der Testanzahl je Profil mit Exponent $1+\#OP\cdot K$ zu.

N_0, N	Bezugstestsanzahl, Anzahl der Tests je Operationsprofil incl. N_0.
K	Formfaktor der Dichte der Fehlfunktionsrate ($0<K<1$).
$\#OP$	Anzahl der unterschiedlichen Profile, mit denen getestet wird.

2.101 Testprofilorientierte Testauswahl

Analyse der Testobjektbeschreibungen auf abzudeckende Kriterien:

- Equivalenzklassen, Equivalenzklassenränder,
- Modellfehler, Erreichbarkeitskriterien, ...

Bestimmung der erforlichen Abdeckungsanzahl je Kriterium.

Instrumentierung (Software) oder Fehlersimulation (Hardware) mit Abdeckungszählern.

Die Testauswahl startet mit einem Nutzungsprofil oder ungewichteten Zufallseingaben und wiederholt, bis alle Kriterien ausreichend oft abgedeckt sind:

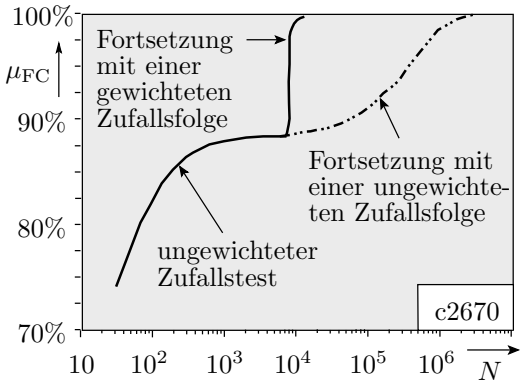
1. Zufallsauswahl entsprechend Profil, für eine vorgegebene Testanzahl oder bis sich die Abdeckung nicht mehr verbessert,
2. Suche eines neuen Testprofil, das noch unabgedeckte Kriterien extrem bevorzugt.

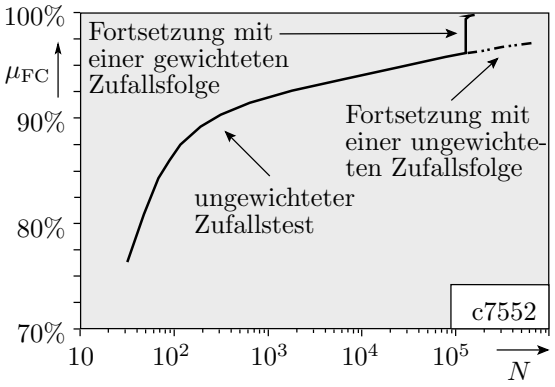
2.102 Anwendung auf digitale Schaltungen

Für digitale Schaltungen sind die Abdeckungskriterien typ. Haftfehler. Abschn. 6.3.4 zeigt eine Selbsttestlösung inc. experimentelle Evaluation mit zwei Profilen:

- ungewichtete (Zufallseingaben) und
- Eingaben mit fehlerorientierter Bitwichtung.

Bestätigung der These, dass der Test mit Testprofil den Anteil der nicht nachweisbaren Fehler noch einmal etwa um $(N/N_0)^{-K}$ und mit beiden Profilen zusammen um $(N/N_0)^{-2K}$ verringert.





Wichtung.

#

(2.55) $$R_F\left(\#OP \cdot N\right) = \frac{N}{K \cdot \mu_F(N_0)} \cdot \left(\frac{N}{N_0}\right)^{\#OP \cdot K} \sim N^{1+\#OP \cdot K}$$

Das Anwendungsbeispiel für den Selbsttest digitaler Schaltungen mit zwei Profilen in Abschn. 6.3.4 bestätigt dieThese, dass der Test mit Testprofil den Anteil der nicht nachweisbaren Fehler noch einmal etwa um $(N/N_0)^{-K}$ und mit beiden Profilen zusammen um $(N/N_0)^{-2K}$ verringern und damit (Gl. 2.55). Andere Experimente hierzu nicht bekannt.

2.103 Software-Test

Für Software lassen sich ein große Menge von im Code instrumentierbarere Erreichbarkeits-, Infektions und mit Einschränkungen auch Weiterleitungskriterien formulieren, für die jeweils eine große Anzahl zufälliger Tests aus deren Eingabemengen zu suchen sind (Abschn. 7.4). Äquivalenzklassenabdeckung eingeschlossen.

> Profilgesteuerte Zufallstestauswahl, solange sich damit die Kriterienabdeckung erhöhen lässt und dann Suche eines neuen Profils, dass bisher unabgedeckte Kriterien extrem bevorzugt«

scheint machbar, aber nur in Kombination mit Lösungen für:

- Automatisierung (händisch unbezahlbar),
- Testausgabekontrolle, ...

Abschn. 7.5 skizziert eine hypothetische Infrastruktur von Werkzeugen für die Testautomatisierung für Software und Teilabschnitt 7.5.3 mögliche Ansätze für die Test- und Profilsuche.

Zusammenfassung

2.104 Vor- und Zuverlässigkeitstest

In den Entstehungs- und Fehlerbeseitigungsprozesse entehen insgesamt im Mittel μ_{FCR} Fehler. Davon erkennen die Vortests

- statisch Tests (Reviews, Syntax, ...)
- dynamischen Grobtests, ob überhaupt etwas funktioniert, ...

mit N_0 enthaltenen dynamischen Tests einen Anteil FC_{PT}, der beseitigt beseitigt wird. Verbleibende Fehleranzahl und Fehlfunktionsrate:

(2.27) $$\mu_{\mathrm{F}}(N_0) = \mu_{\mathrm{FCR}} \cdot (1 - FC_{\mathrm{PT}})$$

(2.28) $$\zeta_{\mathrm{F}}(N_0) \overset{(\leq 1)}{=} \frac{K \cdot \mu_{\mathrm{F}}(N_0)}{N_0}$$

Die weitere Fe
erwartende Fe
K+1 ab.

