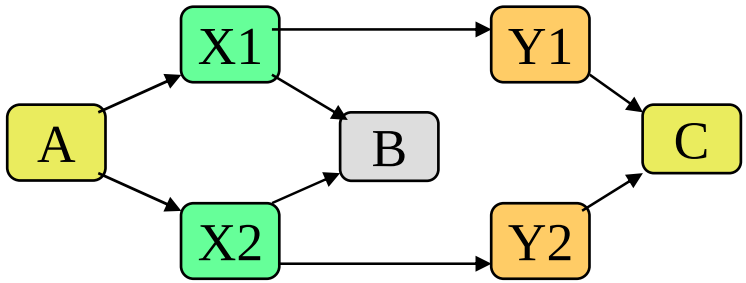
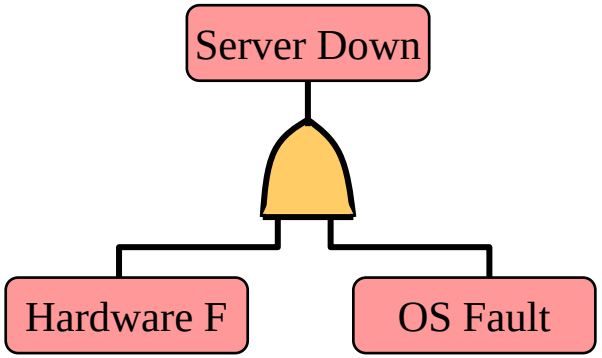
Modellierung und Analyse eingebetteter und verteilter Systeme

—

Thread „Zuverlässigkeit“ *Teil 1*



- Begriffe und Kenngrößen
- Fehler und Ausfälle
- Zuverlässigkeitsblockdiagramm
- Struktur-Funktionsmodell
- Fehlermodell und Ausbreitung
- Redundanz
- Fehlertoleranzverfahren
- Qualitätssicherung, Analyse und Bewertung
- Formale Ansätze



F: Funktionaler Thread – Inhalte

- **Begriffe und Kenngrößen**
- **Fehler und Ausfälle**
- **Zuverlässigkeitsblockdiagramm**
- **Strukturfunktionsmodell**
- **Fehlermodell und Ausbreitung**
- **Redundanz**
- **Fehlertoleranzverfahren**
  - **Fehlerdiagnose**
  - **Fehlerbehandlung**
- **Qualitätssicherung, Analyse und Bewertung**
  - **FMEA**
  - **HAZOP**
  - **Fault Trees**
- **Formale Ansätze**

Literatur

K. Echtle: Fehlertoleranzverfahren, Springer Verlag, 1990.
D. P. Siewiorek, R. S. Swarz: The theory and practice of reliable system design; Digital Press, 1982.
Blockley, David: Engineering Safety, Mcgraw Hill, 1992.
Leveson, Nancy: Safeware: System Safety and Computers, Addison Wesley, 1995.
W. Fredrich: Lehrblattsammlung Zuverlässigkeit und Qualitäts-sicherung, Uni Rostock, Fakultät Informatik und Elektrotechnik, 2003.

RAMSS – Eigenschaften

Reliability, **A**vailability, **M**aintainability, **S**afety, and **S**ecurity

- Reliability – Zuverlässigkeit:
  Funktionsfähigkeit in Betriebszeit
- Availability – Verfügbarkeit:
  Funktionsfähigkeit bei Anforderung
- Maintainability – Instandhaltbarkeit:
  Dauer für Reparaturen
- Safety – Funktionssicherheit:
  Gefährdungsfreiheit
- Security – Datensicherheit:
  Schutz gegen unberechtigten Zugriff

Normen
DIN VDE 31000, DIN 19250, DIN 19251
MIL - Standards
VDI/VDE 2180, VDI/VDE 3541, VDI/VDE 3542
IEC 880, IEC 61508

Funktionssicherheitsgrad – Safety Integrity Level (SIL)

Wahrscheinlichkeit, dass ein sicherheitsrelevantes System die Sicherheitsanforderungen erfüllt

- hinsichtlich systematischer Fehler
- hinsichtlich zufallsbedingter Fehler

Tolerable Hazard Rate (THR)

- Anzahl zu erwartender sicherheitsrelevanter Störungen pro Betriebsstunde und Systemfunktion

Sicherheitsanforderungsstufen (SIL-Werte) nach IEC 61508

- **SIL 4**: THR $< 10^{-8}$
- **SIL 3**: $10^{-8} <$ THR $< 10^{-7}$
- **SIL 2**: $10^{-7} <$ THR $< 10^{-6}$
- **SIL 1**: $10^{-6} <$ THR $< 10^{-5}$

Zuverlässige Systeme: Verfahren

A] Produkte ohne besondere Funktionssicherheitsanforderungen
(z.B. Komfortfunktionen)

Qualitätssicherung, Standards: Fehlervermeidung, Perfektionierung

EN 29000 – EN 29004 (ISO 9000 – ISO 9004)

EN 29000 – EN 29004 (ISO 9000 – ISO 9004)

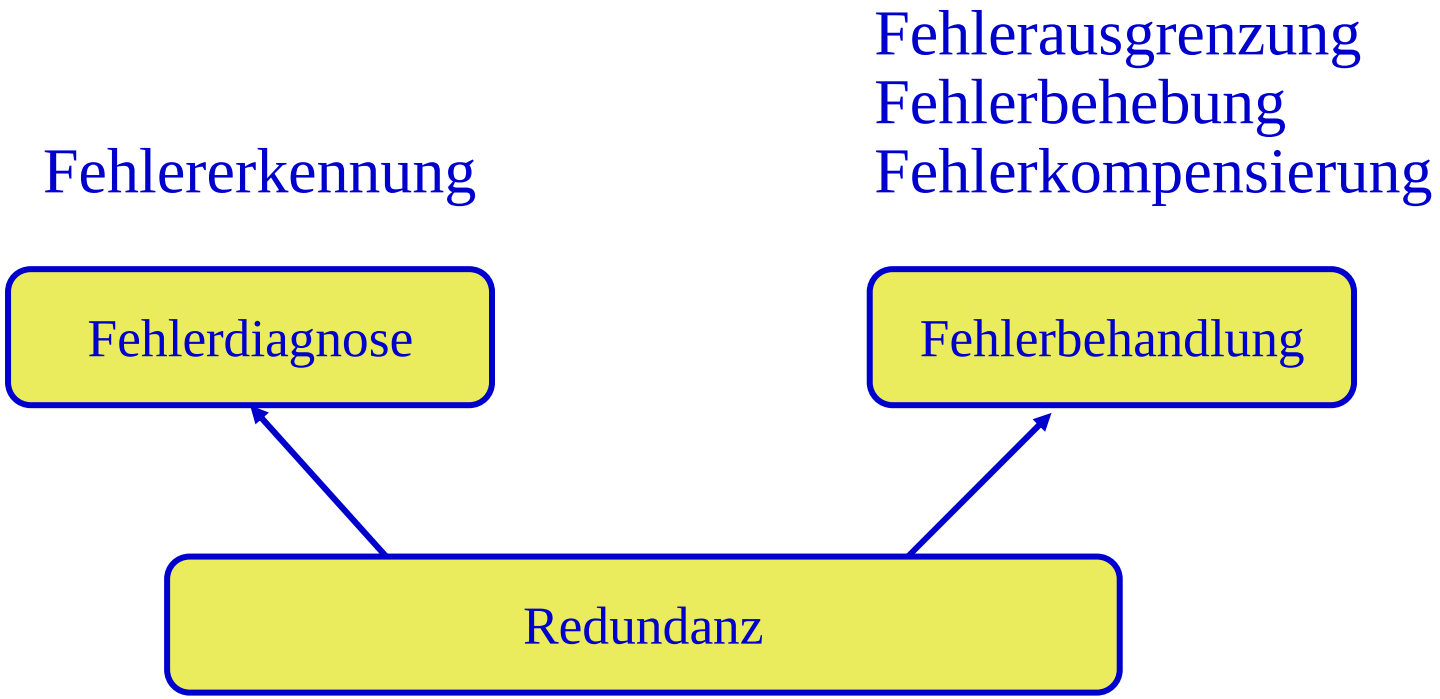
B] Produkte mit besonderen Funktionssicherheitsanforderungen
(z.B. Medizingeräte, Verkehrssteuerung, Chemieanlagensteuerung)

Zertifizierung (gesetzliche Grundlagen)

Produktgestaltung, Bauplan
Komponenten und Materialien
Produktionsverfahren
Analyse
Prüfung, Test
Bewertung

Erfahrungen auswerten, Statistiken, Bauteildatenbanken

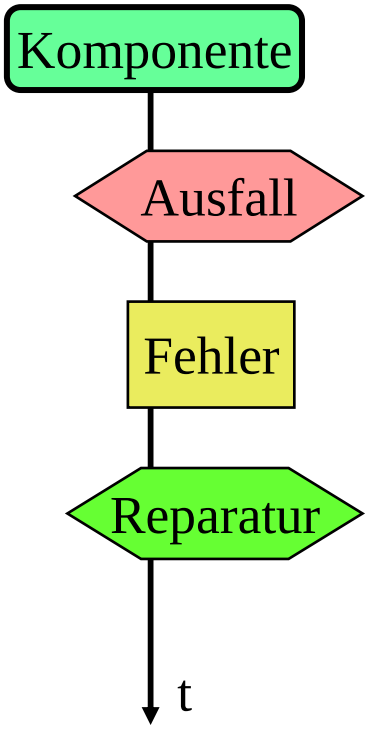
Zuverlässige Systeme – Gestaltung: Fehlertoleranz



- Zeit
- Struktur
- Funktion
- Information

- statisch / dynamisch / hybrid
  (e.g. hot / cold standby)
- ungenutzt, fremdgenutzt, gegenseitig

Z1: Begriffe



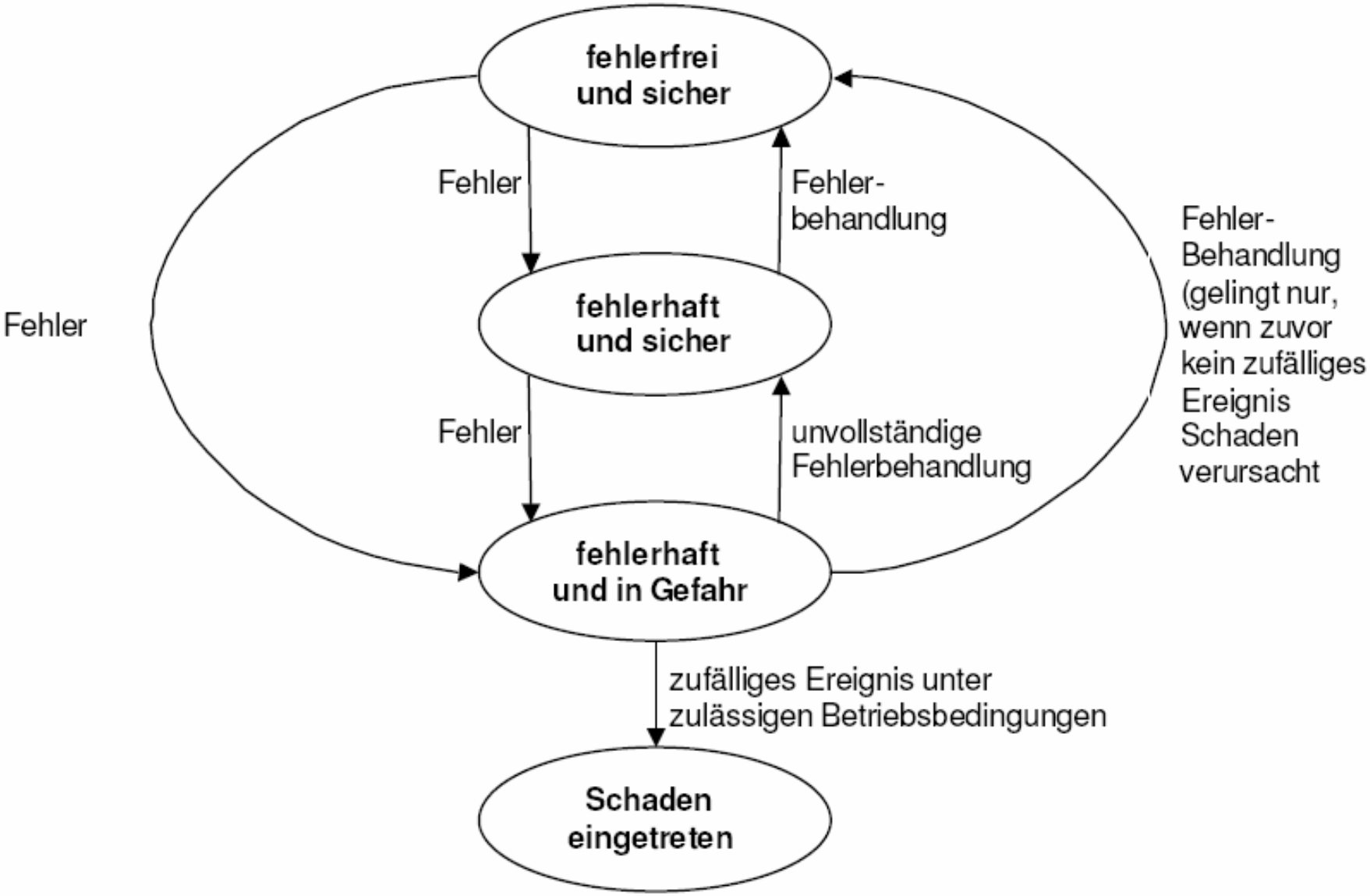
Zuverlässigkeit – Reliability
Fähigkeit einer Komponente ihre Funktion über eine bestimmte Betriebszeit in korrekter Weise zu erfüllen

Ausfall – Failure
Ereignis / Zustandsübergang von funktionsfähig nach fehlerhaft

Fehler – Fault
Zustand des Nichterfüllens einer Anforderung

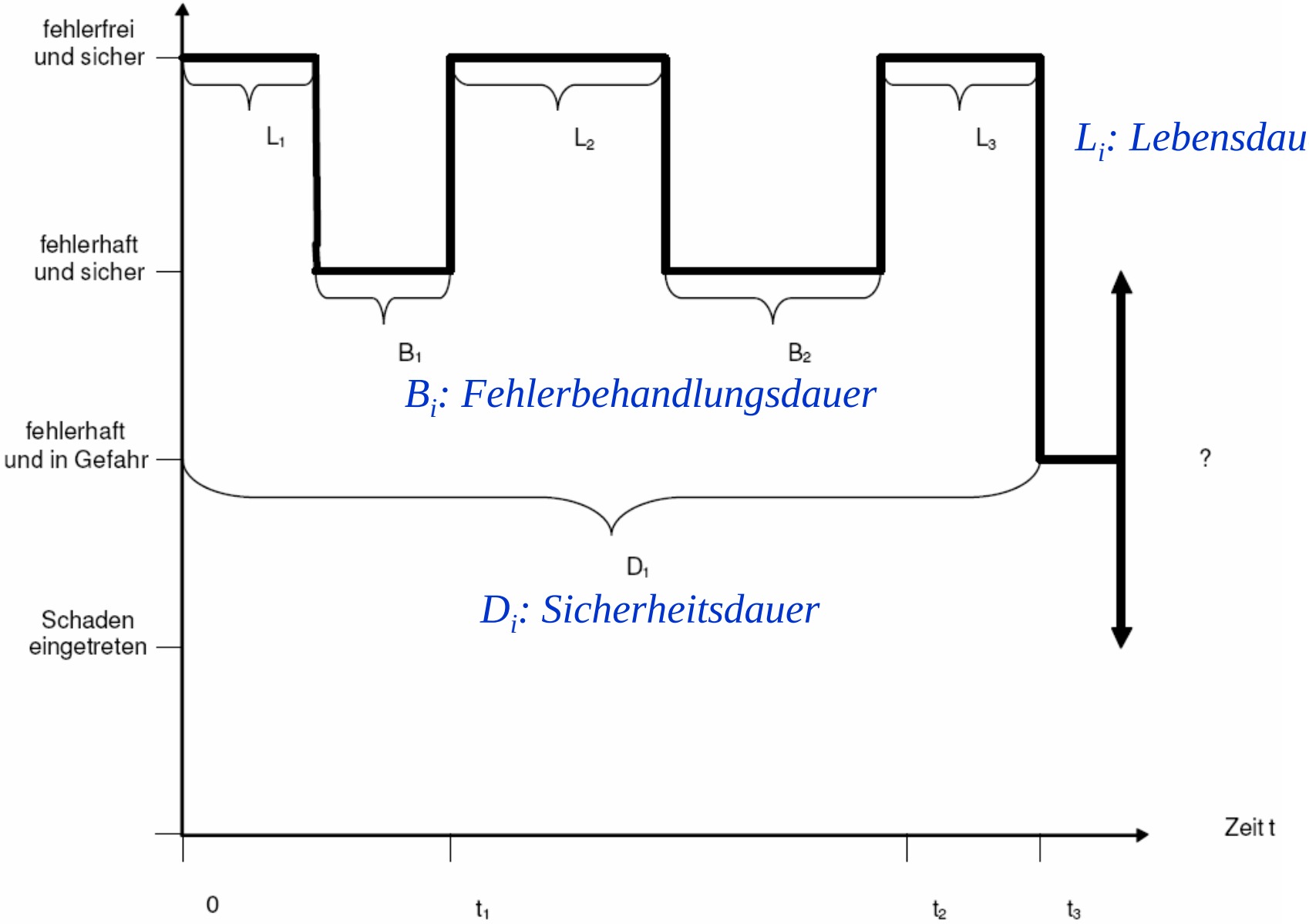
Funktionssicherheit – Safety
Zuverlässigkeit hinsichtlich sicherheitsrelevanter Funktionen:
- *sicherheitsrelevante Ausfälle und Fehler*

Z1: Systemzustände



aus K. Echtle: Fehlertoleranzverfahren, Springer Verlag, 1990.

