Entwicklung sicherheitskritischer Systeme

Definitionen

Definitionen

- Sicherheit = Freiheit von unvertretbaren Risiken
- Risiko = Kombination aus der Wahrscheinlichkeit, mit der ein Schaden auftritt, und dem Ausmaß dieses Schadens

vs.

- Mechanismen zur Risikominimierung:

Fehlervermeidung

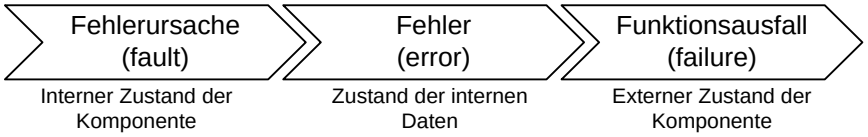
Fehlertoleranz durch Redundanz

Schadensminimierung

- Redundanz bezeichnet den Einsatz von mehr technischen Mitteln als für die spezifizierte Nutzfunktion eines Systems benötigt werden.
- Offene Frage: Wie definiert sich unvertretbar? → siehe Zertifizierungsstandards

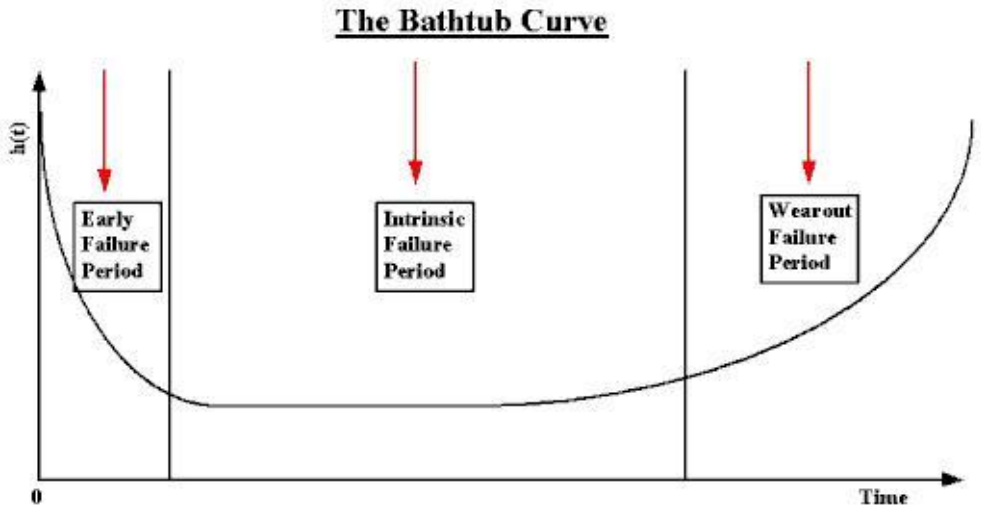
Begriff Fehler

- **Fehlerursache** (**fault**)**:** physikalischer Fehler oder Störstelle in einer Hardware- oder Softwarekomponente
- **Fehler** (**error**): Erscheinungsform eines Fehlzustands, z.B. durch das Abweichen eines Wertes vom erwarteten Wert in den internen Daten
- **Funktionsausfall** (**failure**): Ausfall oder fehlerhafte Durchführung von Funktionen eines Systems, Auftritt an der Benutzerschnittstelle



Fehlerrate

- Die Fehlerrate gibt die erwartete Anzahl an Fehler eines Gerätes oder eines Systems für eine gegebene Zeitperiode an.
- Typischerweise wird die Fehlerrate als konstant angenommen (siehe Badewannenkurve – gültig für Hardwarefehler) und mit λ bezeichnet. Typische Einheit der Fehlerrate ist Fehler pro Stunde.



Aspekte des Begriffs Fehlertoleranz

- Systeme zum Einsatz in sicherheitskritischen Anwendungen erfordern ein hohes Maß an Systemstabilität (**dependability**).
- Dieser Begriff umfasst:
  - Zuverlässigkeit
  - Sicherheit
  - Verfügbarkeit
  - Leistungsfähigkeit
  - Robustheit
  - Wartbarkeit
  - Testbarkeit

Zuverlässigkeit

- Definition: Die Zuverlässigkeit (**reliability**) eines Systems ist eine Funktion $0 \leq R(t) \leq 1$, definiert als die bedingte Wahrscheinlichkeit, dass das System korrekt während des Intervals $[t_0, t]$ funktioniert unter der Annahme, dass das System zum Zeitpunkt t_0 korrekt arbeitete.
- Wird eine konstante Fehlerrate angenommen, so kann die Zuverlässigkeit durch folgende Gleichung angegeben werden:

$$R(t) = e^{-(\lambda*(t-t_0))}$$

Sicherheit

- Sicherheit (**safety**) ist die Wahrscheinlichkeit $0 \leq S(t) \leq 1$, dass ein System zum Zeitpunkt t entweder korrekt arbeitet oder seine Funktion auf eine Art und Weise beendet, so dass es nicht die Funktionsweise anderer Systeme gestört oder Menschen gefährdet werden.
- Sicherheit ist damit ein Maßstab für die Fähigkeit eines Systems auf eine sichere Art und Weise auszufallen.
- Sicherheit und Zuverlässigkeit sind somit gegensätzliche Ziele:
  - Beispiel: Ampel, die keine Signale abgibt ist sicher, da die Autofahrer die Verkehrszeichen beachten, aber nicht zuverlässig