(2.17) $$\mu_{\mathrm{FC}} = 1 - \frac{\mu_{\mathrm{F}}(N_2)}{\mu_{\mathrm{F}}(N_1)} = 1 - \left(\frac{N_2}{N_1}\right)^{-K} \quad \text{mit} \quad 0 < K < 1$$

(2.30) $$\zeta_{\mathrm{F}}(N) \overset{(\leq 1)}{=} \zeta_{\mathrm{F}}(N_0) \cdot \left(\frac{N}{N_0}\right)^{-(K+1)}$$

Effektive Testanzahl: (äquivalente) Testanzahl, für die alle nachweisbaren Fehler beseitigt werden.

2.105 Formfaktor, Zuverlässigkeit und Sicherheit

Der Formfaktor der Verteilung der Fehlfunktionsrate kann sowohl aus der Abnahme der Fehleranzahl als auch aus der Abnahme der Fehlfunktionsrate für eine Vergrößerung der effektiven Testsanzahl von N_1 auf $N_2 \gg N_1$ Tests abgeschätzt werden:

(2.18) $$K = -\log\left(\frac{\mu_{\mathrm{F}}(N_2)}{\mu_{\mathrm{F}}(N_1)}\right) / \log\left(\frac{N_2}{N_1}\right)$$

(2.26) $$K = \log\left(\frac{\zeta_{\mathrm{F}}(N_1)}{\zeta_{\mathrm{F}}(N_2)}\right) / \log\left(\frac{N_2}{N_1}\right) - 1$$

Wenn durch geeignete Systemgestaltung Fehlfunktionen durch Störungen vernachlässigbar sind und bei erkannten Problemen ein sicherer Zustand hergestellt wird, nehmen Zuverlässigkeit und Sicherheit mit der $K+1$-ten Potenz der effektiven Testanzahl zu:

(2.37) $$\frac{R_{\mathrm{MT}}(N_2)}{R_{\mathrm{MT}}(N_1)} = \frac{S(N_2)}{S(N_1)} = \left(\frac{N_2}{N_1}\right)^{K+1}$$

2.106 Effektive und tatsächliche Testanzahl

(2.39) $$N = C \cdot N_{\mathrm{T}}$$

- Für modulinteren Fehler ist die effektive Testanzahl der Modultests viel größer als die der Tests in den Systemumgebung:

  (2.41) $$N = C \cdot N_{\mathrm{M}} \quad \text{mit} \quad C = \frac{1}{p_{\mathrm{FS}} \cdot p_{\mathrm{FO}}} \gg 1$$

  Deshalb werden Module vor Einbau gründlich getestet.

- Fehlermodellspezifische Skalierung. Zu erwartende Fehlerabdeckung etwa Modellfehlerabdeckung der C-fachen Testanzahl:

  (2.42) $$N = C \cdot N_{\mathrm{MF}} \quad \text{für } \mu_{\mathrm{FC}}(N) = \mu_{\mathrm{FCM}}(N_{\mathrm{MF}})$$

  - tendentiell besser nachweisbare Modellfehler: $C > 1$
  - tendendtiell schlechter nachweisbare Modellfehler: $C < 1$.
  - Für Haftfehler wird später der Richtwert $C \approx 0{,}5 \ldots 1$ abgeschätzt.

- Für Reifeprozesse gilt
  $$C = p_{\mathrm{FE}} \ll 1$$
  aber als Testanzahl akkumulieren sich die genutzten Service-Leistungen vieler Nutzer über ein lange Reifedauer.

Effektive Testanzahl: (äquivalente) Testanzahl, für die alle nachweisbaren Fehler beseitigt werden.

2.107 Reifeprozess

Fortsetzung der Fehlerbeseitigungsiteration in der Einsatzphase mit den Nutzern als Tester.

- Zunahme der effektive Testanzahl mit der Reifedauer:

  (2.43) $$N = p_{\mathrm{FE}} \cdot \mu_{\mathrm{NU}} \cdot \eta_{\mathrm{SU}} \cdot (t_{\mathrm{M}} + t_{\mathrm{V0}}) \quad \text{mit} \quad t_{\mathrm{V0}} = \frac{N_{\mathrm{V0}}}{p_{\mathrm{FE}} \cdot \mu_{\mathrm{NU}} \cdot \eta_{\mathrm{SU}}}$$

- Zunahme der effektive Testanzahl mit der Versionsnummer:

  (2.44) $$N = \underbrace{p_{\mathrm{FE}} \cdot \mu_{\mathrm{NU}} \cdot \eta_{\mathrm{SU}} \cdot t_{\mathrm{VR}}}_{N_{\mathrm{MV}}} \cdot (u + u_{\mathrm{V0}}) \quad \text{mit} \quad u_{\mathrm{V0}} = \frac{N_{\mathrm{V0}}}{N_{\mathrm{VM}}}$$

- Abnahme der Fehleranzahl mit Exponent K:

  (2.45) $$\mu_{\mathrm{F}}(t_{\mathrm{M}}) = \mu_{\mathrm{F}}(t_{\mathrm{M0}}) \cdot \left(\frac{t_{\mathrm{M}} + t_{\mathrm{V0}}}{t_{\mathrm{M0}} + t_{\mathrm{V0}}}\right)^{-K}$$

  (2.46) $$\mu_{\mathrm{F}}(u) = \mu_{\mathrm{F}}(v) \cdot \left(\frac{u + u_{\mathrm{V0}}}{v + u_{\mathrm{V0}}}\right)^{-K}$$

- Abnahme der Fehlfunktionsrate durch Fehler mit Exponent $K + 1$.