Z1: Systemzustände



L_i: Lebensdauer

B_i: Fehlerbehandlungsdauer

D_i: Sicherheitsdauer

aus K. Echtle: Fehlertoleranzverfahren, Springer Verlag, 1990.

Z1: Zuverlässigkeitskenngrößen

Fehlerwahrscheinlichkeit $F_L(t)$ – Dichte $f_L(t)$

Wahrscheinlichkeit, dass ein zu Beginn fehlerfreies System im Zeitintervall [0,t] fehlerhaft wird

Überlebenswahrscheinlichkeit R(t)

Wahrscheinlichkeit, dass ein fehlerfreies System im Zeitintervall [0,t] ununterbrochen fehlerfrei ist

$$R(0) = 1 \qquad R(t) = 1 - F_L(t) \in [0, 1] \qquad \lim_{t \to \infty} R(t) = 0$$

Mittlere Lebensdauer E(L)

Erwartungswert der Zeitdauer bis zum ersten Fehler

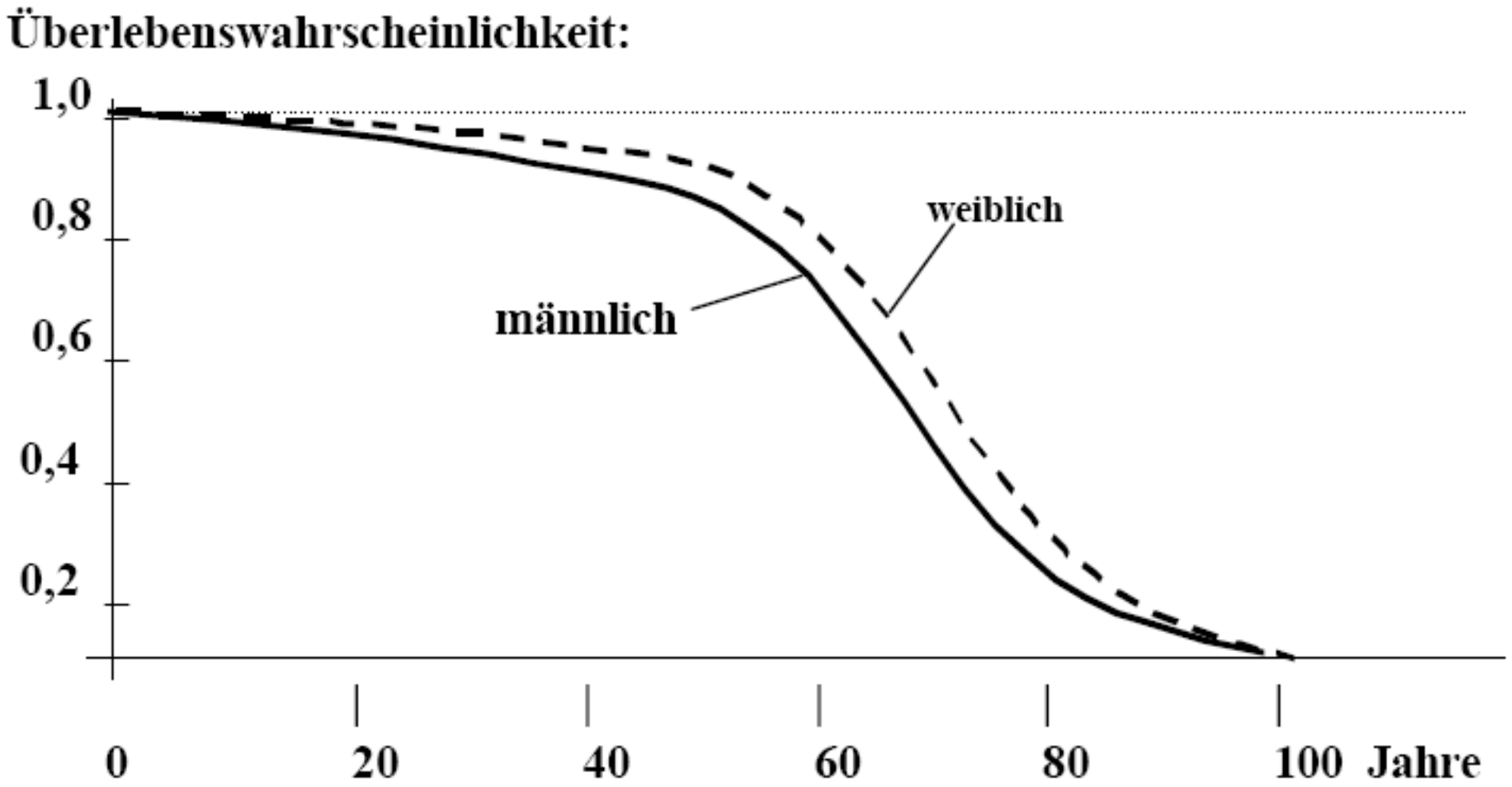
$$E(L) = \int_{-\infty}^{\infty} t \cdot f_L(t)\, dt = \int_{0}^{\infty} R(t)\, dt$$

Ausfallrate λ(t)

Anzahl der in einer Zeiteinheit ausfallenden Komponenten

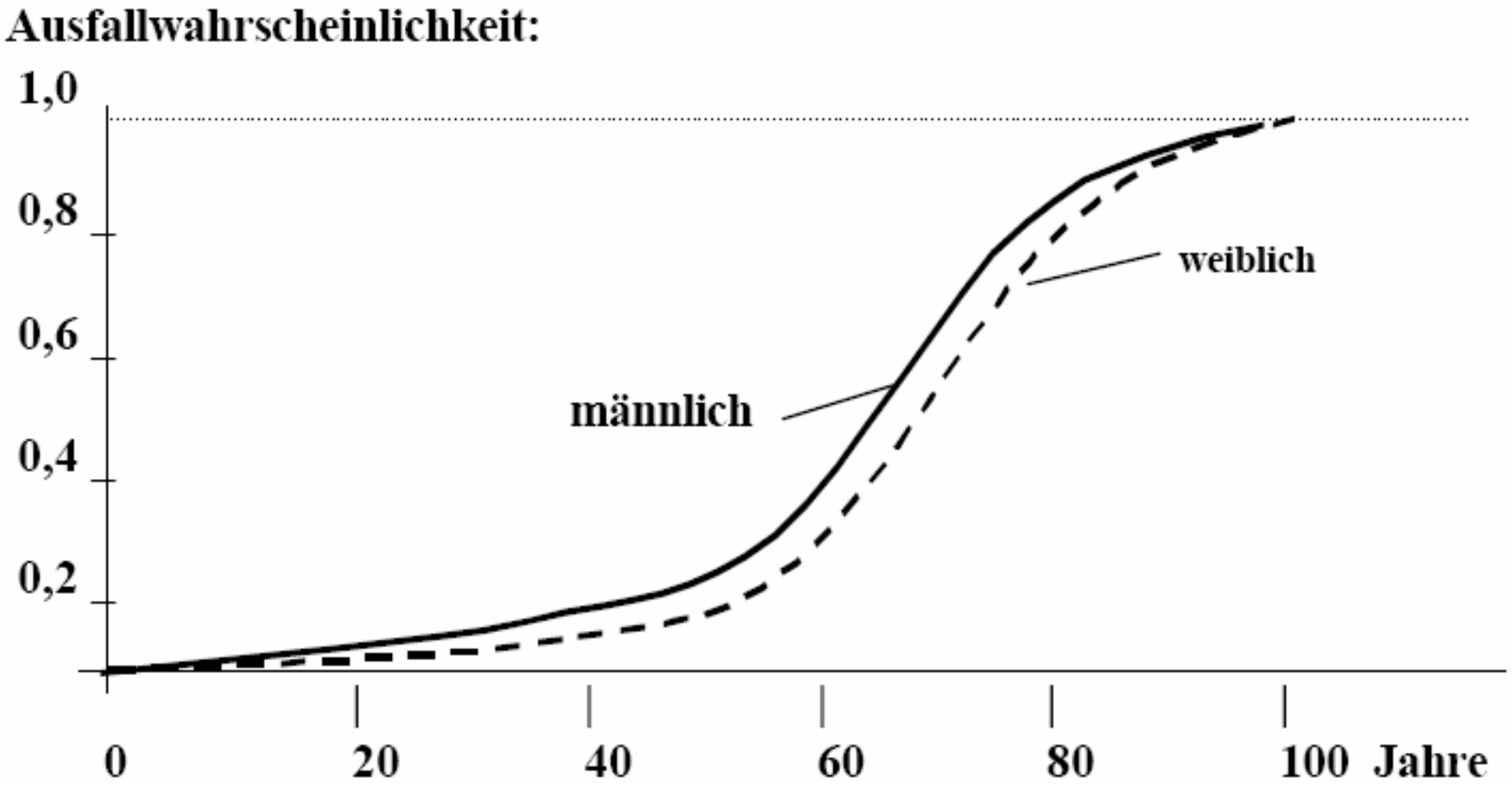
$$\lambda(t) = f_L(t) / R(t)$$

Z1: Beispiel Mensch



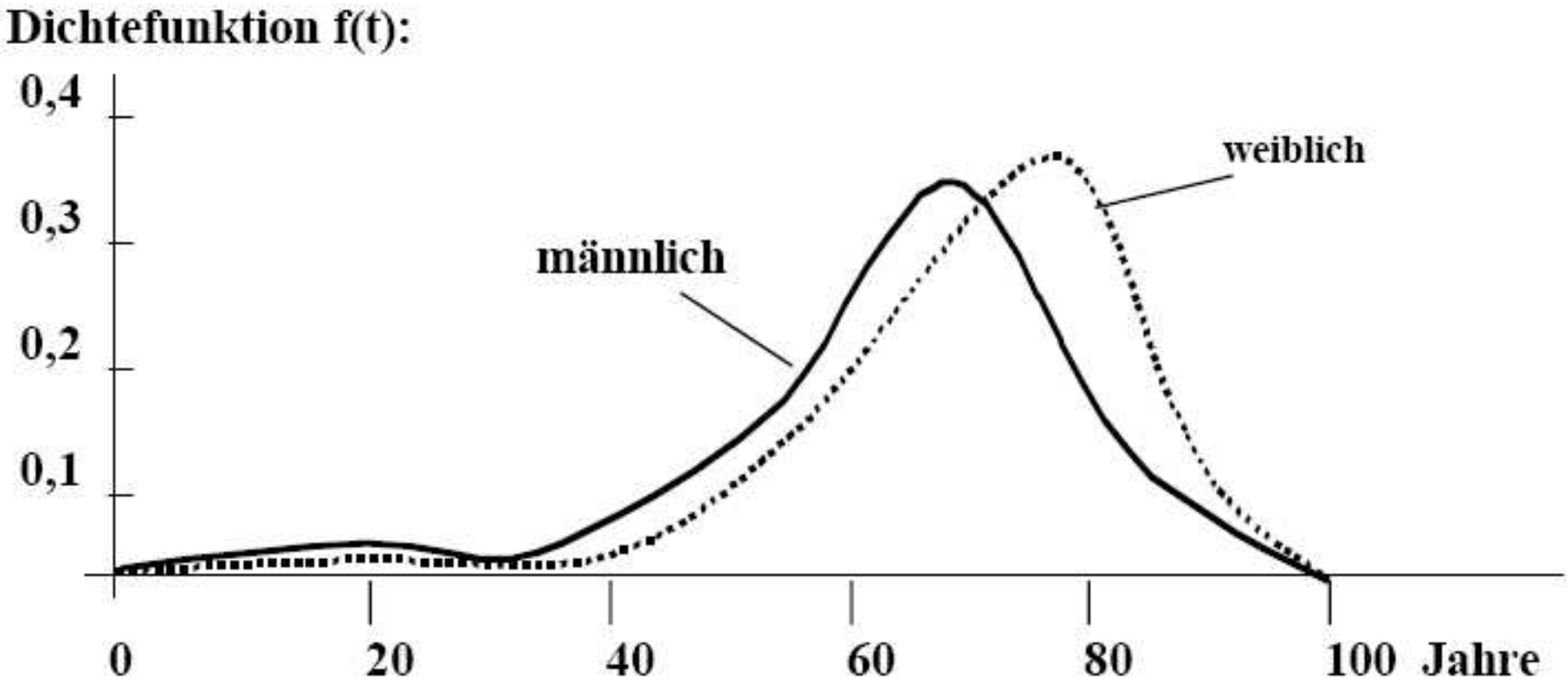
aus W. Fredrich: Zuverlässigkeit und Qualitätssicherung, Uni Rostock, 2003

Z1: Beispiel Mensch



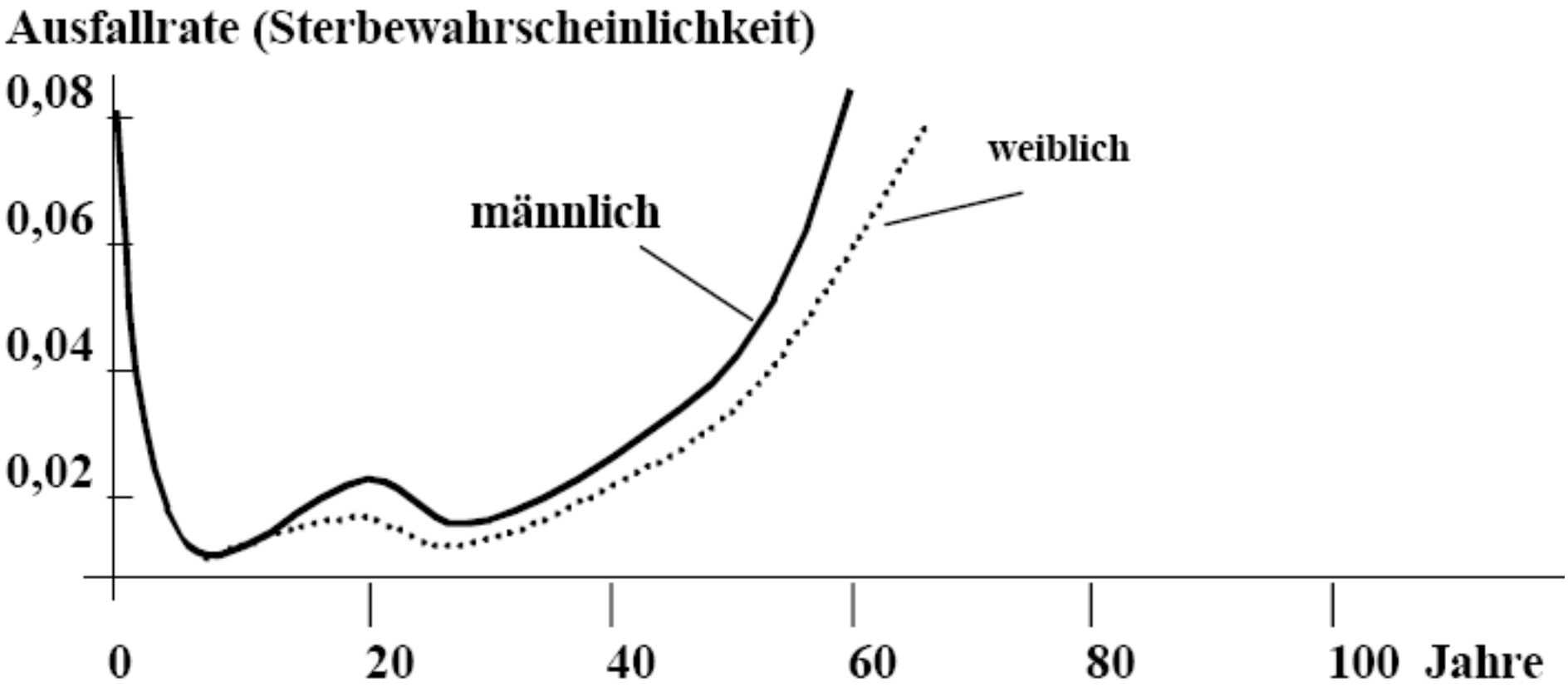
aus W. Fredrich: Zuverlässigkeit und Qualitätssicherung, Uni Rostock, 2003