Verfügbarkeit

- Verfügbarkeit (**availability**) wird als eine Funktion $0 \leq A(t) \leq 1$ über die Zeit ausgedrückt, die die Wahrscheinlichkeit angibt, dass ein System zum Zeitpunkt *t* korrekt arbeitet. Im Gegensatz zur Zuverlässigkeit wird bei der Verfügbarkeit neben der Häufigkeit der Dienstausfälle auch die Dauer der Reparaturen und Wartungsarbeiten berücksichtigt.
- Während bei der Zuverlässigkeit die Korrektheit des Systems zu allen Zeitpunkten eines gegebenen Intervalls gefordert wird, gibt die Verfügbarkeit die momentane Wahrscheinlichkeit der korrekten Ausführung des Systems an.
- Eine hohe Verfügbarkeit ist beispielsweise bei transaktionsbasierten Systemen, z.B. ein Fluglinienreservierungssystem, nötig. Wartungsarbeiten und Reparaturen sollten schnell durchgeführt werden, eine andauernde korrekte Funktion im Sinne der Zuverlässigkeit wird hingegen nicht gefordert.

Leistungsfähigkeit

- In vielen Fällen ist es möglich und sinnvoll Systeme zu konstruieren, die nach Auftreten von Hardware oder Softwarefehler in einzelnen Komponenten (siehe spätere Einführung von Fehlerbereichen) in einem degradierten Modus weiterarbeiten.
- Unter **Leistungsfähigkeit** (**performability**) wird eine Funktion $0 \leq P(L,t) \leq 1$ über der Zeit verstanden, die eine Wahrscheinlichkeit angibt, dass die Funktionalität des Systems zum Zeitpunkt t mindestens das Niveau L erreicht. Im Gegensatz zur Verfügbarkeit, bei der immer nur die Wahrscheinlichkeit angegeben wird, dass alle Funktionen korrekt funktionieren, können nun auch Teilmengen betrachtet werden.

Robustheit, Wartbarkeit, Testbarkeit

- Unter **Robustheit** (**robustness**) eines Systems wird die Fähigkeit verstanden auch unter erschwerten Betriebsbedingungen (z.B. Fehleingaben (siehe Chemiefabrik) oder widersprüchlichen Meßwerten) die korrekte Funktionalität zu wahren.
- **Wartbarkeit** (**maintainability**) ist ein Maßstab für die Reparaturfreundlichkeit eines Systems. Quantitativ kann die Wartbarkeit als die Wahrscheinlichkeit M(t) ausgedrückt werden, dass das fehlerhafte System innerhalb einer Zeitdauer t repariert werden kann.
- **Testbarkeit** (**testability**) ist ein Maßstab für die Möglichkeit bestimmte Eigenschaften eines Systems zu testen. So kann es möglich sein, bestimmte Tests zu automatisieren und als Mechanismen in das System zu integrieren.
- Die Testbarkeit eines Systems ist durch die hohe Bedeutung der schnellen Fehleranalyse direkt mit der Wartbarkeit eines Systems verbunden.

Entwicklung sicherheitskritischer Systeme

Zertifizierungsstandards

Zertifizierungsstandards

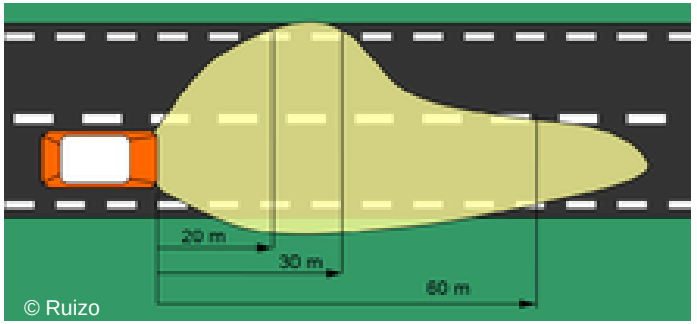
- Gründe für Entwicklung gemäß Zertifizierungsstandards:
  - Produkthaftung
  - Gesetzliche Anforderungen an die Zulassung
- Haftungsgesetz: ein Unternehmen haftet, wenn es zu Fehlern kommt, weil das Unternehmen nicht entsprechend dem Stand der Technik und der Wissenschaft entwickelt hat
- Zertifizierungsstandards sind Richtlinien für einen solchen Stand der Technik, sie definieren auch das akzeptable Risiko
- Anwendung von Zertifizierungsstandards ist notwendig, aber nicht hinreichend
- In manchen Industriebereichen wird die Standardeinhaltung zwingend gefordert (Nachweis durch Zertifizierungsbehörde z.B. TÜV)
- Abweichungen vom Standard sind immer möglich, sie müssen nur gut begründet sein