2.108 Zuverlässigkeit und Sicherheit

Wenn Fehlfunktionen durch Störungen vernachlässigbar sind und bei allen erkannten Problemen ein sicherer Zustand hergestellt wird, ist nehmen Zuverlässigkeit und Sicherheit mit der $K + 1$-ten Potenz der Reifedauer bzw. der Versionsanzahl zu:

(2.51) $$\frac{S(t_{\mathrm{M}})}{S(t_{\mathrm{M0}})} = \frac{R_{\mathrm{MT}}(t_{\mathrm{M}})}{R_{\mathrm{MT}}(t_{\mathrm{M0}})} = \left(\frac{t_{\mathrm{M}} + t_{\mathrm{V0}}}{t_{\mathrm{M0}} + t_{\mathrm{V0}}}\right)^{K+1}$$

(2.52) $$\frac{S(u)}{S(v)} = \frac{R_{\mathrm{MT}}(u)}{R_{\mathrm{MT}}(v)} = \left(\frac{u + u_{\mathrm{V0}}}{v + u_{\mathrm{V0}}}\right)^{K+1}$$

- Lange Reifeprozesse über Jahre und Jahrzehnte erzielen auf andere Weise unerreichbare Zuverlässigkeiten und Sicherheiten.

- Alte, lange gereifte Software ist schwer zu ersetzen, weil gleichwertiger Ersatz zuerst lange bei vielen Nutzern reifen muss.

Es gibt auch einen Reifeprozesse für das Nutzerverhalten. Dieser bewirkt, dass Zuverlässigkeit und Sicherheit eines Systems auch mit der individuellen Nutzungsdauer eines Nutzers zunehmen.

2.109 Eingabe- und Nutzungsprofile

Eingabeprofile beschreiben die relative Nutzungshäufigkeit der Äquivalenzklassen, d.h. der Eingabemengen mit vergleichbarer Verarbeitung. Hinreichende Zuverlässigkeit verlangt eine ausreichend große Testanzahl für jede Äquivalenzklasse entsprechend Nutzungsprofil.

Testprofile sind fehlerorientiert ausgewählte Eingabeprofile, die schlecht nachweisbare Fehler extrem bevorzugen und sind gleichfalls mit genügend Tests abzudecken.

Die Testauswahl startet z.B. mit dem Nutzungsprofil und wiederholt, bis alle Kriterien ausreichend oft abgedeckt sind:

1. Zufallsauswahl vieler Tests entsprechend Profil
2. Suche eines neuen Testprofil für noch unabgedeckte Kriterien.

Für digitale Schaltungen funktioniert eine Kombination ungewichteter mit fehlerorientierter gewichteten Zufallstests gut.

Für Software theoretisch ein erfolgsversprechender Ansatz zur Erzielung hoher Zuverlässigkeit, aber dafür sind noch andere Dinge zu lösen.

4 Fehlervermeidung

2.110 Fehlervermeidung

MF-Behandlung	Fehlerbeseitigung	Fehlervermeidung
Überwachung, robuste Reaktion auf erkannte Probleme	Test und Beseitigung erkannter Fehler	Problembeseitigung in Entstehungsprozessen

Fehlervermeidung bedeutet Minderung der Fehlerentstehungsraten durch Problembeseitigung in den Entstehungsprozessen.

Entstehungsprozesse sind wie IT-Systeme Service-Leister mit Fehlfunktionsraten, Kontrollen und Problembehandlung. Unbehandelte Fehlfunktionen verursachen Produktfehler.

Beseitigung von Entstehungsproblemen wieder auf allen drei Ebenen:

- Prozessüberwachung, Gegenmaßnahmen für erkannte Probleme.
- Test und Fehlerbeseitigung vor und während der Prozessnutzung.
- Fehlervermeidung während der Entstehung der Prozesse.

Wir konzentrieren uns auf das Reifen der Prozesse, d.h. die Problembeseitigung während der Prozessnutzung.

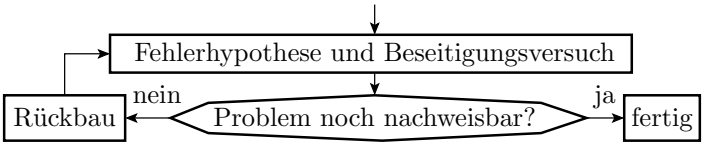
2.111 Prozessfähigkeiten

Entstehungsprozesse werden durch ihre Fähigkeiten (statt Funktionen) charakterisiert. Fähigkeit sind die Möglichkeiten, was geschaffen werden kann, wie gut, wie genau, wie billig, ... Ein Teil der Fähigkeiten sind die zur Fehlervermeidung.

Der wissenschaftlich-technische Fortschritt lässt sich als Iteration aus Schaffung, Nutzung und Reifen von Prozessfähigkeiten beschreiben:

- Schaffung: (Weiter-) Entwicklung von Werkzeuge, Verfahren, Kontrollen (incl. Messverfahren), Programmiersprachen, Theorien, Modellen, ... getrennt von der Prozessnutzung.
- Nutzung: Neue Fähigkeiten »kauft man. Für die Fehlervermeidung bedeuten neue Prozessfähigkeiten nicht nur Fortschritt, sondern auch neue Probleme, die in einem Prozess »Lernen aus Fehlern« umgangen oder beseitigt werden.
- Die Erfahrungen bei der Prozessnutzung fließen in die Entwicklung verbesserter und neuer Fähigkeit ein.