Z1: Beispiel Mensch



aus W. Fredrich: Zuverlässigkeit und Qualitätssicherung, Uni Rostock, 2003

Z1: Beispiel Mensch



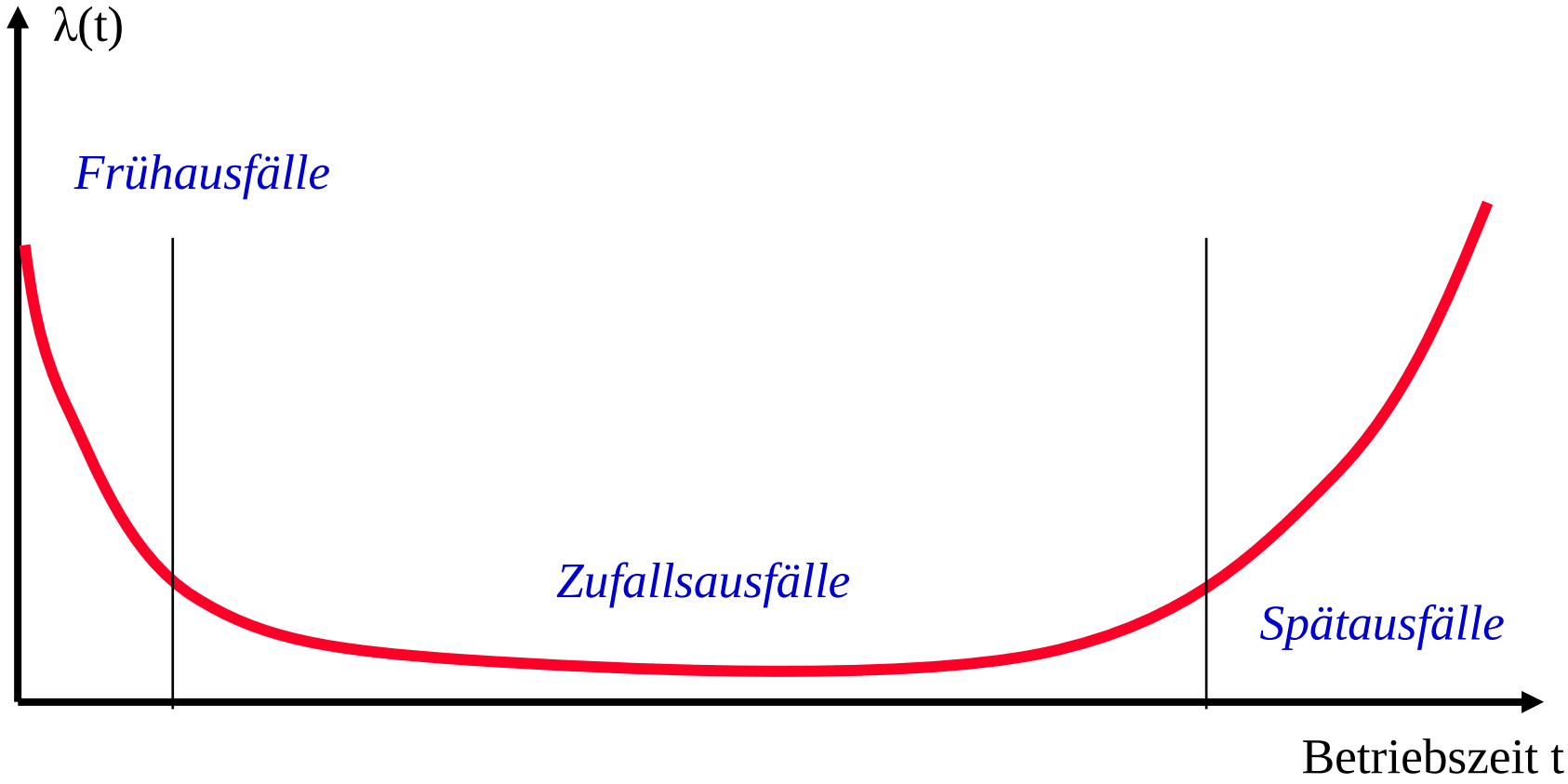
aus W. Fredrich: Zuverlässigkeit und Qualitätssicherung, Uni Rostock, 2003

Z1: Zuverlässigkeitskenngrößen

Ausfallrate λ(t)

Anzahl der in einer Zeiteinheit ausfallenden Komponenten $\lambda(t) = f_L(t) / R(t)$

Badewannen-Kurve



Z1: Zuverlässigkeitskenngrößen

Verfügbarkeit V

Wahrscheinlichkeit, ein System zu einem beliebigen Zeitpunkt fehlerfrei anzutreffen

$$V = \frac{E(L)}{E(L) + E(B)}$$

E(L), E(B) Erwartungswerte für die Nutzzeitdauer L und die Defektzeitdauer B

Mean Time Between Failures MTBF

Reziproker Wert der Ausfallrate,
mittlere Zeit zwischen zwei Fehlern,
mittlere ausfallfreie Betriebszeit $MTBF = 1/z_{gesamt}$

Mean Time To Repair MTTR

mittlere Zeit zur Reparatur nach Fehler

Mittlere Verfügbarkeit V

Wahrscheinlichkeit, dass das System funktionsfähig ist

$$V = MTBF / (MTBF + MTTR)$$

Z1: Zuverlässigkeitskenngrößen

Gefährdungswahrscheinlichkeit $F_D(t)$

Wahrscheinlichkeit, dass ein System im Zeitintervall [0, t] in einen Gefahrenzustand gerät

Sicherheitswahrscheinlichkeit S(t)

Wahrscheinlichkeit, das sich ein System im Zeitintervall [0, t] ununterbrochen in sicheren Zuständen befindet $S(t) = 1 - F_D(t) \in [0, 1]$

Mittlere Sicherheitsdauer E(D)

Erwartungswert der Zeitdauer, bis ein unsicherer Zustand auftritt

$$E(D) = \int_{-\infty}^{\infty} t \cdot f_D(t)\, dt = \int_{0}^{\infty} S(t)\, dt$$

Z1: Zuverlässigkeit von Bauteilen

Beanspruchungen erhöhen die Ausfallrate

- Temperatur und Temperaturwechsel bei allen elektrotechnischen Bauteilen
- Spannung bei Kondensatoren
- Verlustleistung bei Halbleiterbauelementen, Widerständen, Transformatoren
- Strom bei Kontaktstellen, Leitungen, Dioden
- Beschleunigung bei Röhren, Leitungsverbindungen
- Atmosphäre bei Kontaktstellen
- Elektrische Feldstärke bei Halbleiter–Bauelementen (insb. MOS – Bauelemente)

Weitere Einflüsse

- geschützter Transport (richtige Verpackung)
- richtige und zeitbegrenzte Lagerung
- konstruktive Ausführung
- Materialeinsatz (Materialkombinationen)
- Klima, Feuchte, Korrosion
- mechanische und dynamische Beanspruchung
- elektrische Belastung

aus W. Fredrich: Zuverlässigkeit und Qualitätssicherung, Uni Rostock, 2003

Z1: Zuverlässigkeit von Bauteilen

Relative Häufigkeit der Ausfallarten elektronischer Halbleiterbauelemente

Bauelement	*Kurzschluss*	*Unterbr.*	*Drift*	*Fehler*
Digitale bipolare IC`s	30%	30%	10%	30%
Digitale MOS – IC´s	20%	10%	30%	40%
Lineare IC´s	30%	10%	10%	50%*
Bipolare Transistoren	70%	20%	10%	-------
Feldeffekt-Transistoren	80%	10%	10%	-------
Mehrzweck-Dioden	70%	30%	-------	-------
Zener-Dioden	60%	30%	10%	-------
Hochfrequenz-Dioden	80%	20%	-------	-------
Thyristoren	20%	20%	60%	-------
Optoelektr.Bauelemente	10%	50%	40%	-------

**: Überlast*

aus W. Fredrich: Zuverlässigkeit und Qualitätssicherung, Uni Rostock, 2003

Z1: Zuverlässigkeit von Bauteilen

Relative Häufigkeit der Ausfallarten passiver elektronischer Bauelemente

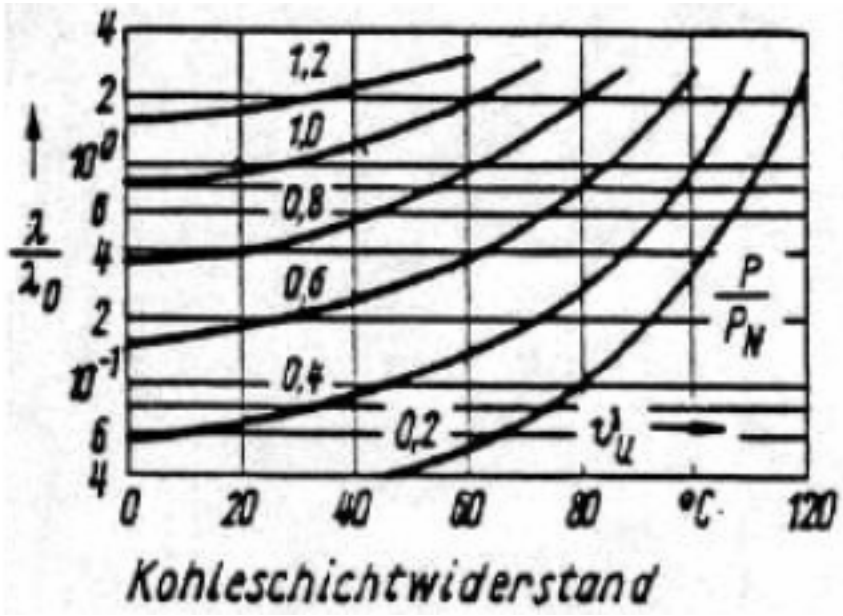
Bauelement	*Kurzschluss*	*Unterbr.*	*Drift*	*Fehler*
Festwiderstände	-------	90%	10%	-------
Variable Widerstände	-------	60%	20%	20%**
Folienkondensatoren	80%	10%	10%	-------
Metallfolienkondensatoren	40%	60%	-------	-------
Keramikkondensatoren	50%	40%	10%	-------
Tantalkondensatoren	60%	20%	20%	-------
Aluminiumelektrolytkond.	20%	10%	70%	-------
Spulen	10%	30%	--------	60%***
Relais	15%	15%	--------	70%
Schwingquarze	--------	80%	20%	-------

***: Verschleiß, ***: Isolation*

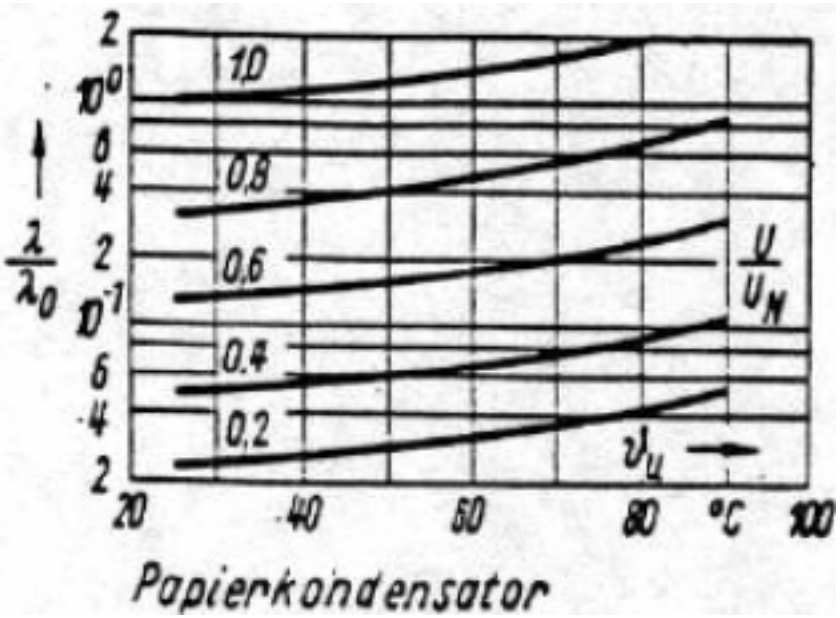
aus W. Fredrich: Zuverlässigkeit und Qualitätssicherung, Uni Rostock, 2003

Z1: Zuverlässigkeit von Bauteilen

Ausfallrate von elektronischen Bauteilen in Abhängigkeit von der Temperatur und Belastung



Kohleschichtwiderstand

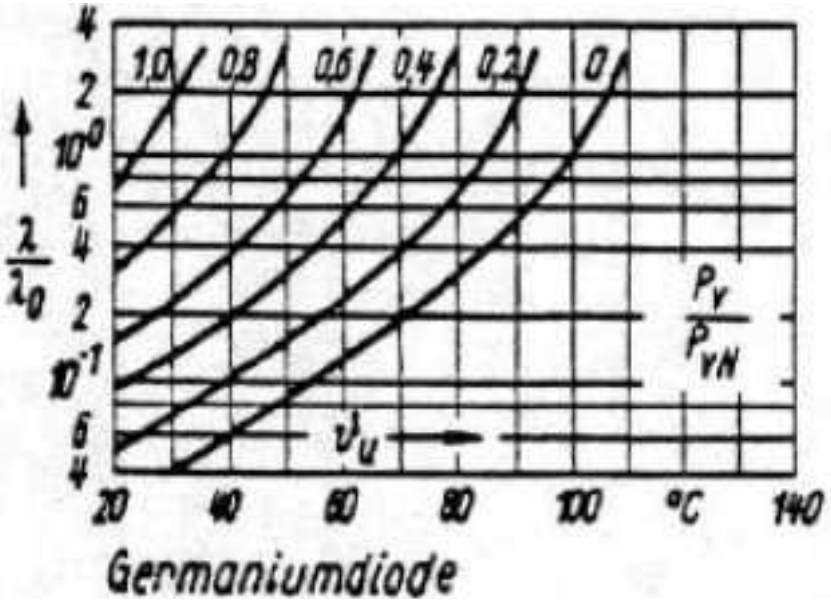


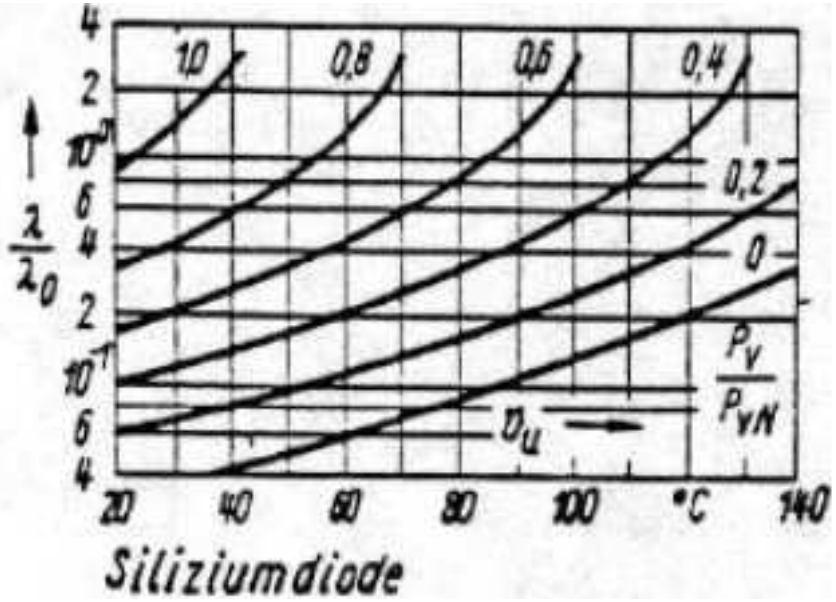
Papierkondensator

 aus W. Fredrich: Zuverlässigkeit und Qualitätssicherung, Uni Rostock, 2003

Z1: Zuverlässigkeit von Bauteilen

Ausfallrate von elektronischen Bauteilen in Abhängigkeit von der Temperatur und Belastung