Zertifizierungsstandards im Überblick (Auswahl)

- IEC 61508 „Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme“ ➔ Domänenübergreifend, aber vor allem Automatisierungsdomäne
- Anpassungen der IEC 61508 an bestimmte Domänen:
  - IEC 61513: Kernkraftwerke — Leittechnik für Systeme mit sicherheitstechnischer Bedeutung – Allgemeine Systemanforderungen
  - EN 50128: Bahnanwendungen — Telekommunikationstechnik, Signaltechnik und Datenverarbeitungssysteme - Sicherheitsrelevante elektronische Systeme für Signaltechnik
  - ISO 26262: Road vehicles – Functional safety
- Im Luftfahrbereich gilt für die Softwareentwicklung der Standard DO-178C

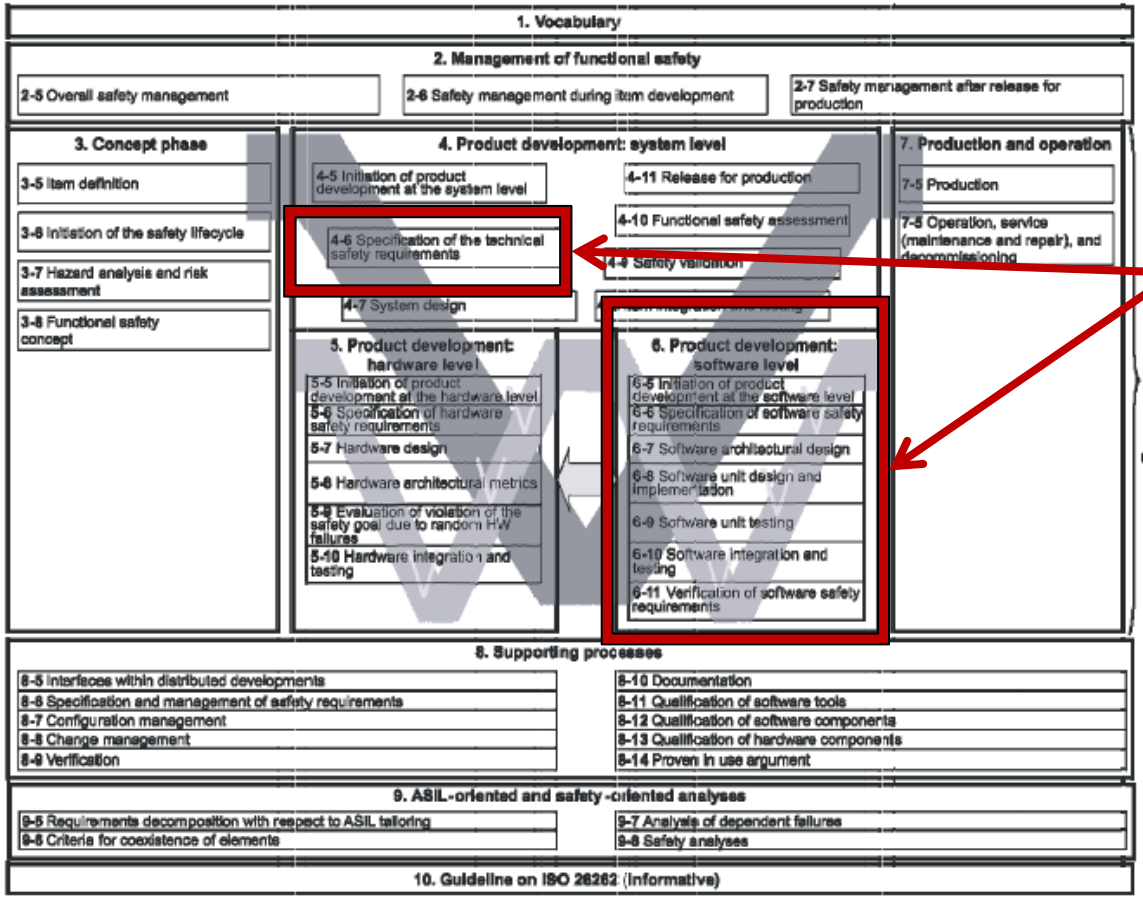
Laufendes Beispiel – Lichtmanagement im Fahrzeug