2.112 Fähigkeiten zum Lernen aus Fehlern



Lernen aus Fehlern verlangt häufige Wiederholung gleicher oder ähnlicher Abläufe, Kontrollen und Problembeseitigungsiterationen nach dem Prinzip der experimentellen Reparatur:

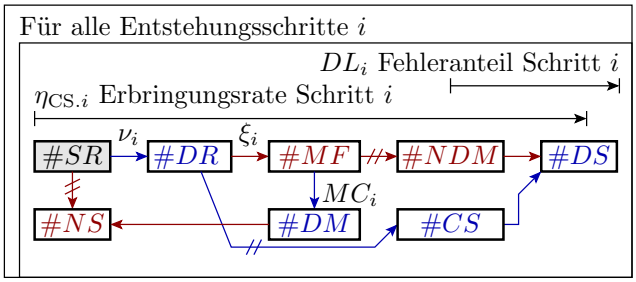
- Aufstellen von Hypothesen über die Problemursache.
- Überprüfung der Hypothesen durch Beseitigungsversuche.
- Wenn Problem nicht beseitigt, Rückbau und nächster Versuch.

Wiederholbare Abläufe, Kontrollen, Problembeseitigungs- und Rückbaumöglichkeiten, ... verlangt Prozessfähigkeiten.

Lernen aus Fehlern selbst ist ein extrem arbeitsintensiver stochastischer Prozess, angetrieben, weil es sich ökonomisch lohnt.

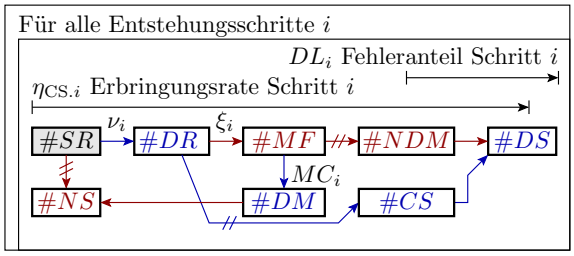
4.1 Fehlerentstehung

2.113 Fehler als MF der Entstehungsprozesse



SR	Schritt-Anforderung	MF	Fehlfunktion (Prozessfehler)
NS	verweigerte Leistung	DM	erkannter Prozessfehler
DR	erbrachtes Ergebnis	NDM	nicht erkannter Prozessfehler
CS	korrekte Leistung	DS	erbrachte Leistung
ν_i	Erfolgsrate	MC_i	Prozessfehlfunktionsabdeckung
ξ_i	Fehlerentstehungsrate	Aussprache: ν: ny, ξ: xi	

Ein Entstehungsprozess besteht aus vielen Schritten, in denen Leistungen erbracht und Fehler entstehen, erkannt und beseitigt werden.



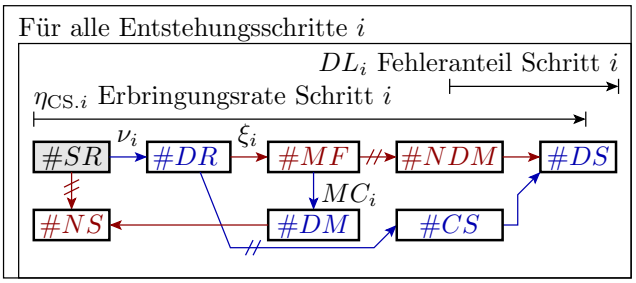
Wenn alle Leistungen mit erkannten Problemen aussortiert werden, hat jeder Schritt eine Erbringungsrate und einen Fehleranteil:

$$\eta_{\mathrm{CS}.i} = \left.\frac{\#DS}{\#SR}\right|_{\mathrm{ACR}} = \nu_i \cdot ((1-\xi_i) + \xi_i \cdot (1-MC_i)) = \nu_i \cdot (1 - MC_i \cdot \xi_i)$$

$$DL_{\mathrm{i}} = \left.\frac{\#NDM}{\#DS}\right|_{\mathrm{ACR}} = \frac{\nu_i \cdot \xi_i \cdot (1-MC_i)}{\nu_i \cdot (1-MC_i \cdot \xi_i)} = \frac{\xi_i \cdot (1-MC_i)}{1-MC_i \cdot \xi_i}$$

- Ein Produkt entsteht, wenn alle Schritte erbracht werden.
- Ein fehlerhaftes Produkt entsteht, wenn in mindestens einem Schritt ein Fehler entsteht.

Ein Entstehungsprozess aus vielen Schritten lässt sich wie ein IT-System aus vielen Bausteinen mit Kontrollen und Problembeseitigungsprozessen modellieren. Wichtig sind wieder die Kontrollen.



Beispiele folgen in (Abschn. 3.4) nach einer themenspezifischen Einführung in das Rechnen mit Wahrscheinlichkeiten.

2.116 Fehlerentstehungsraten und Metriken

Statt aus den Fehlerentstehungs, -erkennung und -korrekturraten der einzelnen Entstehungsschritte wird die zu erwartende Fehleranzahl in der Regel über Metriken für die Produkt- oder Prozessgröße geschätzt:

$$\mu_{\mathrm{CF}} = \xi_{<\mathrm{C}>} \cdot M_{\mathrm{C}} \tag{2.56}$$

Als Metriken werden gut bestimmbare Kenngrößen verwendet, z.B.:

- Entwurfsaufwand in Arbeitsstunden,
- Entwurfsumfang in NLOC, Transistoren, Dokumentationsseiten, ...
- Fertigungsaufwand in Arbeitsschritten,
- ...

Die Entstehungsraten ergeben sich umgekehrt aus dem Verhältnis experimentell abgeschätzter Erwartungswerte zum Bezugswert:

$$\xi_{<\mathrm{C}>} = \frac{\mu_{\mathrm{CF}}(M_{\mathrm{C}})}{M_{\mathrm{C}}} \tag{2.57}$$

$\xi_{<\mathrm{C}>}$ Fehlerentstehungsrate in Fehlern je Bezugsgröße der Metrik M_{C}.
μ_{CF} Zu erwartende Anzahl der Fehler aus den Entstehungsprozessen.
M_{C} Metrik für den Entstehungsaufwand oder die Größe des Produkts.
Aussprache: μ: my, ξ: xi.

Beispiel 2.4 Programmfehler

$\xi_{\mathrm{NLOC}} = 30$ Fehler / 1000 NLOC, Programm mit $C = 2000\,\mathrm{NLOC}$.