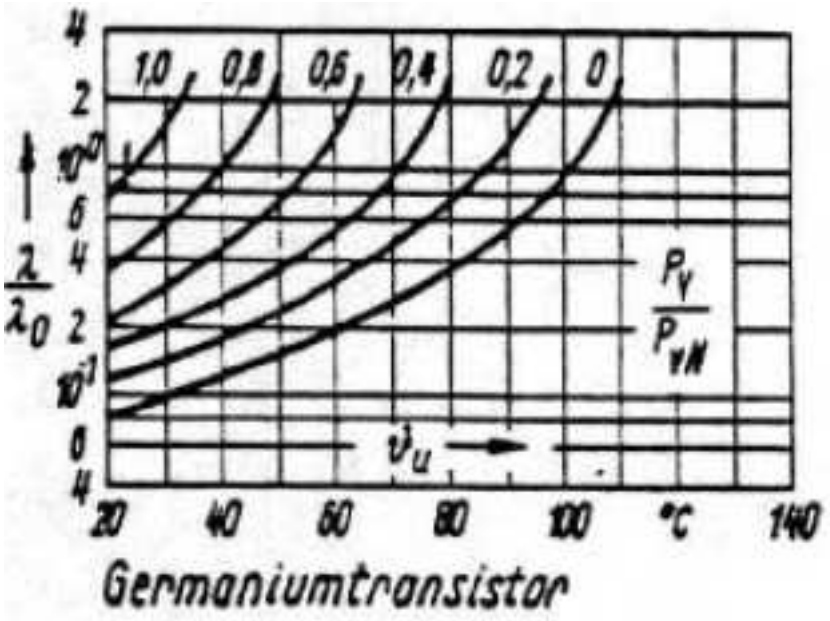


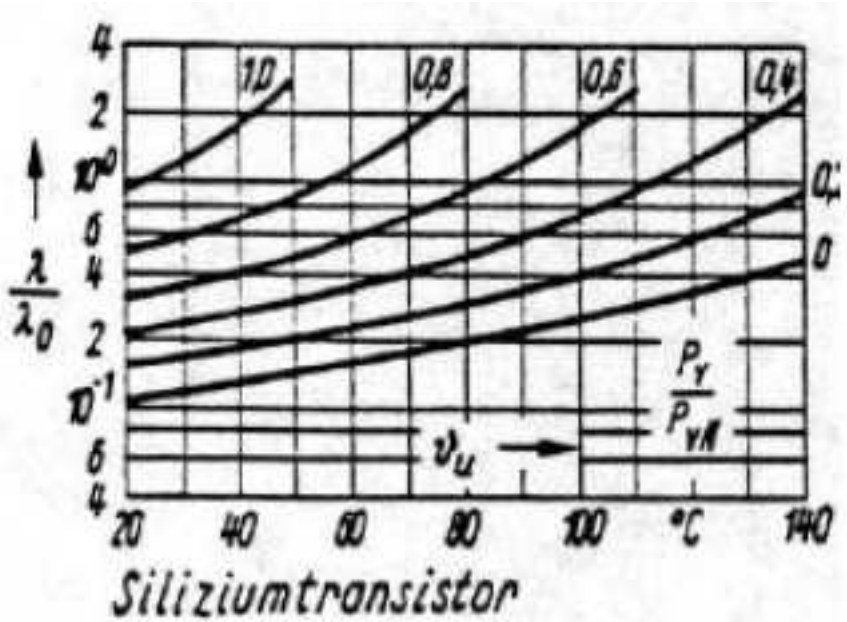


 aus W. Fredrich: Zuverlässigkeit und Qualitätssicherung, Uni Rostock, 2003

Z1: Zuverlässigkeit von Bauteilen

Ausfallrate von elektronischen Bauteilen in Abhängigkeit von der Temperatur und Belastung

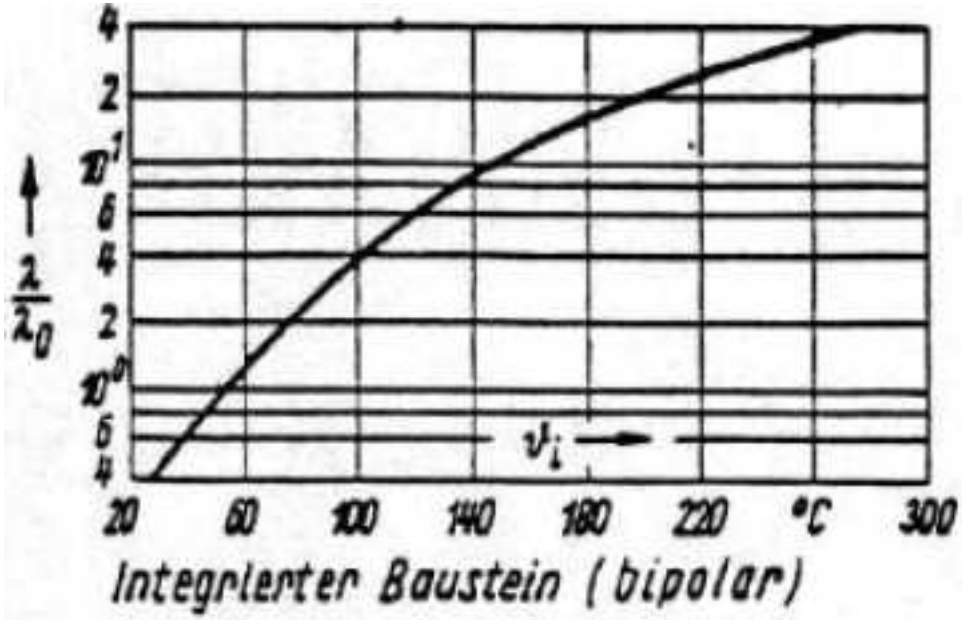


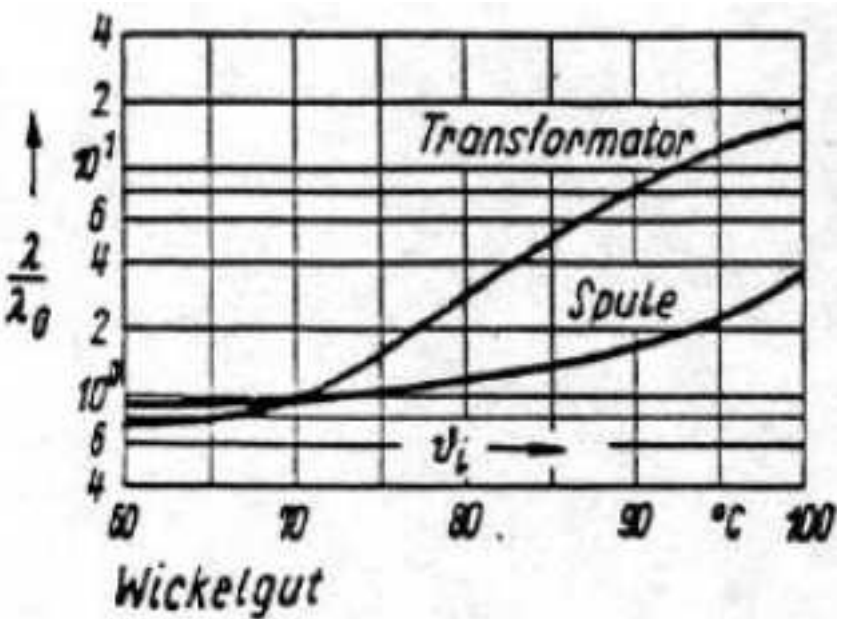


aus W. Fredrich: Zuverlässigkeit und Qualitätssicherung, Uni Rostock, 2003

Z1: Zuverlässigkeit von Bauteilen

Ausfallrate von elektronischen Bauteilen in Abhängigkeit von der Temperatur und Belastung





aus W. Fredrich: Zuverlässigkeit und Qualitätssicherung, Uni Rostock, 2003

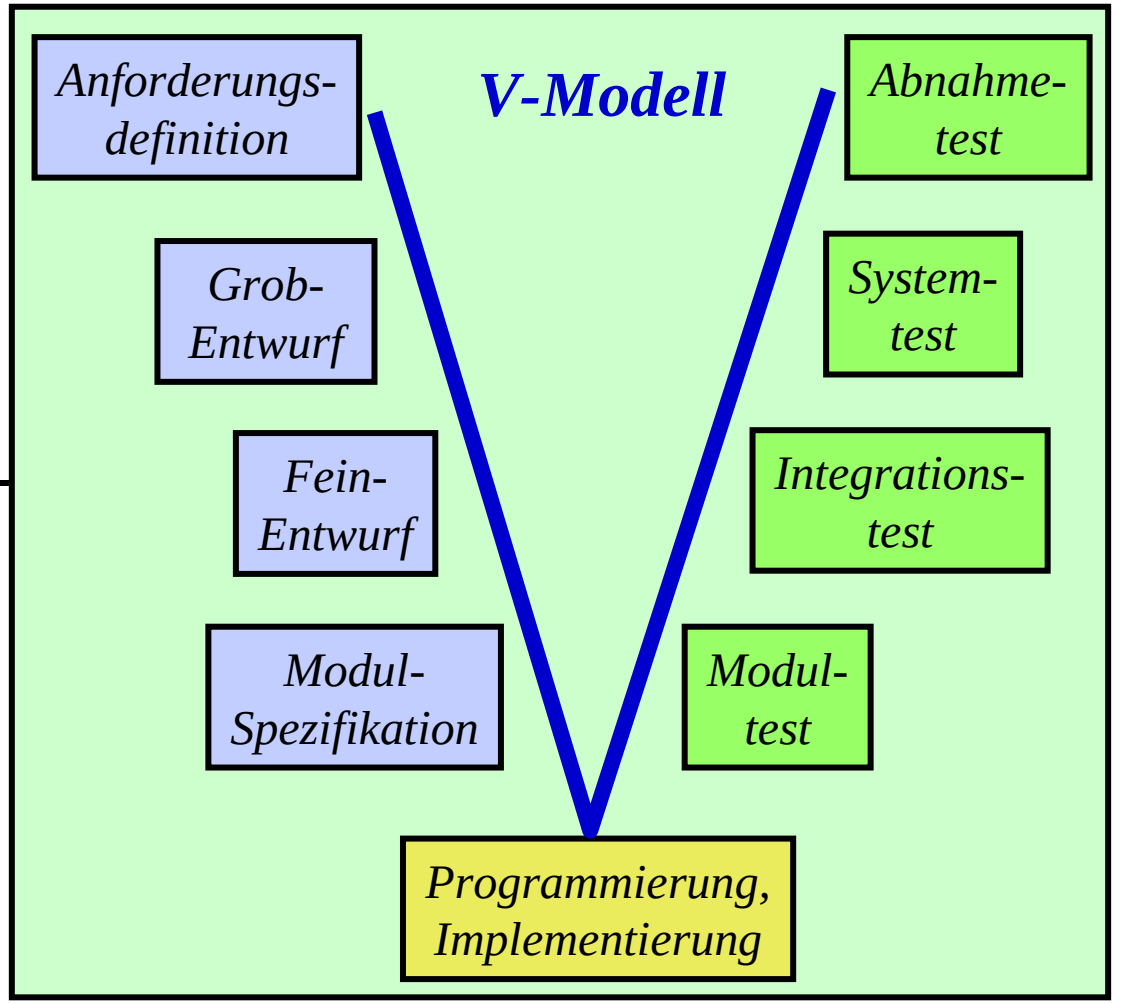
Z2: Fehler und Ausfälle

Fehler

- Fehlzustände
- Funktionsausfälle

Fehlerursachen

- Entwurfsfehler
  - Spezifikation
  - Implementierung
  - Dokumentation
- Herstellungsfehler
- Betriebsfehler
  - Störungen
  - Verschleiß
  - Zufall
  - Bedienung
  - Wartung
  - Absichtliche Eingriffe



Z2: Fehler und Ausfälle

Fehler

- Hardware – Fehler
- Software – Fehler

Zeitverlauf

- Intermittierende Fehler
- Permanente Fehler

Beispiel: Sporadische Computerfehler
SEUs (*Single Event Upsets*) / *Soft Errors*
Hauptursache: Höhenstrahlen
Höhenstrahlpartikel bringen punktuell Energie in eine mikroelektronische Struktur so ein, dass sich ein dort vorhandener binärer Zustand verändert

Einheit FIT (Failures In Time)
1 FIT: im Mittel 1 Fehler pro 10^9 Stunden

Speicherfehler (Bit kippt)
Einheit FIT pro Megabit
typisch:
1.000 bis 5.000 FIT
pro Megabit RAM-Speicher

Z2: Fehler und Ausfälle – Beispiele

- Fehlerhafte Ausgabe
- Fehlende Ausgabe, Stop

- Leitungsbruch
- Leitungsstörung
- Nachrichtenverlust
- Nachrichtenverfälschung
- Nachrichten-Fehlleitung
- Nachrichten-Duplizierung

- Entwurfsfehler, Spezifikationsfehler
- Programmierfehler
- Dokumentationsfehler

- Hardwareentwurfsfehler
- Chip-Fehlstelle
- Chip-Leitungsschluss
- Chip-Leitungsunterbrechunng

- Sabotage (kein Fehler im engeren Sinn)

- Herstellungsfehler: Exemplarstreuung
- Herstellungsfehler: Compilerfehler

- Äußere Einflüsse
  - mechanisch
  - elektrisch
  - thermisch
  - Strahlung
  - magnetisch
  - elektromagnetisch
- Plattencrash

- Verschleißfehler
- Alterung
  - Gebläse-Ausfall, Überhitzung

- Wartungsfehler
  - „Kaputtreparatur“ (HW u. SW)
- Fehlbedienung

Z2: Fehler und Ausfälle – Fehlerorte

Fehler und Ausfälle

- Ausfall:
  Ereignis, Zustandsübergang
- Fehler:
  Systemzustand

- *Fehler ohne Ausfall ?*
  - *Ja: Latenter Fehler*
    Latenzdauer = Zeit bis zum Ausfall

- *Ausfall ohne Fehler ?*
  - *nicht möglich*

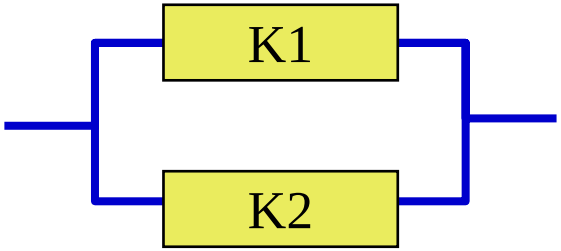
Orte

- Anwendungssoftware
- Middleware
- Betriebssystem
- Kommunikationssystem

- Hardware
- Peripherie

Z3: Zuverlässigkeitsblockdiagramm

- Knoten: Komponente
- Kanten: Anordnung von Komponenten, parallel und in Serie, entsprechend Beitrag zur gewünschten Systemfunktion

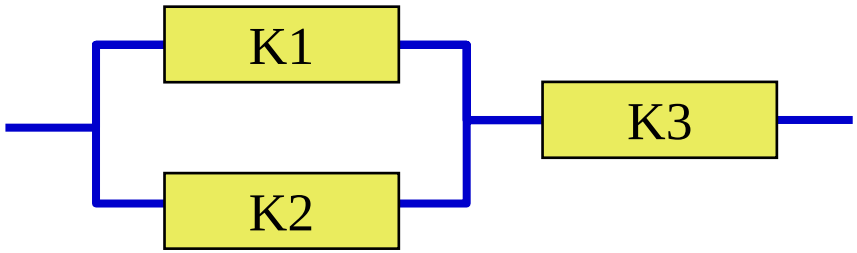


Parallelschaltung
wenn beide Komponenten unabhängig ausfallen:
$p(Ausfall_{ges}) = p(Ausfall_{K1}) * p(Ausfall_{K2})$

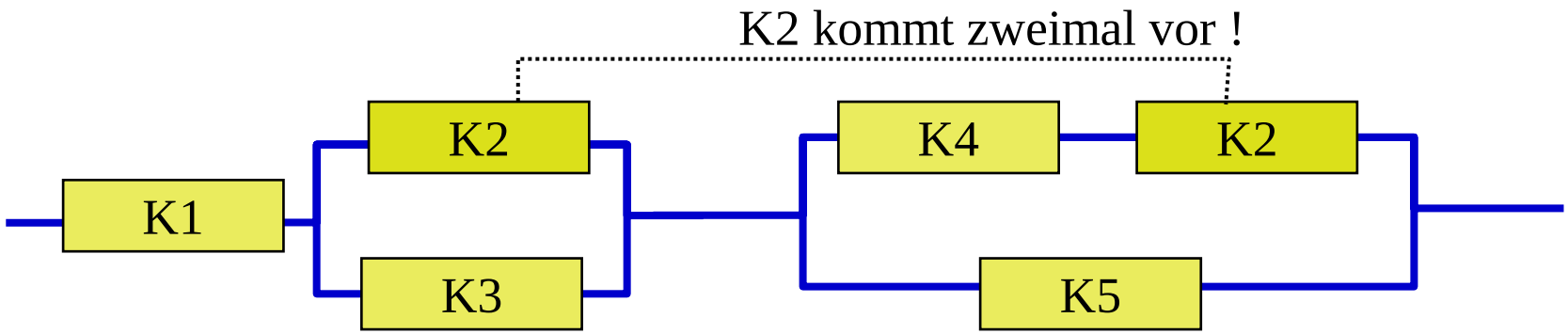
Serienschaltung
wenn beide Komponenten unabhängig ausfallen:
$p(Funktion_{ges}) = p(Funktion_{K1}) * p(Funktion_{K2})$