- Entwickelt werden soll die Ansteuerung des Abblendlichtes im Fahrzeug



- Mögliche Fehlfunktionen (vereinfacht):
  - Licht schaltet sich trotz Betätigung des Lichtschalters nicht ein
  - Licht schaltet sich plötzlich ohne Benutzerinteraktion aus

Für Beispiel relevanter Standard: ISO 26262



Schwerpunkte dieser Vorlesung (Softwareentwicklung)

Nicht behandelt werden:
-Management der funktionalen Sicherheit (Organisationsbezogene Massnahmen)
- Sicherheitslebenszyklus: Fokus Vorlesung auf Entwicklung
- Dokumentation

Entwicklung sicherheitskritischer Systeme

Sicherheitsanalyse & ASIL-Einstufung

ASIL-Einstufung

- Sicherheitskritikalitätslevel:
  - Der Automotive Safety Integrity Level gibt eine Einschätzung ab, wie kritisch eine Funktion einzustufen ist
  - Die niedrigste Stufe ist QM (Qualitätsmanagement), die höchste ASIL D
  - Es wird zwischen zwei Betriebsarten unterschieden: Betriebsart mit niedrigen Anforderungsraten (z.B. Auslösen des Airbags bei Unfall) bzw. Betriebsart mit hoher oder kontinuierlicher Anforderungsrate (z.B. Nichtauslösen des Airbags bei normaler Fahrt)
- Einstufung:
  - Im ISO 26262 Standard wird die Einstufung anhand von drei Parametern vorgenommen:
    - Faktor S (Severity): mögliches Schadensausmaß im Fall eines Systemversagens
    - Faktor E (Exposure): Aufenthaltsdauer / -wahrscheinlichkeit in der gefährlichen Fahrsituation
    - Faktor C (Controllability): Kontrollierbarkeit der gefährlichen Fahrsituation
  - Die Einstufung muss für jede Art der Fehlfunktion durchgeführt werden, dabei werden typischerweise verschiedenen Szenarien durchgespielt

ASIL-Einstufung - Konsequenzen

- Konsequenzen der Einstufung:
  - Auf Basis der Einstufung werden quantitative Anforderungen (Grenzwerte für verschiedene Parameter des Systems), sowie qualitative Anforderungen an den Entwicklungsprozess vorgegeben
- Ausfallsgrenzwert - Probability of Failure on Demand / per Hour (PFD/PFH)
  - Der ASIL gibt vor wie wahrscheinlich ein Ausfall pro Betriebsstunde höchstens sein darf (z.B. $<10^{-8}$ Ausfälle/Betriebsstunde bei ASIL D und kontinuierlichen Betrieb)
    Einstufung:
- Grenzwert für den minimalen Anteil der sicher erkannten Ausfälle (Safe Failure Fraction - SFF)
  - Die SFF ergibt sich aus der gesamten sicheren Fehlerrate (sichere Ausfälle und erkannte gefährliche Ausfälle) geteilt durch die gesamte Fehlerrate (gesamte sichere Fehlerrate plus unerkannte gefährliche Ausfälle)
  - Wichtig: Die Berechnung erfolgt auf Teilsystemebene und das Teilsystem mit dem niedrigsten erreichbaren SIL bestimmt den erreichbaren SIL der gesamten Sicherheitsfunktion
  - Die vorgegebenen Grenzwerte richten sich auch nach dem Redundanzgrad, sowie der Komplexität der Hardwarekomponenten (z.B. mind. 90% bei einfachen Teilsystemen mit Redundanzgrad 2 für SIL 4 in der IEC 61508)

Parameter S: Potentielles Schadensmaß

Table 1 — Classes of severity

Class	S0	S1	S2	S3
Description	No injuries	Light and moderate injuries	Severe and life-threatening injuries (survival probable)	Life-threatening injuries (survival uncertain), fatal injuries

- Beispiele für verschiedene Klassen:
  - S1: Oberflächliche Wunden bis hin zu Gehirnerschütterung mit Bewusstlosigkeit bis zu 15 Minuten
  - S2: Schädelbrüche ohne Gehirnverletzung
  - S3: Wirbelsäulenfrakturen
- Der Standard berücksichtigt auch die Verteilung der möglichen Verletzungen auf die einzelnen Klassen
- Einstufung im laufenden Beispiel (kein Gewähr):
  - Kein Licht trotz Einschalten:
    - S3
  - Lichtausfall:
    - S3

Parameter E: Aufenthaltsdauer

Table B.2 — Classes of probability of exposure regarding duration/probability of exposure in driving situations

Class	E1	E2	E3	E4
Description	Very low probability	Low probability	Medium probability	High probability
Definition of duration/ probability of exposure	Not specified	< 1% of average operating time	1% - 10% of average operating time	> 10% of average operating time