Wie groß ist die zu erwartende Anzahl der Programmierfehler vor Test und Fehlerbeseitigung?

$$\mu_{\mathrm{CF}} = \xi_{\mathrm{NLOC}} \cdot M_{\mathrm{C}} = \frac{30\,\mathrm{Fehler} \cdot 2000\,\mathrm{NLOC}}{1000\,\mathrm{NLOC}} = 60\,\mathrm{Fehler}$$

Beispiel 2.5 Schaltkreisfehler

$\xi_{\#\mathrm{Tr}} = 1$ Fehler je 10^6 Transistoren. Schaltkreis mit $M_{\mathrm{C}} = 10^5$ Transistoren.

Wie groß ist die zu erwartende Anzahl der Fehler je Schaltkreis vor dem Aussortieren der erkennbar defekten Schaltkreise?

$$\mu_{\mathrm{CF}} = \xi_{\#\mathrm{Tr}} \cdot M_{\mathrm{C}} = \frac{1\,\mathrm{Fehler} \cdot 10^5\,\mathrm{Transistoren}}{10^6\,\mathrm{Transistoren}} = 0{,}1\mathrm{Fehler}$$

NLOC Netto Lines of Code, Anzahl der Code-Zeilen ohne Kommentar und Leerzeilen.

Es gibt auch empirische Modelle, die eine Zunahme der Fehlerentstehungsrate mit der Systemgröße postulieren. Für Software-Module wird z.B. unterstellt, dass die Fehleranzahl je NLOC ab etwa 3 Quellcode-Seiten je Funktionsbaustein überproportional zunimmt, weil die Entwerfer die Übersicht verlieren. Bekanntes und damit beseitigbares Problem.

Unserer Fehlerkultur »verbietet« bekannte Probleme*:

Entstehungsprozesse sind so zu gestalten, dass

- offenkundige negative Einflüsse auf die Fehlerentstehungsrate, wie die Zunahme der Fehlerentstehungsrate mit dem Entstehungsaufwand, durch die Prozessgestaltung vermieden werden.
- Das rechtfertigt die Annahme, dass die Fehleranzahl bei akzeptabler Prozessgestaltung nur proportional mit der Systemgröße und dem Entstehungsaufwand zunimmt.

* Unsere Idealisierung der Fehlerkultur dient der Modellvereinfachung.

Aussprache: μ: my, ξ: xi.

Die Brauchbarkeit von Metriken hängt davon ab, wie gut von Werten der Metrik auf die zu verbessernden Zielgrößen

- Problementstehungsraten
- Problemerkennungsraten und
- Problemvermeidungsraten

geschlussfolgert werden kann.

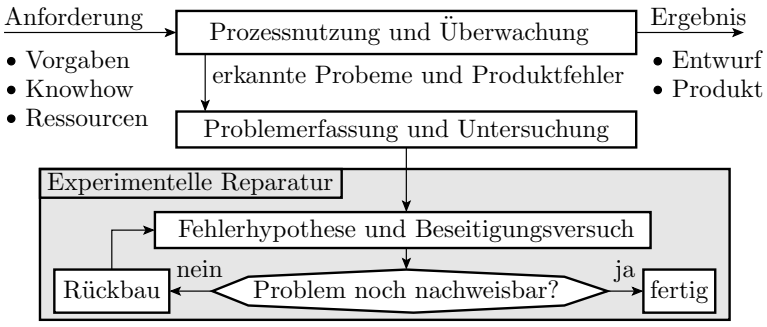
Problembeseitigungsprozesse bewirkt nur solange Verbesserungen, wie die Erfolgskontrolle der experimentellen Reparatur mehrheitlich richtig entscheidet. Danach halten sich Verbesserungen und Verschlimmbesserungen(IW) die Wage.

Die wichtigste Prozessfähigkeit für das Lernen aus Fehlern sind brauchbare Metriken für die Prozessgüte.

(IW) Verschlechterungen, die die Kontrolle als Verbesserungen ausweist.

4.2 Reifen von Prozessen

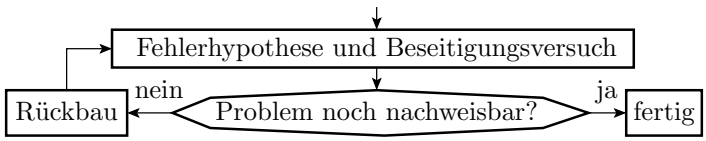
2.120 Reifen von Entstehungsprozessen



Nach Einführung neuer / verbesserter Fähigkeiten müssen die Prozessnutzer erst einmal lernen, diese zu nutzen:

- Ausprobieren der neune Fähigkeiten durch Prozessnutzung.
- Beseitigung erkannter Probleme, insbesondere Produkfehlerursachen durch experimentelle Reparatur.

2.121 Determinismus und Erfolgskontrolle



Für IT-Systeme gibt es ideale Erfolgskontrollen (siehe später):

- determistisches Verhalten, für gleiche Eingaben entstehen ohne oder mit gleichen Fehlern gleiche Ausgaben,
- »getestete« Tests mit Soll/Ist-Vergleich,
- eindeutige ja/nein-Aussabe über Problembeseitigung,
- Beseitigung aller nachweisbaren Probleme.

Entstehungsprozesse sind oft nicht deterministisch. Keine perfekte Erfolgskontrolle. Im ungünstigen Fall erfordert die Kontrolle auf Verbesserungen/Verschlechterungen viele Prozessdurchläufe und erlaubt nur eine unsichere Aussage.

2.122 Reifepotential und -geschwindigkeit

Für den typischen Entstehungsprozess verlangt die Erfolgskontrolle nach jedem Problembeseitigungsversuch

- viele Prozesswiederholungen zur Schätzung von Metriken,
- deren Vergrößerung/Verkleinerung Wahrscheinlichkeitsaussagen über die Verbesserung/Verschlechterung erlaubt.