Z3: Zuverlässigkeitsblockdiagramm

- Komplexere Strukturen



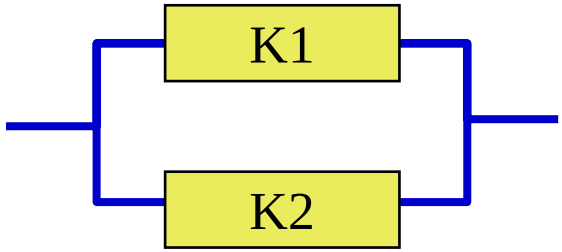
Serien-Parallelschaltung



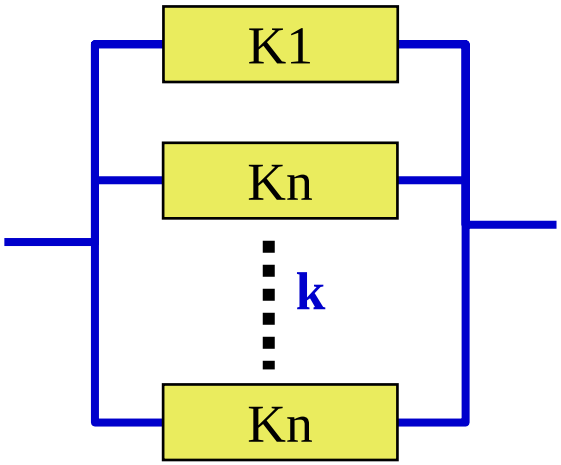
Mehrfach vorkommende Komponenten

Z3: Zuverlässigkeitsblockdiagramm

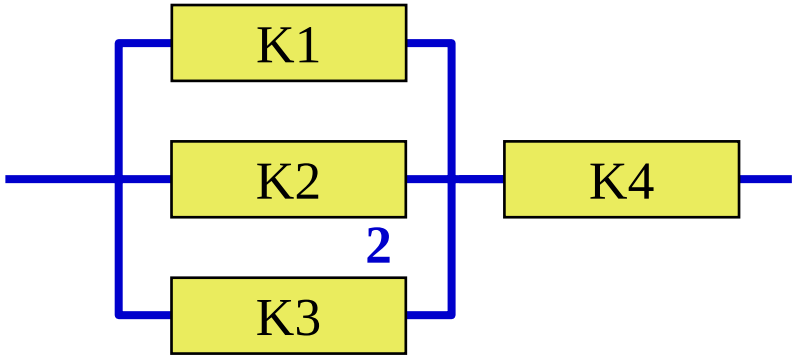
- Komplexere Strukturen



Redundanz 1-aus-2



Redundanz k-aus-n



Redundanz 2-aus-3 mit Entscheider K4

Z4: Struktur–Funktionsmodell

nicht-formal: Adressat Mensch

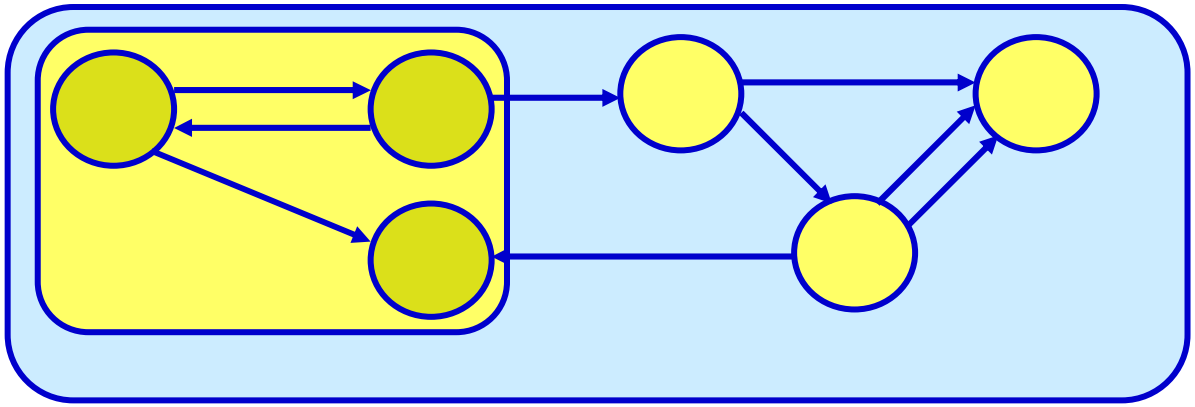
- *abstrakte System–Sicht*
- *Verständnis der Systemzusammenhänge*

- Struktur des Systems
- Funktionen des Systems und Zuordnung
- Abhängigkeiten

➔ Bereiche der Fehlerentstehung

Bereiche der Fehlerausbreitung

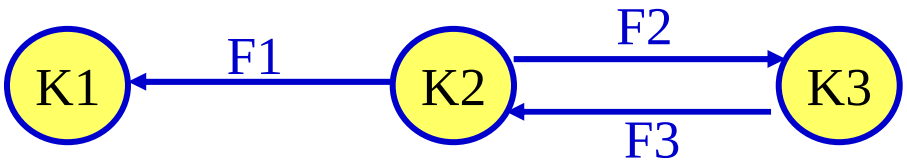
Wirkung der eingesetzten Fehlertoleranz-Verfahren



Z4: Struktur–Funktionsmodell: Graph

- Knoten: ***Komponenten***

- Kanten (gerichtet): ***Funktionen sowie Zuordnungen zu Erbringer und Nutzer***

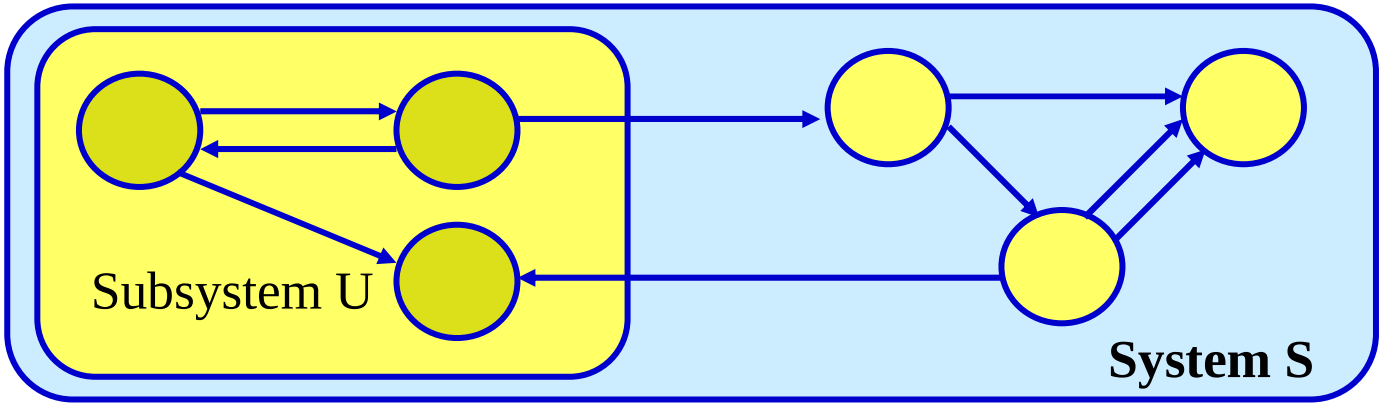


  - K2 erbringt F1 für K1 und F2 für K3
  - K3 erbringt F3 für K1

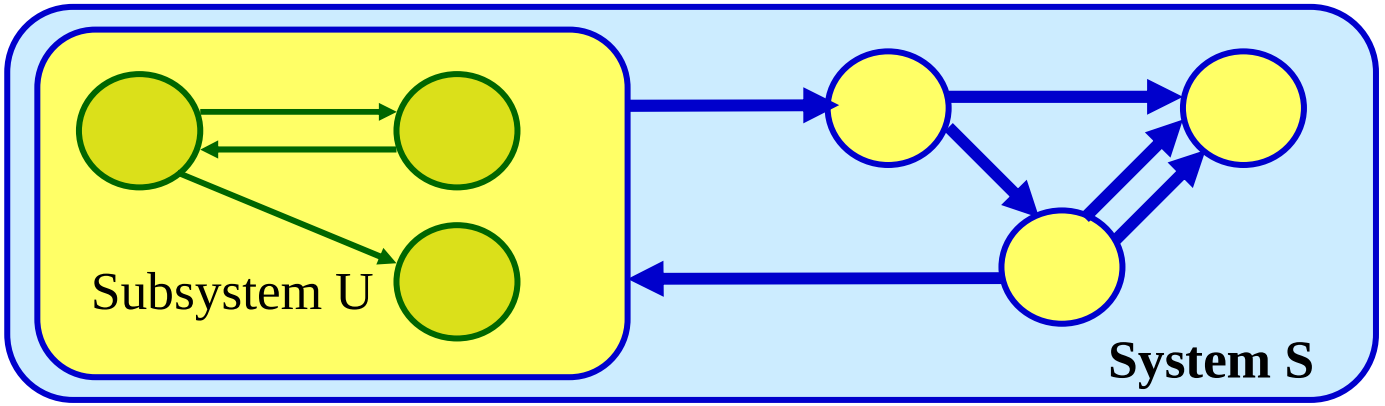
- ***Prinzip:***
  Funktionsausfall geht immer auf Fehlzustand der erbringenden Komponente zurück

Z4: Struktur–Funktionsmodell

- Zusätzlich: Strukturierung in Subsysteme möglich
  - A] Als Komponenten-Sammlungen

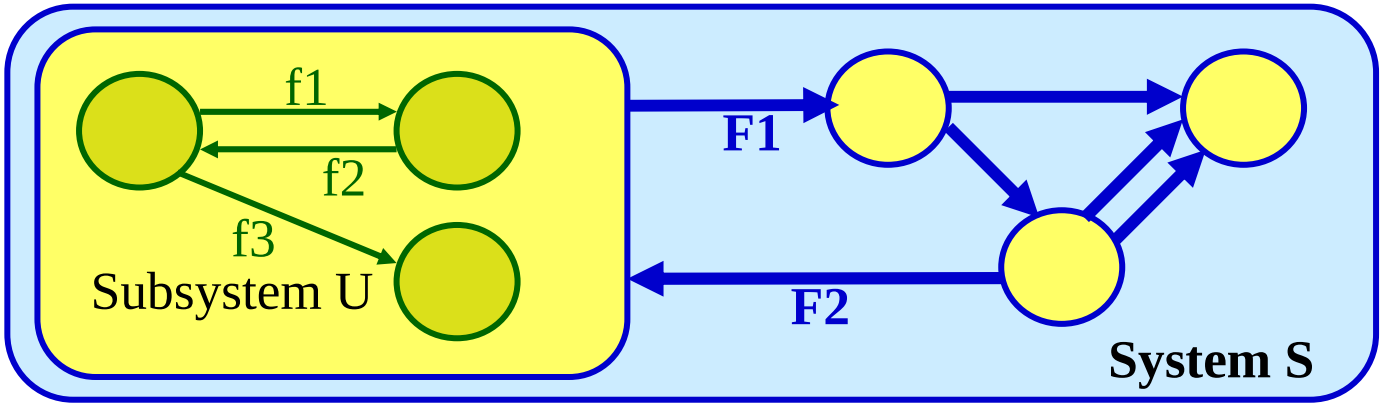


  - B] Als zusammenfassende Abstraktion der inneren Komponenten



Z4: Struktur–Funktionsmodell

B] Als zusammenfassende Abstraktion der inneren Komponenten



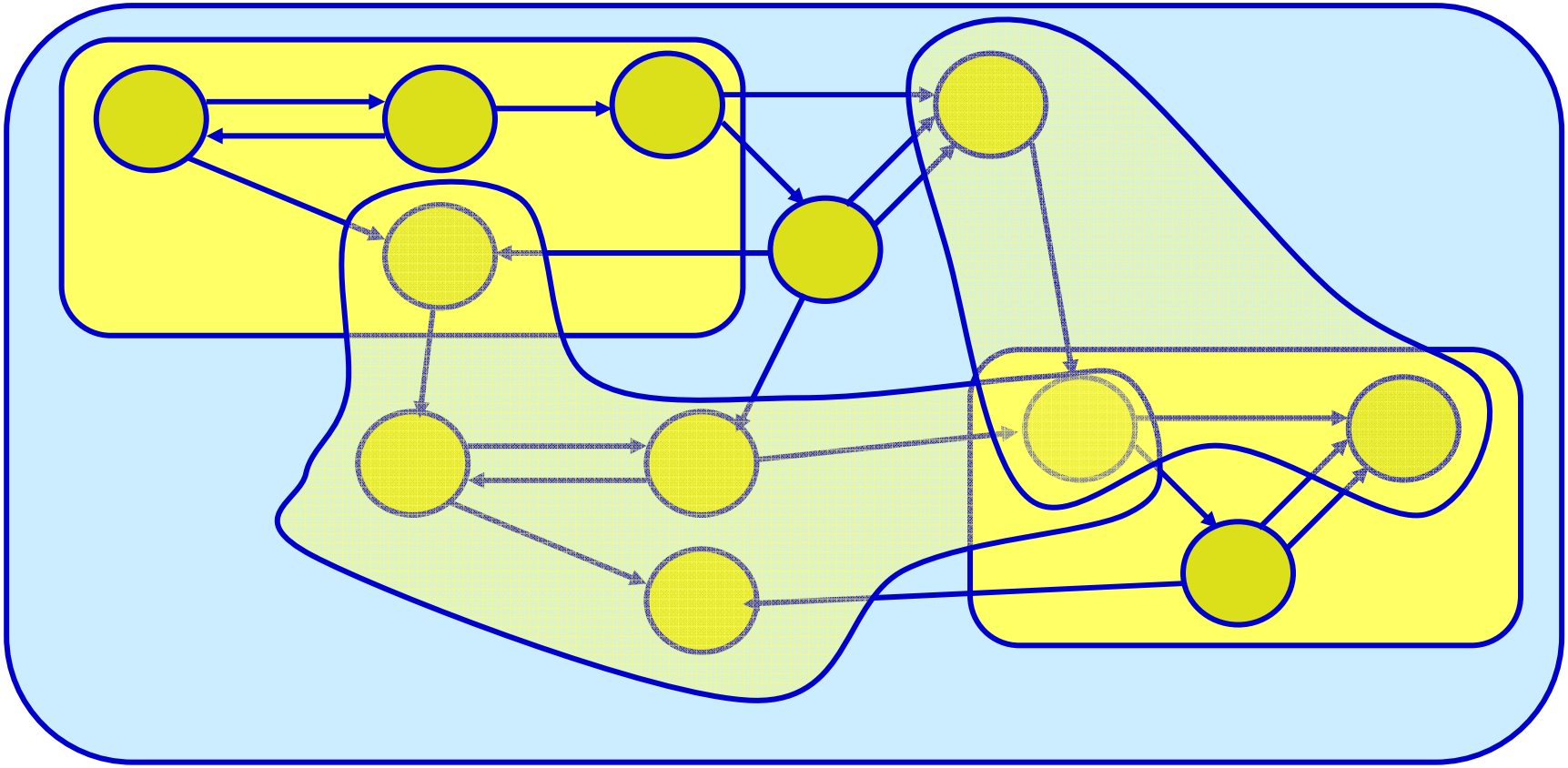
– ***Innere Spezifikation*** eines Subsystems:
  Komponentenstruktur und interne Funktionszuordnungen

– ***Äußere Spezifikation*** eines Subsystems:
  an den Subsystemgrenzen von außen benötigten
  sowie die nach außen erbrachten Funktionen

Z4: Struktur–Funktionsmodell

C] Weitere Strukturierungen

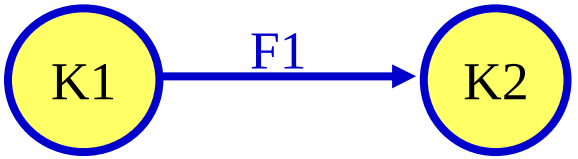
- Überlappende Subsysteme



Komplexes ist komplex – aber, Abstraktion und Übersicht sind nötig

Z4: Struktur–Funktionsmodell

- **K1 erbringt Funktion F1 für K2**
  - K1 ist Bestandteil von K2
  - K1 stellt K2 Betriebsmittel zur Verfügung
  - K1 kann von K2 aufgerufen werden
  - K1 bietet K2 einen Dienst an