- Beispiele für verschiedene Klassen:
  - E1: Verlorenes Gepäck auf der Autobahn
  - E2: Anhängerbetrieb
  - E3: Tunnelfahrt
  - E4: Beschleunigen
- Einstufung im laufenden Beispiel (kein Gewähr):
  - Kein Licht trotz Einschalten:
    - E3
  - Lichtausfall:
    - E3

Parameter C: Beherrschbarkeit

Table B.4 — Examples of possibly controllable hazards by the driver or by the endangered persons

Class	C0	C1	C2	C3
Description	Controllable in general	Simply controllable	Normally controllable	Difficult to control or uncontrollable
Definition	Controllable in general	99% or more of all drivers or other traffic participants are usually able to avoid a specific harm.	90% or more of all drivers or other traffic participants are usually able to avoid a specific harm.	Less than 90% of all drivers or other traffic participants are usually able, or barely able, to avoid a specific harm.

- Beispiele für verschiedene Klassen:
  - C0: Unerwartete Lautstärkesteigerung des Radios
  - C1: Anfahren mit verriegeltem Lenkradschloss
  - C2: Ausfall des ABS während einer Notbremsung
  - C3: Fehlverhalten der Lenkung bei mittlerer oder hoher Geschwindigkeit
- Einstufung im laufenden Beispiel (kein Gewähr):
  - Kein Licht trotz Einschalten:
    - C1
  - Lichtausfall:
    - C2

ASIL-Einstufung – Kein Gewähr

		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

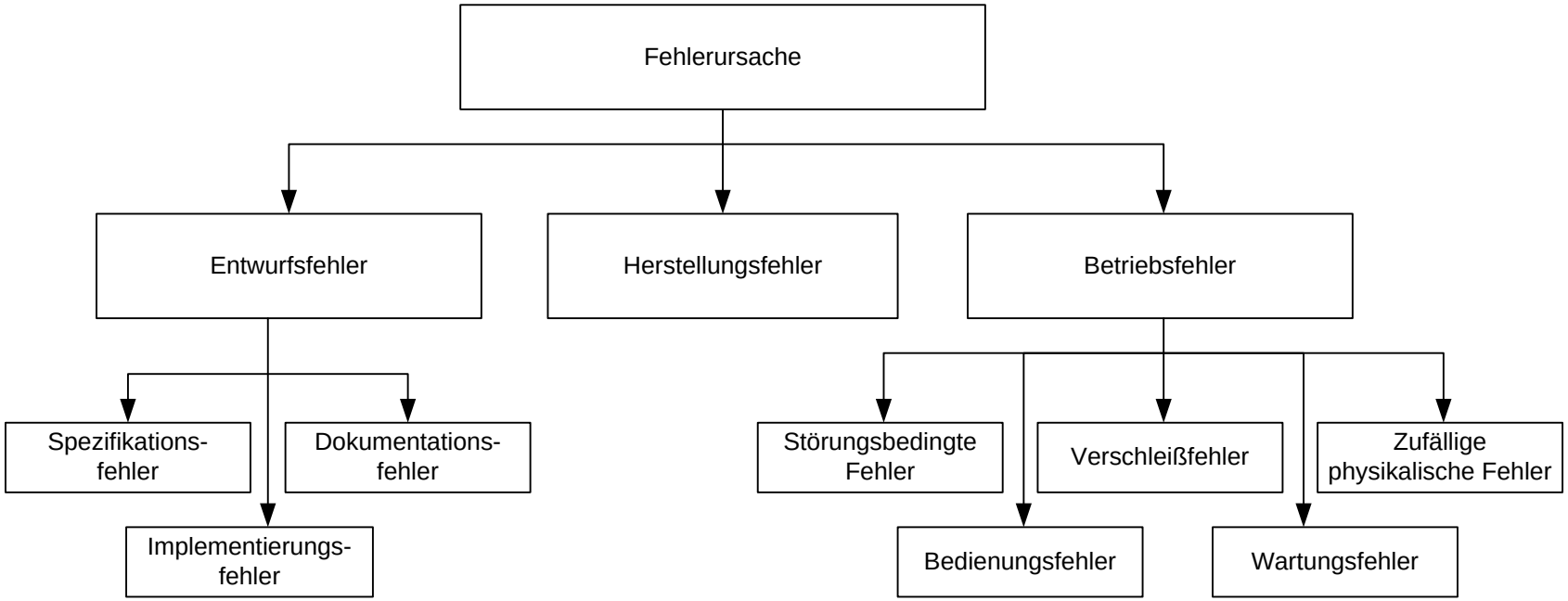
Kein Licht trotz Einschalten

Lichtausfall

Entwicklung sicherheitskritischer Systeme

Analyse der möglichen Fehler

Fehlerursachen



Klassifizierung von Fehlern

- Unterscheidung nach Entstehungsort:
  - Hardware
  - Software
- Unterscheidung nach Fehlerdauer:
  - permanent
  - intermittierend (flüchtig)
    - periodisch
    - wiederkehrend
    - einmalig