Vergleich der Möglichkeiten zu Reifen mit denen von IT-Systemen:

- dauert viel länger,
- Problemneuentstehung durch die ständigen Verbesserungsversuche und falschen Rückbau.

Die Reifegeschwindigkeit und das erzielbares Minimum der Fehlerentstehungsrate wird von der Fähigkeit bestimmt, anhand der nachfolgender Prozessdurchläufe den Verbesserungserfolg richtig zu bewerten. Wichtiger Teilaspekt, Wiederholbarkeit der Situation, in der das Problem sichtbar geworden ist. Wiederholbarkeit wird in der Regel daran bewertet, wie gut Prozessergebnisse insgesamt vorhersehbar sind.

4.3 Zentrierung, Verbesserung

2.123 Prozesszentrierung

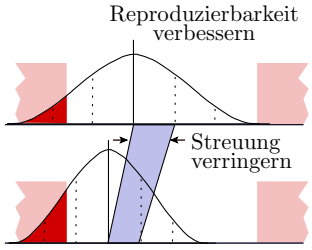
Einfaches Beispiel für die komplexen Zusammenhänge* zwischen

- Fähigkeiten und
- Fehlerentstehungsrate

ist ein mechanischer Fertigungsschritt mit einem streuenden Parameter z.B. dem Durchmesser einer Bohrung.

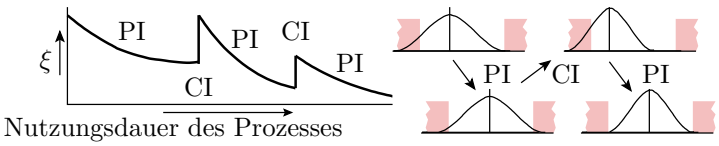
Die Fehlerentstehungsrate ist hier die Wahrscheinlichkeit, dass der Parameterwert nicht im Toleranzbereich liegt:

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



* Arbeitsaufwändiger Prozess, bei dem nach dem Prinzip der experimentellen Reparatur viele Prozesseinstellungen durchprobiert werden (Ameisarbeit).

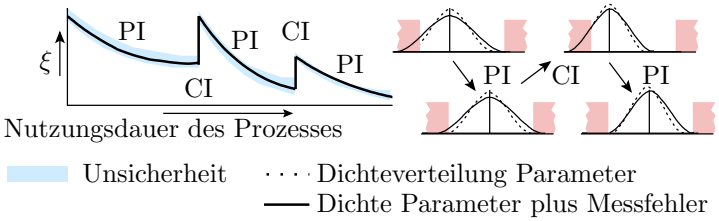
2.124 Sägzahnverlauf der Fehlerentstehungsrate



- Die Fähigkeitsverbesserung schafft die Möglichkeit, Toleranzen besser einzuhalten, aber bei Neuerungen geht die Zentrierung verloren. Sprunghafte Zunahme der Fehlerenstehungsrate.
- Während der Prozessnutzung Nachjustierung an den Einstellmöglichkeiten zur Verschiebung des Erwartungswerts in die Mitte des Toleranzfensters. Abnahme der Fehlerenstehungsrate.

CI, PI Fähigkeitsverbesserung, Prozessverbesserung.

2.125 Schätz- und Messgenauigkeiten

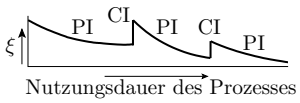


Mess- und Schätzungenauigkeiten haben in einem Fehlerbeseitigungsprozess Irrtümer zu Folge:

- Rückbau von Verbesserungen und
- und kein Rückbau von Verschlechterungen.

Je geringer die Schätz- und Messfehler, desto größer die Fähigkeit des Prozesses zur Fehlervermeidung.

CI, PI Fähigkeitsverbesserung, Prozessverbesserung.



- Fähigkeitsverbesserung in größeren Zeitschritten und
- Reifen durch »Lernen aus Fehlern« in kleinen Schritten

ist typisch für alle technologischen Prozesse incl. für Entwurfs- und Fertigungsprozesse von IT-Systeme.

Das Modell der alternierenden Abfolge von Fähigkeitsverbesserung und Reifen liefert auch zahlreiche nützliche weitere Einblicke:

- Ausreichend Reifezeit zwischen Fähigkeitverbesserungen,
- Nur was kontrollierbar ist, lässt sich zielgerichtet verbessern,
- Falsche Zielgrößen führen zur »Verschlimmbesserung«,
- Am qualitativ hochwertigsten sind oft die Produkte, die kurz vor einer technologischen Neuerung gefertigt wurden, ...

Reifeprozesse und Vorstufe Reifefähigkeit sind auch interessant für:

- andere Fertigungs- und Entwurfsprozesse,
- Organistionsabläufe in Institutionen, Management,
- Lernprozesse in Bildungseinrichtung, ...

2.127 Die Fähigkeit zum Reifen

Das CMMI (Capability Maturity Model Integration) definiert u.a. fünf Fähigkeitsstufen* zur Klassifikation von Prozessen, Organisationen, ...:

1. Wiederholte Abläufe, undokumentiert. Ermöglicht individuelles Lernen aus Fehlern.
2. Dokumentation der Abläufe und beobachteten Probleme. Ermöglicht personenübergreifendes Lernen aus Fehlern.
3. Verwaltung und Steuerung der Abläufe. Ermöglicht Fokusierung auf das sichere Erreichen angestrebter Ziele (Dauer, Qualität, ...).
4. Quantitatives Management: Definition und Erfassung von Leistungskennzahlen. Ermöglicht Beobachtung des Reifeverhaltens.
5. Kontinuierliche Prozessverbesserung durch quantitatives Feedback aus dem Prozess, d.h. gezielter Reifeprozess**.