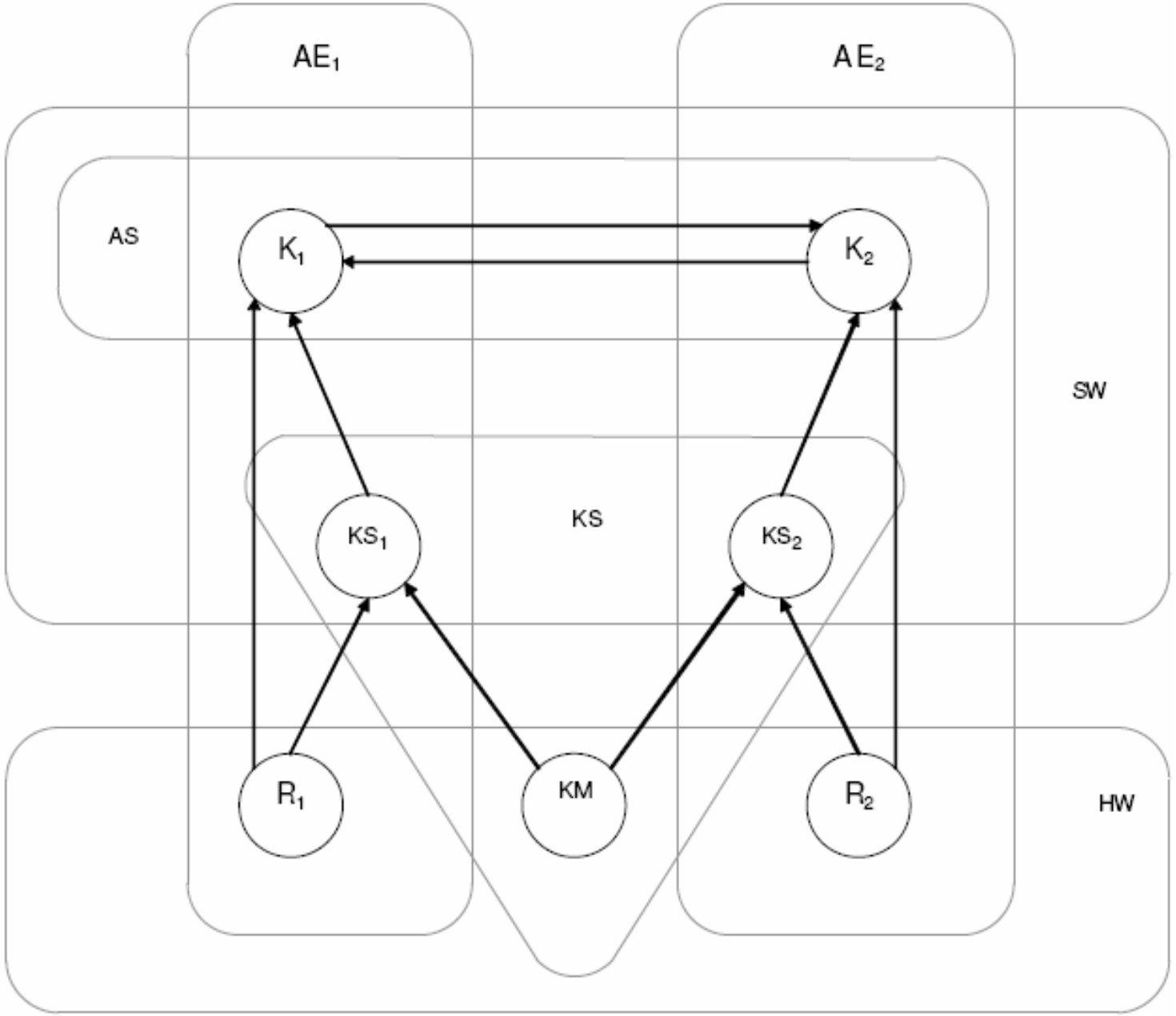


- **Statische und dynamische Systemstrukturen**
  - Komponenten erzeugen und vernichten
  - Komponenten (Dienste) suchen, binden und freigeben

- **Unterschiedliche Komponenten--Arten**
  - Software-Prozess, Software-Modul, Software-Funktion, Services und Operationen
  - Betriebssystem, Betriebssystem-Komponente
  - Hardware, Hardware-Komponente, Prozessor-Zeitscheibe, permanenter Speicher

Z4: Struktur–Funktionsmodell

Beispiel aus K. Echtle: Fehlertoleranzverfahren



Hardware
HW = {R1, KM, R2},

Software
SW = {K1, K2,
KS1, KS2}

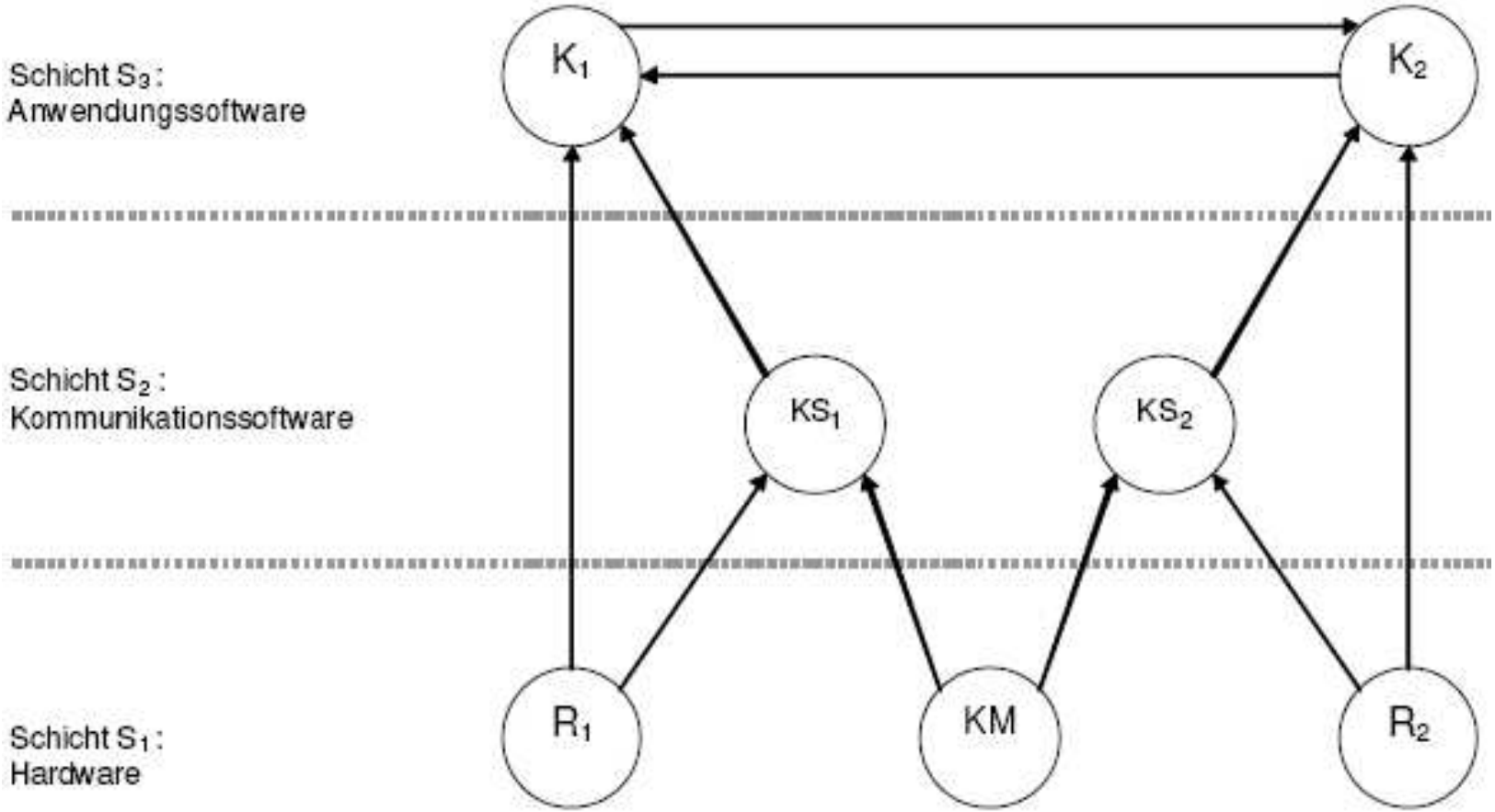
Anwendungssoftware
AS = {K1, K2}

Kommunikationssystem KS = {KS1, KM, KS2}

Erste Ablaufeinheit
AE1 = {K1, KS1, R1}

Zweite Ablaufeinheit
AE2 = {K2, KS2, R2}.

Z4: Struktur–Funktionsmodell: Schichtenmodell



Schicht

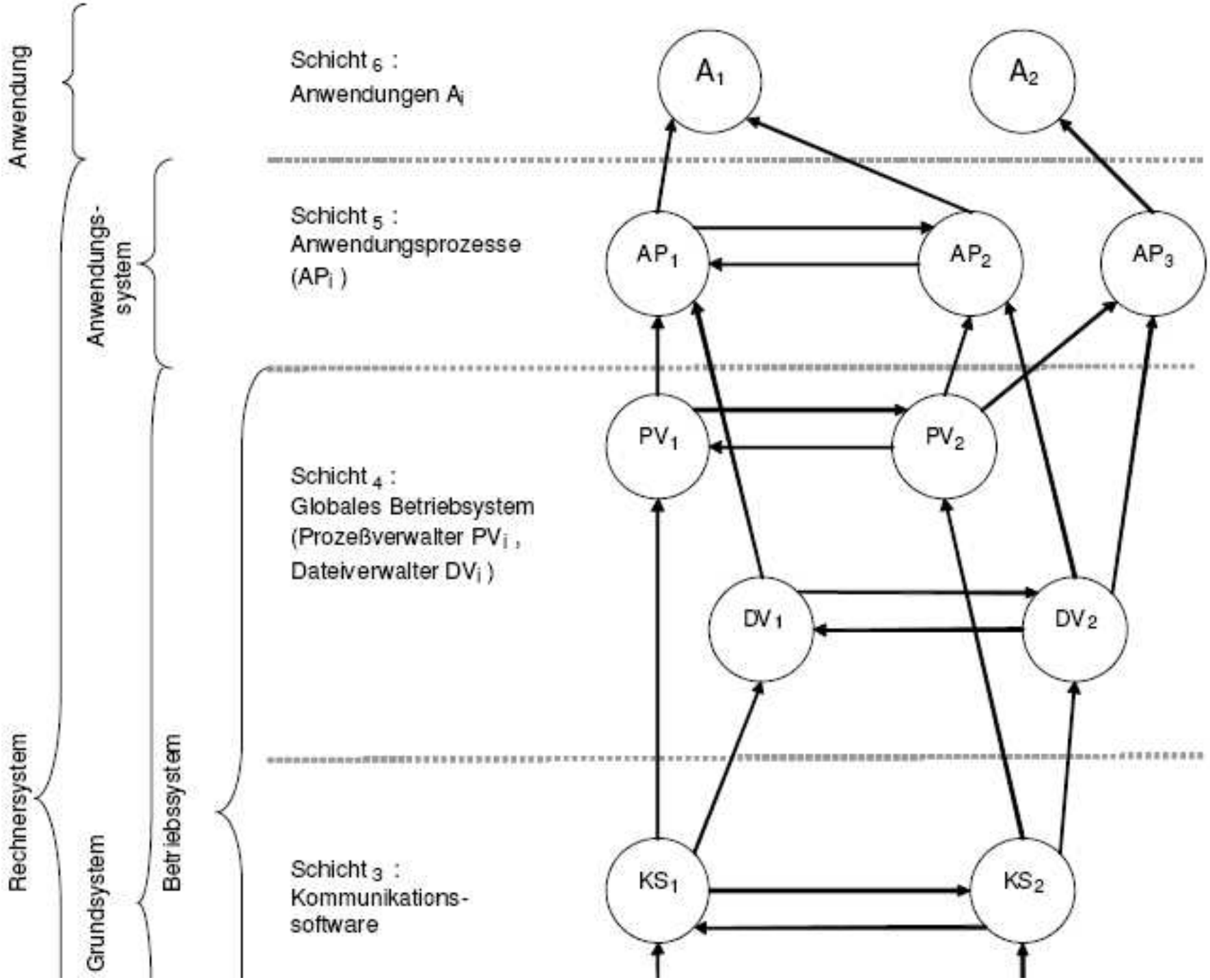
- *Subsystem*

Halbordnung zwischen den Schichten

Funktionszuordnungen

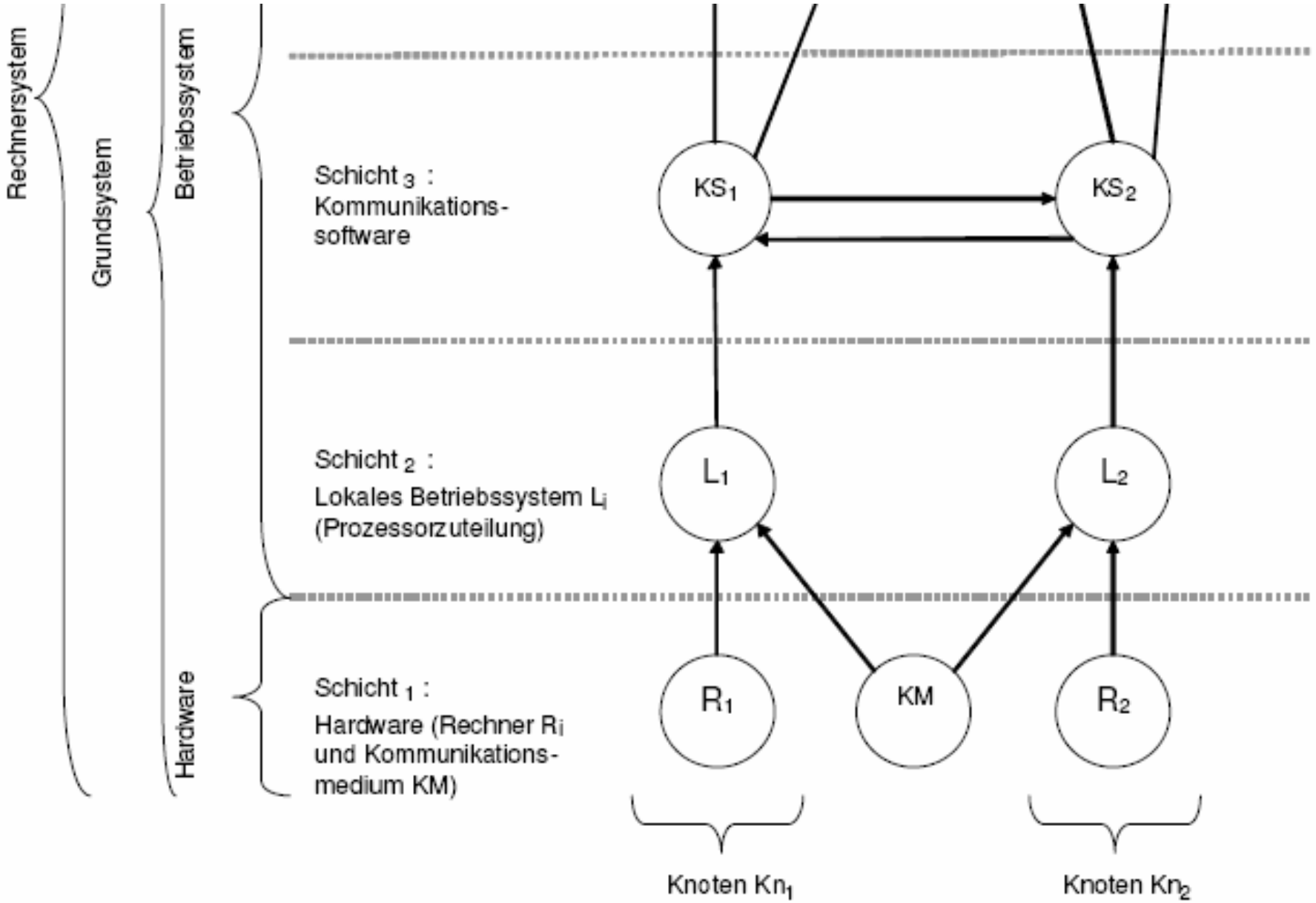
- *innerhalb von Schichten sowie nur von niedrigeren an höhere Schichten*

Beispiel aus K. Echtle: Fehlertoleranzverfahren



Beispiel aus K. Echtle: Fehler-toleranz-verfahren

Zwei-Rechner-System

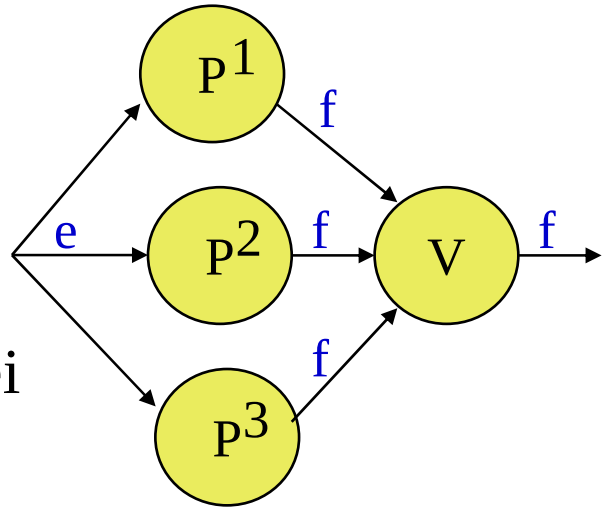


Beispiel aus K. Echtle: Fehler-toleranz-verfahren

Zwei-Rechner-System

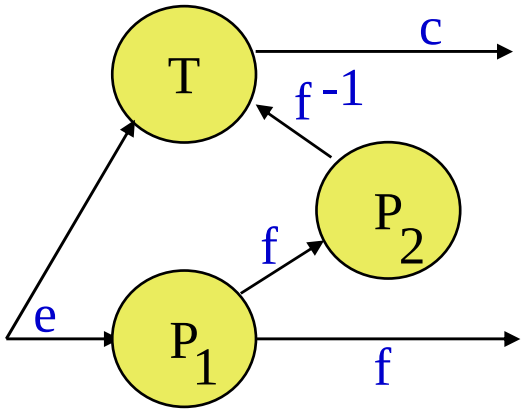
Z4: Struktur–Funktionsmodell: Spezielle Komponentenarten

- **Replikate**
  Vervielfältigte Komponenten mit gleicher Funktion
  Replizierte Komponente
  Exemplar
  z.B. P^1, P^2, P^3 *(hochgestellte Indizes)* als die drei Prozessexemplare eines 2-von-3-Systems



- Replikate mit veränderter Implementierung
  diversitäre Exemplare oder *Varianten*

- **Komponentenmengen** aus **ungleichen** Komponenten
  Module
  z.B. Modul aus Funktions-, Umkehrfunktions- und Vergleichskomponente zur Fehlerdiagnose



Z5: Fehlermodell und Ausbreitung: Fehlermodell

Spezifikation der Fehlertoleranz-Fähigkeit eines Systems benötigt:

- *Fehlervorgabe*
  - *Menge der zu tolerierenden Fehler* (bezogen auf ein formales *Fehlermodell*)
- *Fehlermodell*
  - definiert die betrachteten Fehlermöglichkeiten eines Systems als Obermenge der *Menge der zu tolerierenden Fehler*
  - muss die von einem Fehler betroffenen Komponenten nennen (Ort: ***strukturelle Fehlerbetrachtung***)
  - muss angeben, in welcher Weise deren Funktion beeinträchtigt wird (Art: ***funktionelle Fehlerbetrachtung***)
- *Komponentenbezogenes strukturelles Fehlermodell* unterscheidet häufig für jede Komponente nur "fehlerfrei" von "fehlerhaft“:
  - ***Binäres Fehlermodell***
    System **S**, Fehlermodell **FS** als Funktion
    FS: {K : K ist Komponente des Systems S} x Zeit → {**fehlerfrei, fehlerhaft**}

Z5: Fehlermodell und Ausbreitung: Fehlerbereiche

Annahme, dass zu tolerierende Fehler nur in bestimmten Komponentenmengen auftreten.