Beispiel Fehlerbilder Prozessoren und vorgeschlagene Erkennung

Table D.1 — Faults or failures to be analysed in the derivation of diagnostic coverage

Component	See Tables	Recommendations for diagnostic coverage		
		Low (60 %)	Medium (90 %)	High (99 %)
Processing units				
Register, internal RAM	D.4	Stuck-at (see footnote) for data and addresses	d.c. fault model (see foot note)for data and addresses	d.c. fault model for data and addresses Dynamic cross-over for memory cells No, wrong or multiple address-ing
Coding and execu-tion including flag register		Wrong coding or no execution	Wrong coding or wrong execution	No generic fault model avail-able. Detailed analysis neces-sary. Depends on CPU architec-ture
Address calculation		Stuck-at	d.c. fault model	No generic fault model avail-able. Detailed analysis neces-sary.
Interrupt handling		No or continuous inter-rupts	No or continuous inter-rupts Cross-over of interrupts	No or continuous interrupts Cross-over of interrupts

Table D.4 — Processing units

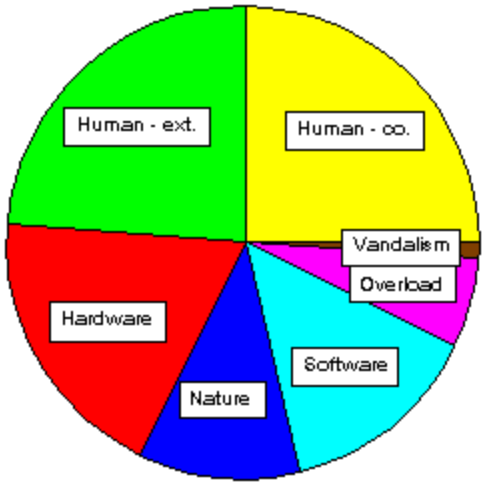
Diagnostic technique/measure	See overview of techniques	Maximum diagnostic coverage considered achievable	Notes
Self-test by software: limited num-ber of patterns (one channel)	D.2.3.1	Medium	Depends on the quality of the self test
Self-test by software cross ex-change between two independent units	D.2.3.5	Medium	Depends of the quality of the self test
Self-test by software: walking bit (one-channel)	D.2.3.2	Medium	Depends on the quality of the self test
Self-test supported by hardware (one-channel)	D.2.3.3	Medium	Depends on the quality of the self test
Coded processing (one-channel)	D.2.3.4	High	-

Beispiel: Fehlerquellen im öffentlichen Telefonnetz

- Welche Ursachen können Fehler haben:
  - Fehler durch Menschen (intern/extern)
  - Hardwarefehler
  - Softwarefehler
  - Fehler verursacht durch die Natur
  - Überlast
  - Vandalismus
- Weitere Informationen unter http://hissa.ncsl.nist.gov/kuhn/pstn.html.

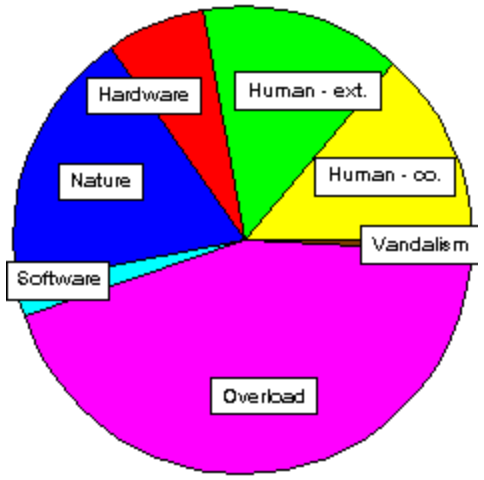
Ursachen und Wirkung

Figure 1: Number of Outages
(percent)



- Human error - company: 25%
- Human error - external: 24%
- Hardware failure: 19%
- Act of nature: 11%
- Software failure: 14%
- Overload - 6%
- Vandalism - 1%

Figure 2: Magnitude of Failure
(customer minutes- percent of total)



- Human error - company: 14%
- Human error - external: 14%
- Hardware failure: 7%
- Act of nature: 18%
- Software failure: 2%
- Overload: 44%
- Vandalism - 1%

Fehlermodell

- Um die Fehlertoleranz-Fähigkeit eines Rechensystems spezifizieren zu können, ist eine Fehlervorgabe erforderlich, welche die Menge der zu tolerierenden Fehler auf ein formales Fehlermodell angibt.
- Ein Fehlermodell hat den Zweck zu jedem Zeitpunkt die Fehlermöglichkeiten eines Systems als eine Obermenge der Menge der zu tolerierenden Fehler anzugeben.
- Das Fehlermodell beinhaltet daher
  - die Komponenten, die von Fehlern betroffen sein können (strukturelle Fehlerbetrachtung) und
  - in welcher Art und Weise deren Funktion beeinträchtigt wird (funktionelle Fehlerbetrachtung)