Je höher der Reifegrad, um so größer die Fähigkeit zu reifen und desto geringer die erzielbaren Fehlerentstehungsrate der Prozesse.

* Abstufung als Vorstufe einer Metrik charakterisiert den aktuellen Entwicklungsstand.

** Fähigkeit zum Reifen sind nicht selbstverständlich.

.

4.4 Vorgehensmodelle

2.128 Der Technologiegedanke und Projekte

Reproduzierbare Entstehungsabläufe werden auch als Technologie bezeichnet*. Technologien reifen dadurch, dass ähnliche Abläufe oft wiederholt, dabei überwacht und erkannte Probleme beseitigt werden.

Wie verhält es sich mit Projekten:

- Manuelle kreative Teile der Entwurfsprozesse 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 häufige Wiederholung ähnlicher Abläufe, um aus erkannten Fehlern lernen zu können.

Schließt das Projekte von der Fehlervermeidung aus?

* Der Begriff *Technologie* wurde erstmals vom Göttinger Professor Johann Beckmann (1739-1811) im Lehrbuch "Grundsätze der teutschen Landwirthschaft" verwendet.

2.129 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:

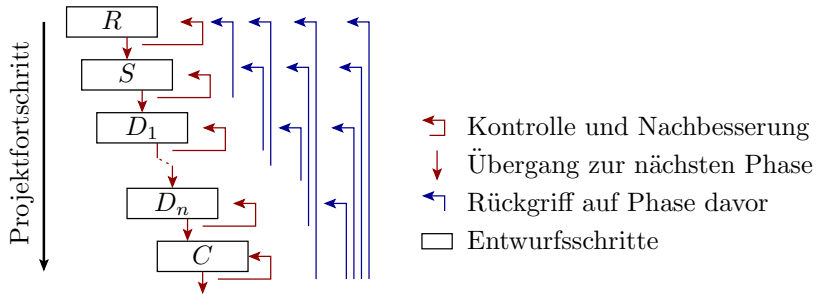
- Aufteilung in Schritte und Phasen,
- Referenzabläufe,
- Definition von Zwischen- und Endkontrollen, ...

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

- Anforderungsanalyse, Spezifikation der Ziele,
- Lösungsfindung in mehreren Stufen,
- Codierung ist erst die letzte Stufe.

Fehlervermeidung bei Projektarbeit ist die kontinuierliche empirische Verbesserung, d.h. das Reifen des Vorgehens- [modells].

2.130 Stufenmodelle



Stellgroßen zur Prozessverbesserung:

- Arbeitsorganisation der Phasen,
- geforderte Tests, Dokumentation, ... bei Phasenübergängen,
- Genehmigungsverfahren für Rückgriffe über mehrere Stufen*, ...

R, S Anforderungsanalyse, Spezifikation.

D_i, C Schritt *i* des Architektur- und Funktionsentwurfs, Codierung.

* 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.

2.131 Bewertung von Vorgehensmodellen

Reifen als komplexer arbeitsintensiver stochastische Prozess schafft nur Verbesserungen, wenn diese überprüfbar sind.

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:

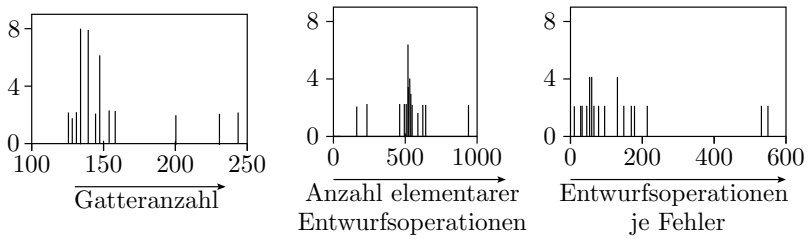
- Dauer, Kosten bezogen auf die Projektgröße,
- Arbeitsschritte je entstehender Fehler, ...

Erwartungswerte, Streuungen, Skalierbarkeit auf Projektgröße, ...

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

2.132 Ein Experiment [1]

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, ...



2.133 Rückschlüsse aus dem 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 bei signifikannten Unterschieden schlussfolgern, an welcher Hochschule Studierende für diese Aufgabe besser ausgebildet werden.

4.5 Qualität und Kreativität

2.134 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.

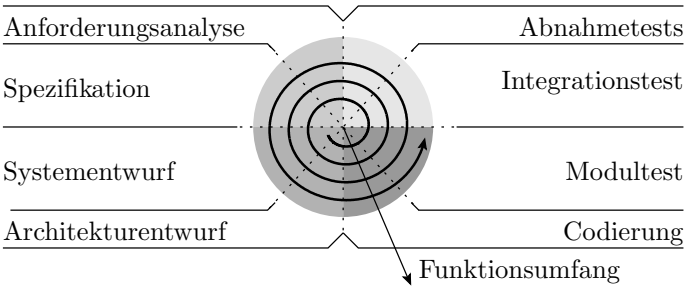
Daraus resultierende Fragestellung

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

2.135 Evolutionäre Vorgehensmodelle

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:

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



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 im entstandenen Code nur noch Fehler beseitigt werden.

Neue Features, Ideen und Werkzeuge können aber nachträglich grundlegende Änderungen an existierenden Systemteilen, Architekturentscheidungen, Modularisierung, ... ratsam erscheinen lassen.

Grundidee gut, tatsächlicher Nutzen steckt in den Umsetzungsdetails.

2.137 Ein Abstecher zu Lernprozessen

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 MF und Zeit.

Da steckt über Jahrhunderte gereiftes Knowhow drin.

Stufen der Wissensvermittlung an Hochschulen:

1. Wissenvermittlung: anlesen, erklären, ...

   Vorlesung, Seminare, Selbststudium, ...

2. Training, bis Ergebnisse vorhersagbar.

   Übung, Klausurvorbereitung*, Praktika.