- ***Fehlerbereich***
  Komponentenmenge einer zugeordneten Fehlermenge
- ***Fehlerbereichs-Annahme***
  Zuordnung der zu tolerierenden Fehler zu Fehlerbereichen
  - ein Fehler f bleibt stets auf die von f zugeordnete Komponentenmenge beschränkt, z. B. weil Fehler in den einzelnen Fehlerbereichen unabhängig voneinander entstehen und sich nicht auf weitere Komponentenmengen ausbreiten
- ***Fehlerfall***
  Aussage, in welchem der *Fehlerbereiche* aktuell Fehler aufgetreten sind

- **Grundannahme**
  Kein Fehlerbereich umfasst das gesamte System
  (Andernfalls könnten sich Fehler auf sämtliche Komponenten auswirken und man könnte sie nicht tolerieren).

Z5: Fehlermodell und Ausbreitung: Fehlerbereiche

Fehlerbereiche sind nicht unbedingt disjunkte vollständige Zerlegung

- Überschneidende Bereiche
  $\exists$ K, i, j: K$\in$ FB_i $\wedge$ K$\in$ FB_j
- Perfektionskern
  $FB_1 \cup FB_2 \cup .. \cup FB_n \neq S$
  Perfektionskern = $S \setminus (FB_1 \cup FB_2 \cup .. \cup FB_n)$

Einzelfehlerbereiche

- Größeres System mit höherer Anzahl von zu tolerierenden Fehlern
  ➔ *Fehlerbereiche müssen* alle Kombinationen von zulässigen "Fehlerstellen" beschreiben
  Viele Kombinationsmöglichkeiten
- Stattdessen disjunkte Zerlegung der Vereinigung aller Fehlerbereiche in Einzelfehlerbereiche
  - Jede maximal große Komponentenmenge, bei der alle Komponenten zu genau den gleichen Fehlerbereichen gehören, ist ***Einzelfehlerbereich***

Z5: Fehlermodell und Ausbreitung: Einzelfehlerbereiche

Beispiel: 3-Rechner-System

- Prozesse P1, P2, P3, P4 und P5, die
- Lokale Betriebssysteme BS1, BS2 und BS3
- Rechnerhardware R1, R2 und R3

Bereichsgliederung

- einander zugeordnete Hardware und Software sind wegen der Ausbreitung von Hardwarefehlern auf die Software zusammengefasst
- auch ohne Hardwarefehler kann es Softwarefehler in P2, P3 und P4 geben

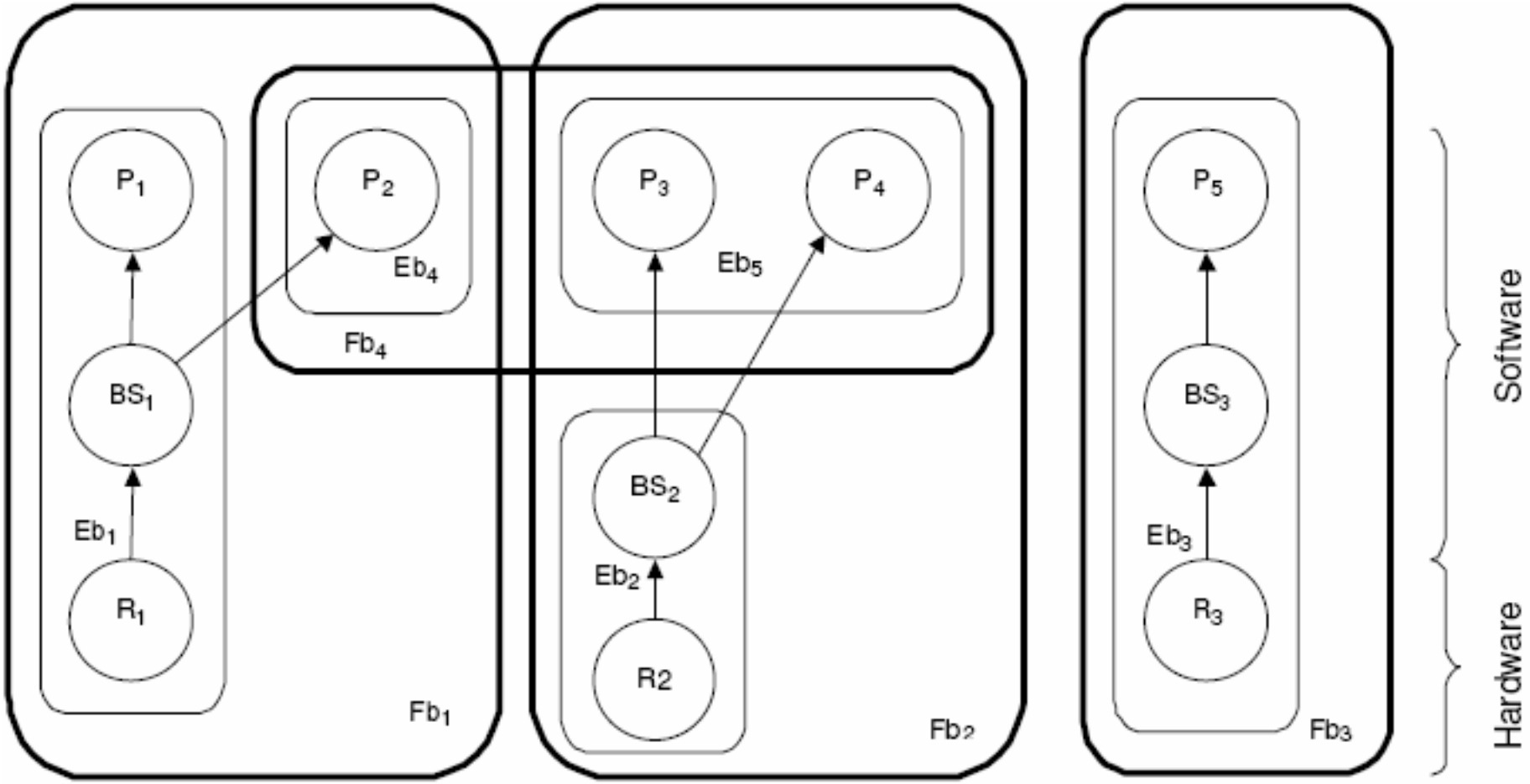
Menge der zu tolerierenden Fehler, formuliert anhand der ***Fehlerbereiche***:

- Fb = {Fb1, Fb2, Fb3, Fb4}
  = {{P1, P2, BS1, R1}, {P3, P4, BS2, R2}, {P5, BS3, R3}, {P2, P3, P4}}

Daraus resultierende ***Einzelfehlerbereiche***

- Eb = {Eb1, Eb2, Eb3, Eb4, Eb5}
  = {{P1, BS1, R1}, {BS2, R2}, {P5, BS3, R3}, {P2}, {P3, P4}}.

Z5: Fehlermodell und Ausbreitung: Einzelfehlerbereiche



3-Rechner-System
- R: Hardware
- BS: Betriebssystem
- P: Anwendungsprozesse

dick umrandet: Fehlerbereiche
dünn umrandet: Einzelfehlerbereiche

Beispiel aus K. Echtle: Fehlertoleranzverfahren

Z5: Fehlermodell und Ausbreitung: k-Fehlerannahme

Zur Vereinfachung bei großer Zahl von Fehlerbereichen als
Spezialfall der *Fehlerbereichs-Annahme:*
k-Fehler-Annahme mit ***Zeitredundanz t_R*** und ***Mindestanzahl n***

- Disjunkte Zerlegung eines Systems S in
  Einzelfehlerbereiche $Eb_1, \ldots, Eb_e$
  (mit $Eb_1 \cup \ldots \cup Eb_e = S$)

- k-Fehler-Annahme
  Forderung alle Fehler zu tolerieren, die sich auf bis zu **k** Einzelfehlerbereiche erstrecken
- Zeitredundanz $\mathbf{t_R}$
  Zu jedem Zeitpunkt dürfen höchstens k Einzelfehlerbereiche Fehler aufweisen oder in der durch $\mathbf{t_R}$ begrenzten Vergangenheit fehlerhaft gewesen sein
- Mindestanzahl **n**
  Bei permanenten Fehlern können im Zuge der Fehlerbehandlung Komponentenmengen ausgegliedert werden.
  Die Fähigkeit, trotz zuvor aufgetretener und erfolgreich behandelter Fehler weitere zu tolerieren, wird solange gefordert, wie noch eine Mindestanzahl n an Einzelfehlerbereichen im System S vorhanden sind

Z5: Fehlermodell und Ausbreitung: k-Fehlerannahme

Zur Vereinfachung bei großer Zahl von Fehlerbereichen als Spezialfall der *Fehlerbereichs-Annahme:*
k-Fehler-Annahme mit ***Zeitredundanz*** t_R und ***Mindestanzahl n***

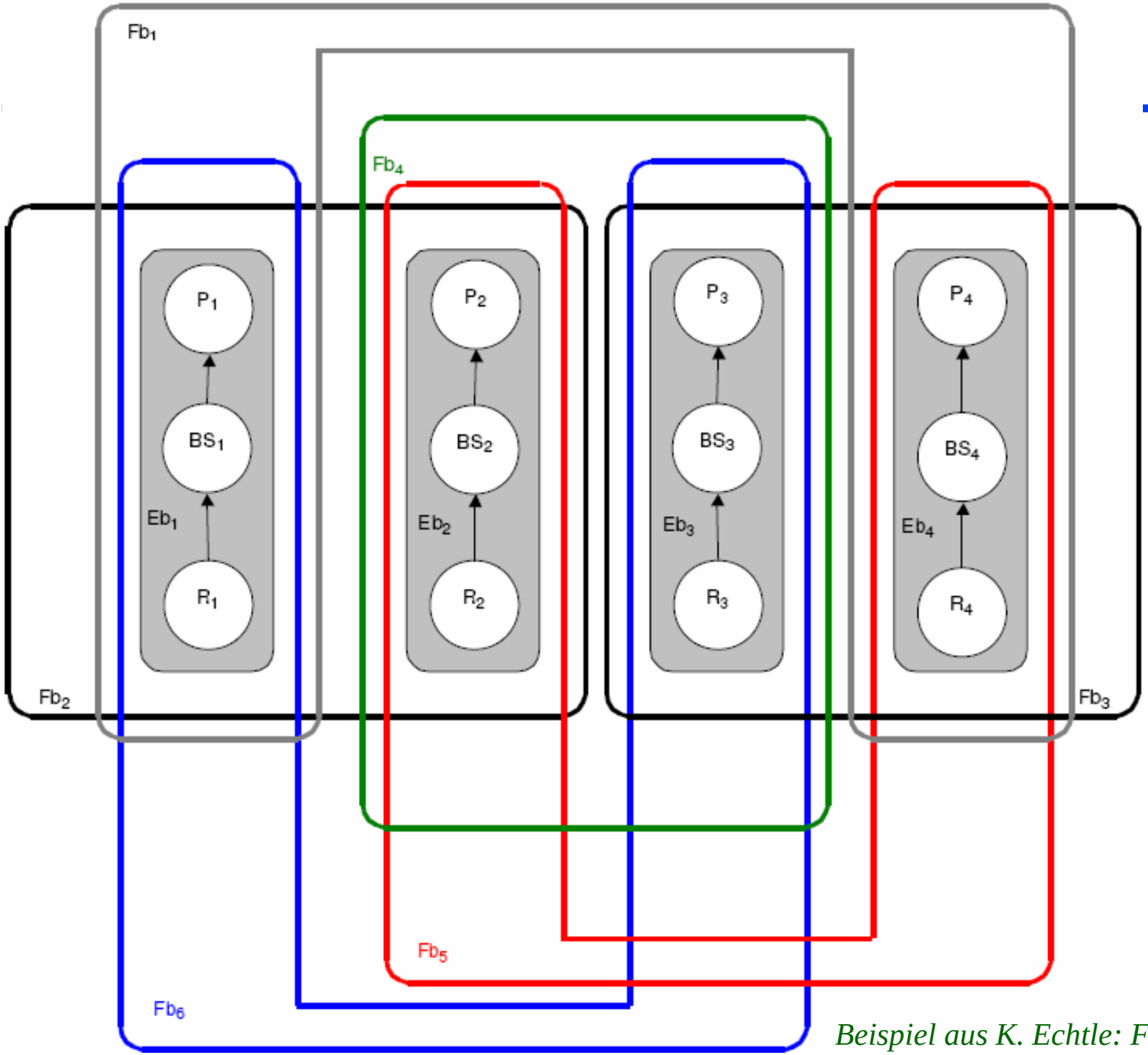
- Disjunkte Zerlegung eines Systems S in
  Einzelfehlerbereiche $Eb_1, \ldots, Eb_e$
  (mit $Eb_1 \cup \ldots \cup Eb_e = S$)

- Die *k-Fehler-Annahme* läßt sich auf die *Fehlerbereichs-Annahme* zurückführen, indem jede Vereinigungsmenge beliebiger k Einzelfehlerbereiche als eigener Fehlerbereich definiert wird:

$$Fb = \{Eb_{i1} \cup \ldots \cup Eb_{ik} : i_1, \ldots, i_k \in \{1, \ldots, e\}\}$$

  - Kombinatorik: Aus e Einzelfehlerbereichen entstehen

$$\binom{e}{k} \text{ Fehlerbereiche}$$



Beispiel:

- **4-Rechner-System mit 2-Fehler-Annahme**

Einzelfehlerbereiche und Fehlerbereiche: *Alle 6 Kombinationen von je 2 Einzelfehlerbereichen sind zu je einem Fehlerbereich zusammengefasst.*

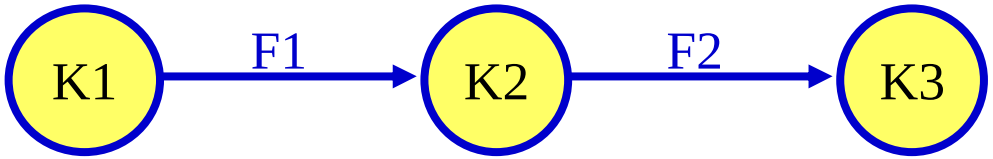
Beispiel aus K. Echtle: Fehlertoleranzverfahren

Z5: Fehlermodell und Ausbreitung: Fehlfunktionsannahme

Eine zusätzlich zur Fehlerbereichs-Annahme getroffene **Fehlfunktions-Annahme**, dient der Detaillierung der Fehlervorgabe