Fehlerbereich

- Typischerweise wird angenommen, dass Fehler nur in bestimmten Teilmengen der Menge aller Komponenten *S* auftreten. Jede dieser Komponentenmengen wird als **Fehlerbereich** *Fb* bezeichnet.
- Die Annahmen
  - $Fb_1 \cup ... \cup Fb_n \neq S$ (→ es gibt einen Perfektionskern $S \setminus (Fb_1 \cup ... \cup Fb_n)$
  - EXISTS $i,j \in \{1...n\}: Fb_i \cap Fb_j \neq \emptyset$ (→Überschneidungen sind erlaubt)

  sind zulässig.

k-Fehler-Annahme

- Da die Anzahl der Fehlerbereiche mitunter sehr groß werden kann, bietet sich als Spezialfall der Fehlerbereichsannahme die k-Fehler-Annahme an.
- Grundlage hierfür ist die disjunkte Zerlegung eines Systems S in Einzelfehlerbereiche $Eb_1,...,Eb_m$ mit $Eb_1 \cup ... \cup Eb_m = S$. Die k-Fehlerannahme fordert die Tolerierung von allen Fehlern, die sich auf bis zu k Einzelfehlerbereiche erstrecken.
- Die bei k-Fehler-Annahme mit $k \geq 2$ zu tolerierenden Fehlerfälle werden Mehrfachfehler genannt. Es wird jedoch nicht zwischen zufälligen und systematischen Mehrfachfehlern unterschieden. Dieser Unterschied muss jedoch bei der Anfälligkeitsanalyse genau betrachtet werden.
- Beispiel: 3-Rechner-System, als Einzelfehlerbereiche werden die einzelnen Rechner angesehen

Fehlfunktionsannahmen

- Detaillierung der Fehlervorgabe durch **Fehlfunktionsannahme**. Sinnvolle Annahmen sind:
  - Teil-Ausfall: nur manche Funktionen eines Systems fallen aus, die übrigen werden korrekt erbracht
  - Unterlassungs-Ausfall: es wird entweder ein richtiges oder gar kein Ergebnis ausgegeben (ommision fault, fail-silent)
  - Anhalte-Ausfall: sobald ein Fehler aufgetreten ist, gibt das System kein Ergebnis mehr aus (fail-stop): jedes ausgegebenen Ergebnis ist korrekt und es fehlt kein früheres Ergebnis
  - Haft-Ausfall: ab Auftreten eines Fehlers wird immer das gleiche Ergebnis ausgegeben
  - Inkonsistenz-Ausfall: ausgegebene fehlerhafte Ergebnisse sind in sich nicht konsistent (z.B. CRC)
  - Binärstellen-Ausfall (oder k-Binärstellenausfall): Fehler verfälschen maximal k Binärstellen eines Ergebnisses
  - Nicht-Angriffs-Ausfall: z.B. Schutz von fehlerfreien Komponenten vor falscher Authentifikation fehlerhafter Komponenten

Fehlerausbreitung und -eingrenzung

- Fehler breiten sich in der Regel ohne geeignete Maßnahmen innerhalb eines Systems aus. Fehlertoleranzverfahren basieren jedoch zumeist auf einer eingeschränkten Fehlervorgabe. So kann zumeist nur eine begrenzte Anzahl an fehlerhaften Komponenten toleriert werden.

→ Eingrenzungsmaßnahmen müssen getroffen werden.

- Typischerweise werden deshalb Maßnahmen zur Isolierung getroffen:
  - Hardwarekomponenten werden räumlich getrennt oder gekapselt.
  - Software wird so strukturiert, dass möglichst viele Berechnungen in einzelnen Modulen erfolgt.
  - An Schnittstellen werden Inkonsistenzprüfungen zwischen den einzelnen Komponenten durchgeführt.