3. Professionalisierung: Prozessüberwachung; Beseitigung von Schwachstellen und Problemen in den Abläufen.

   Aus Zeitgründen erst in der Berufspraxis für den eigenen eingeschränkten Tätigkeitsbereich möglich.

* Auch Bewertung in Arbeitsmenge pro Klausurdauer und Fehlern pro Arbeitsmenge.

2.138 Querverbindung Drittmittelprojekte

- Die Professionalisierungsphase durchlaufen erst die Absolventen in der Praxis.
- Akademiker und Studenten sind noch 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, Demonstratoren, ... aber im IT-Bereich ungeeignet für die Einbindung in die Produktentwicklung.

Demonstrator: Vereinfachte Impementierung zur Untersuchung und Demonstration der Machbarkeit.

2.139 Qualitätssicherung an unser Hochschule

Die Master-Bachelor-Einführung (Bolonia-Prozess) strebt nach Referenzabläufen, vergleichbare Abschlüsse, ...

Das ist eine Etablierung grundlegender Prozessfähigkeiten:

- große Wiederholanzahl vergleichbarer Abläufe,
- Prozesseinstellungen zum Durchprobieren,
- Kenngrößenerhebung für die Erfolgskontrolle.

um ein Reifen der Ausbildungqualität zu ermöglichen.

Wie ist das an unserer Uni:

- Welche Prozessüberwachungen gibt es?
- Wo sind Vorgehensmodelle erkennbar?
- Was für Ressourcen bindet der angestoßene Reifeprozess?
- Wie wird verhindert, dass die Kreativität nicht darunter leidet?

Fehlervermeidung eröffnet interessante Blickwinkel auf Technologien, Institutionen, Behörden bis hin zu unserer gesamten wissenschaftlich-technische Weiterentwicklung.

Zusammenfassung

2.140 Fehlerentstehung

- Fehler entstehen in den Entwurf-, Fertigungs- und Reparaturprozessen mit den Produkten.

- Entstehungsprozesse sind wie IT-Systeme als Service-Leister modellierbar mit Erbringungsraten (⇒ Ausbeute), Fehlfunktionsraten (⇒ Fehlerenstehungsrate), ...
- In der Praxis werden die Fehlerentstehungsraten auf Zählwerte von Metriken bezogen:

(2.56) $$\mu_{\mathrm{CF}} = \xi_{<\mathrm{C}>} \cdot M_{\mathrm{C}}$$

(2.57) $$\xi_{<\mathrm{C}>} = \frac{\mu_{\mathrm{CF}}(M_{\mathrm{C}})}{M_{\mathrm{C}}}$$

2.141 Fehlervermeidung, Determinismus

Fehlervermeidung ist Fehlerbeseitigung in den Entstehungsprozessen. Fokus Reifen der Prozesse, d.h. Fehlerbeseitigung in der Nutzungsphase mit seinen Bestandteilen:

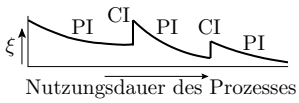
- Prozessnutzung und Überwachung,
- Problemerfassung und Untersuchung,
- Problembeseitigung durch experimentelle Reparatur als Iteration aus Beseitigungsversuchen und Erfolgskontrolle.

Das erfordert Fähigkeiten:

- große Wiederholrate, große Problemerkennungsrate,
- Stellschrauben zur Prozessnachbesserung,
- Kontrollmöglichkeiten für den Beseitigungserfolg, ...

Problematisch oft fehlender Determinismus, insbesondere bei kreativen Arbeiten. Kein Determinismus erschwert Lernen aus Fehlern, dabei insbesondere die Erfolgskontrolle nach Problembeseitigungsversuchen.

2.142 Fähigkeitsverbesserung und Reifen



Fähigkeiten, auch die für eine geringere Fehlerentstehungsrate werden »Offline«, d.h. getrennt vom Prozess entwickelt und in größeren Zeitschritten übernommen. Dabei geht die »Zentrierung« verloren, d.h. es kommen neue Probleme in den Prozess. Die Fehlerenstehungsrate steigt sprunghaft.

Reifen umfasst viele kleine Verbesserungsversuche mit Erfolgkontrolle. Abnehmende Fehlerentstehungsrate, bis das Potential der neuen Fähigkeiten ausgereizt ist.

Unter Einbeziehung der Fähigkeitsverbesserung in größeren Zeitschritten nimmt die Fehlerentstehungsrate tendentiell sägezahnförmig ab*.

* Für IT-Service-Leistungen hatten wir das Abnahmeverhalten der Fehlfunktionsrate, beim Reifen als IT-Entstehungsleistungen die Fehlerentstehungsrate, genauer untersucht, Abnahme mit Exponent 1 bis 2.

2.143 Projekte, Vorgehensmodelle, Kreativität

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 hat.

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

- erforderliche Fähigkeiten, Aufwand für den Reifeprozess,
- Phasenaufteilung, Beschränkung der Kreativität, ...

Allein diese anwendungsunabhängige Gemeinsamkeiten eröffnet interessante Blickwinkel, wie und wohin die Entwicklung von Technologien, Arbeitsabläufen in Institutionen und Behörden und auch die Ausbildung an Schulen verläuft.

2.144 Literatur

Literatur

[1] In *Ebert, C„ Dumke R. (Hrsg.), Software-Metriken in der Praxis*, pages 105–116. Springer, 1996.

[2] L.A. Clarke. A system to generate test data and symbolically execute programs. *IEEE Transactions on Software Engineering*, SE-2(3):215–222, 1976.

[3] Richard A. DeMillo, Richard J. Lipton, and Fred G. Sayward. Hints on test data selection: Help for the practicing programmer. 11(4):34–41, April 1978.

[4] Ludovic Pintard, Jean-Charles Fabre, Karama Kanoun, Michel Leeman, and Matthieu Roy. Fault injection in the automotive standard iso 26262: An initial approach. Oct 2017.