Entwicklung sicherheitskritischer Systeme

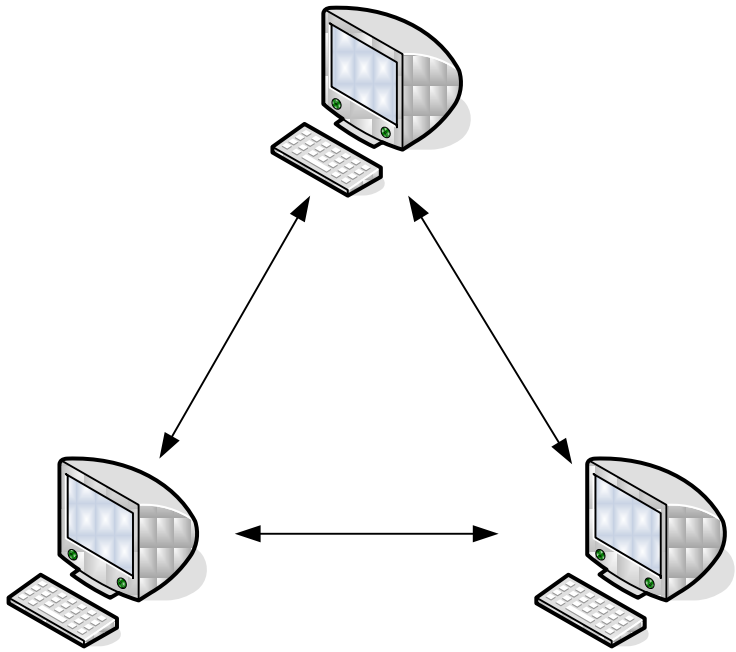
Fehlererkennung

Grundlage der Fehlererkennung: Redundanz

- Die beiden grundsätzlichen Schritte eines Fehlertoleranzverfahrens, die Diagnose und Behandlung von Fehlern, benötigen zusätzliche Mittel, die über die Erfordernisse des Nutzbetriebs hinausreichen.
- All diese zusätzlichen Mittel sind unter dem Begriff **Redundanz** zusammengefasst.
- Redundanz bezeichnet also den Einsatz von mehr technischen Mitteln, als für die spezifizierte Nutzfunktion eines Systems benötigt werden.

Typische Ausprägung von Redundanz: 2-von-3-System

- Ein 2-von-3 System / TMR-System (triple modular redundancy) besteht aus 3 gleichwertigen Komponenten.
  - Ein Ausfall einer Komponente kann toleriert werden, ohne dass die Funktion beeinflusst wird.
  - Bei einem Ausfall einer zweiten Komponente muss in einen sicheren Modus geschaltet werden (nur eingeschränkt möglich).
- Betriebmodi:
  - sicherer und zuverlässiger Betrieb (2-von-3-Betrieb)
  - sicherer Betrieb (2-von-2-Betrieb)



Zuverlässigkeit redundanter Systeme

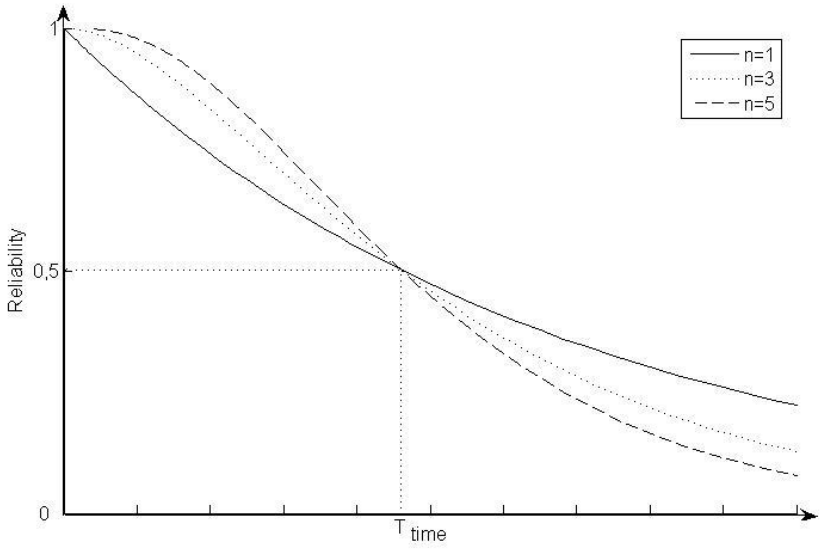
- Redundanz kann, muss aber nicht die Zuverlässigkeit verbessern:
- Beispiel: 2-von 3 System, stochastisch unabhängige Fehler, konstante Ausfallsrate λ, R_1: Zuverlässigkeit einer Komponte, R_3:Zuverlässigkeit des TMR-Systems

$$\rightarrow R_3(t) = R_1(t)^3 + 3 * R_1(t)^2 * (1 - R_1(t))$$

- Allgemeiner Fall m-von-n System:

$$\rightarrow R_n(t) = \sum_{k=m}^{n} \binom{n}{k} R_1^k * (1 - R_1)^{n-k}$$

→ ohne Möglichkeiten zur Reparatur sinkt die Zuverlässigkeit des Redundanten Systems nach einer Zeitdauer T unter die Zuverlässigkeit eines einfach ausgelegten Systems.



Redundanzarten

- Redundanz ist möglich in:
  - Hardware (strukturelle Redundanz)
  - Information
  - Zeit
  - Software (funktionelle Redundanz)
    - Zusatzfunktion
    - Diversität

→ Fehlertolerante Rechensysteme setzen zumeist Kombinationen verschiedener redundanter Mittel ein.