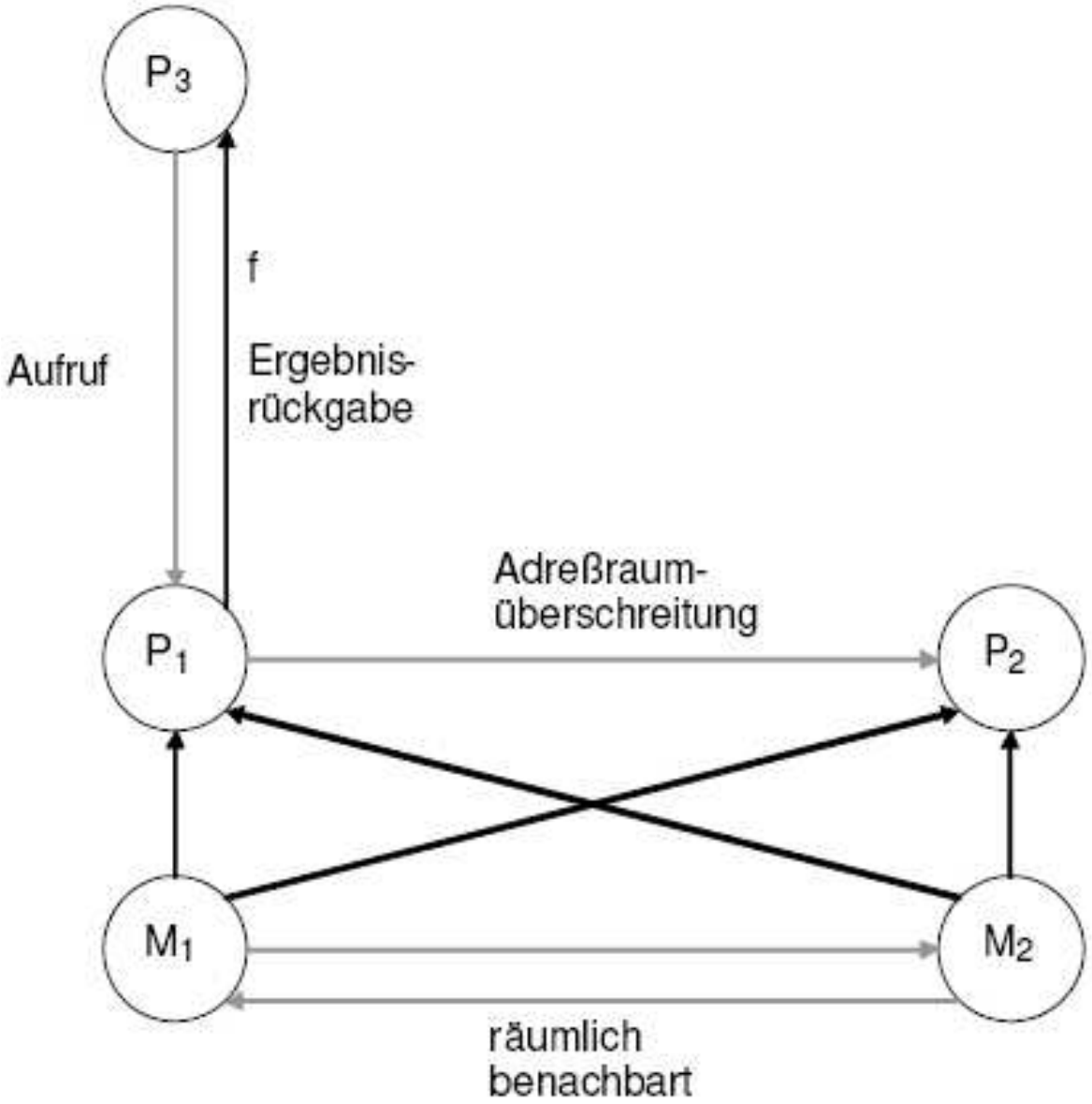
- *bestimmte Fehlermöglichkeiten sind u.U. allzu unwahrscheinlich*
- *Fehler, die nur mit allzu hohem Aufwand zu tolerieren wären, werden aus der Fehlervorgabe ausgenommen*
- ***Beispiele***
  - Teilausfall *(bestimmte Teilfunktionen fallen aus)*
  - Unterlassungsausfall *(eine Operation wird insgesamt nicht ausgeführt)*
  - Anhalte-Ausfall *(die Komponente stoppt)*
  - Haft-Ausfall *(die Komponente gibt immer dasselbe aus)*
  - Inkonsistenz-Ausfall *(durch Plausibilitätstest erkennbarer Ausfall)*
  - k-Binärstellen-Ausfall *(maximal k-Stellen sind verfälscht)*

Z5: Fehlermodell und Ausbreitung: Fehlerausbreitung



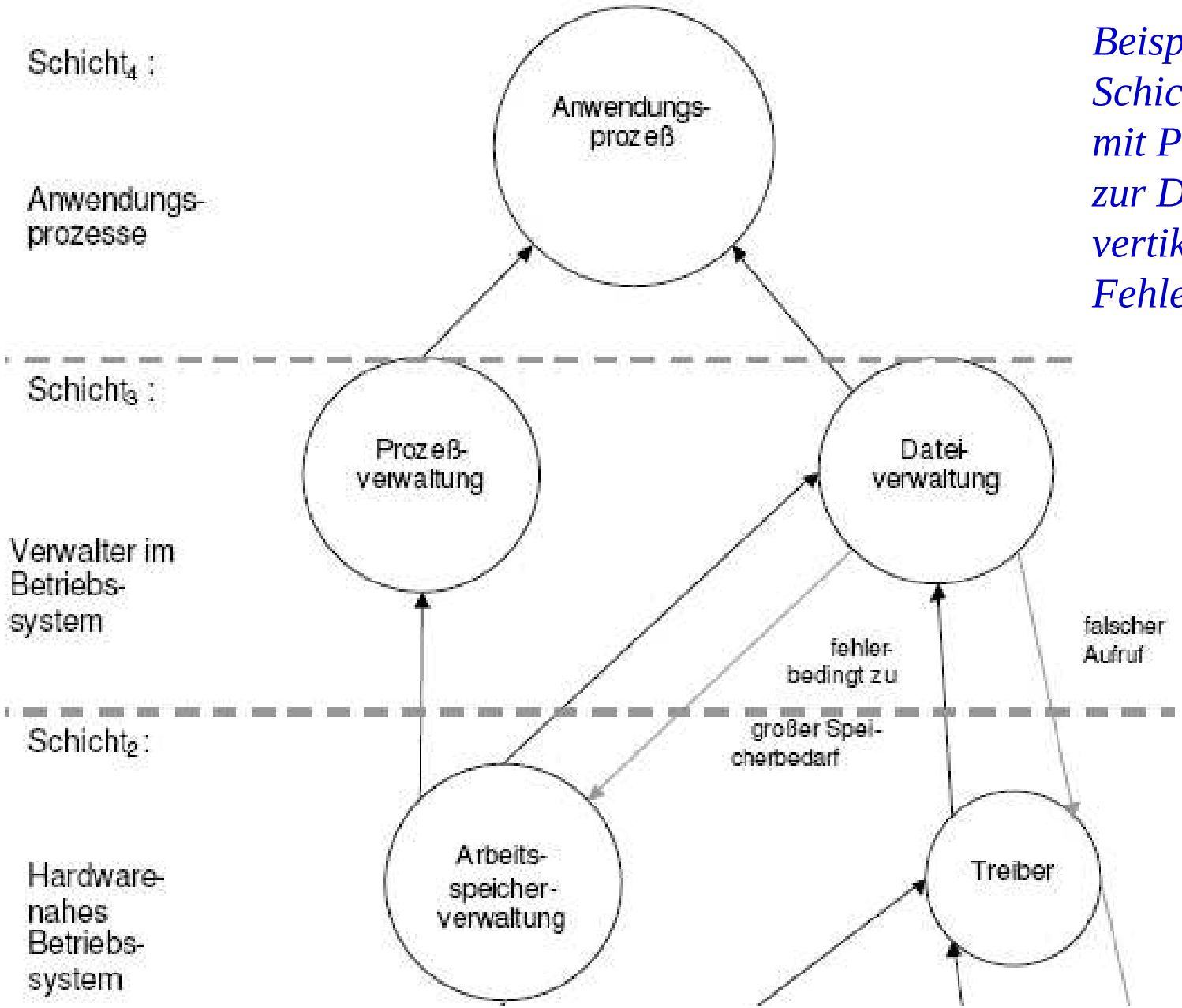
- Fehler können sich über Funktionen ausbreiten
  Wenn z.B. K1 ausfällt ist, kann F1 fehlerhaft sein und K1 ausfallen, so dass F2 fehlerhaft wird und K3 ausfällt
- Fehler können weiterhin über sich zusätzliche Wege ausbreiten
  - Leitungen sprechen über
  - Ein Prozess greift aufgrund eines Programmierfehlers in den Adressraum eines anderen Prozesses
  - Zwei Hardware-Module sind räumlich so eng benachbart, dass eine Überhitzung des einen Moduls auch zur Überhitzung des anderen führt
- Darstellung im Struktur-Funktionsmodell
  - **Pseudofunktionen**
    - » z.B. eng benachbart, Übersprechen, Adressraum-Übergriff, etc.

Z5: Fehlermodell und Ausbreitung: Fehlerausbreitung



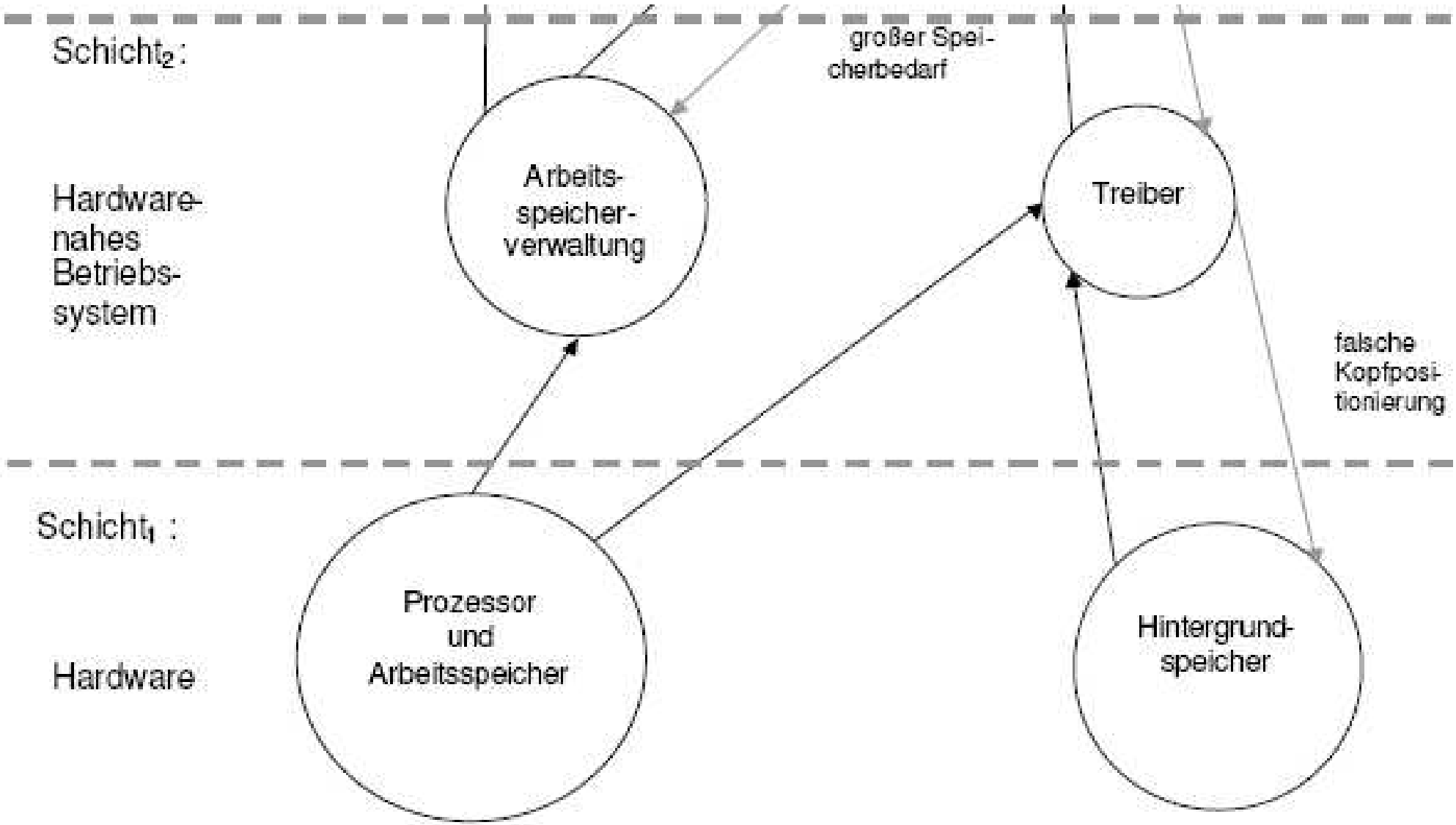
- Funktionen
- Pseudofunktionen

Beispiel aus K. Echtle: Fehlertoleranzverfahren



Beispiel eines Schichtenmodells mit Pseudofunktionen zur Darstellung der vertikalen Fehlerausbreitung

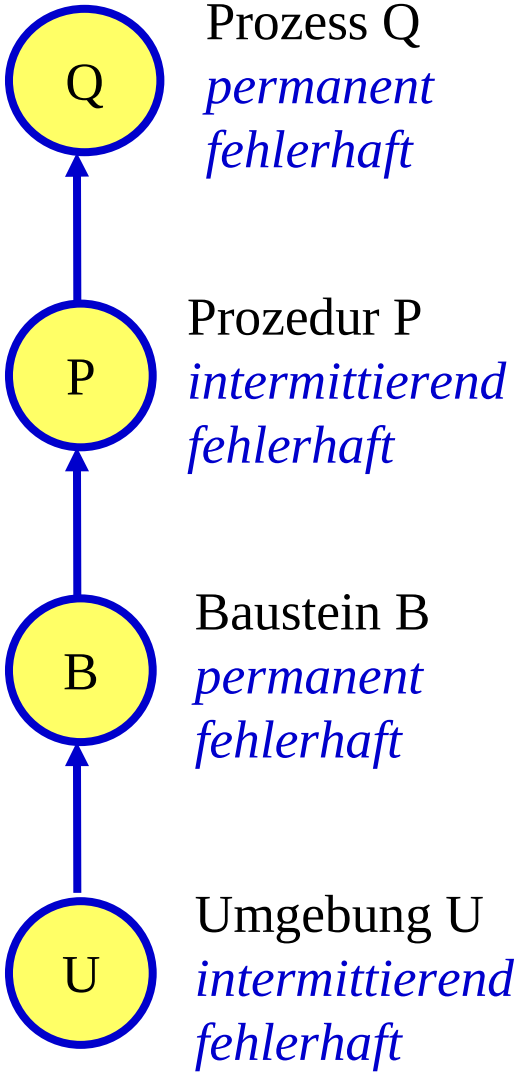
oberer Teil



Beispiel eines Schichtenmodells mit Pseudofunktionen zur Darstellung der vertikalen Fehlerausbreitung, unterer Teil

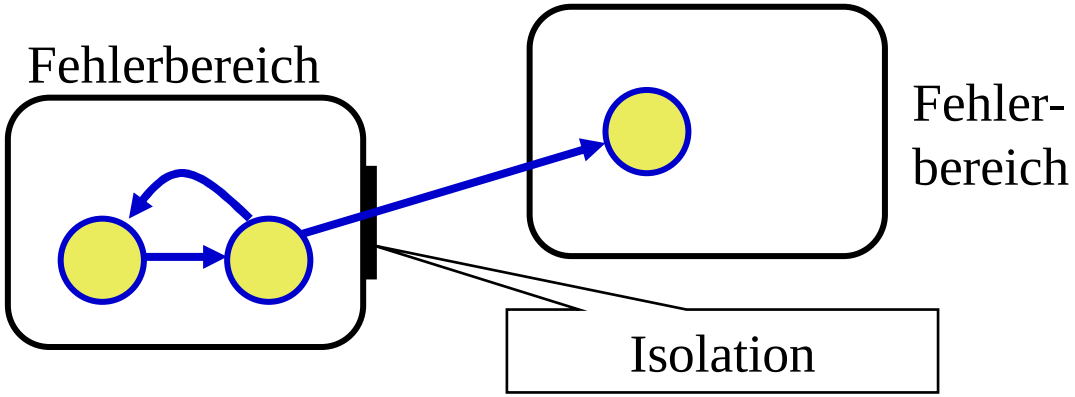
Z5: Fehlermodell und Ausbreitung: Fehlerausbreitung – Zeit

- Fehler breiten sich häufig nicht sofort aus
  - Programmierfehler wirkt sich erst aus, wenn fehlerhaftes Programmstück ausgeführt wird
  - Dateiverfälschung wirkt sich erst aus, wenn die Datei gelesen wird
- Fehlerlatenzdauer d(P,Q) zwischen P und Q
  - Zeitdauer, bis Fehler in K1 zu Fehler in K2 führt
- Fehlertypen
  - ein permanenter Fehler in einem Programm K1 kann zu einem intermittierenden Fehler in einem Prozess K2 führen
  - kurzfristige Überhitzung (ein intermittierender Fehler) kann zu einer permanenten Störung einer Hardwarekomponente führen



Z5: Fehlermodell und Ausbreitung: Fehlereingrenzung

- Fehler können sich über viele Wege ausbreiten
  - umfassende Fehlerbereiche
  - Fehlertoleranz schwer realisierbar



➔ Fehlereingrenzung
  - wenige Funktionszuordnungen im System
  - Maßnahmen zur Isolation zwischen Komponenten
    - » Hardware-Schutzmaßnahmen (Adressraum-Überwachung)
    - » Fehlerkorrektur (z.B. ECC-Speicher, z.B. wiederholtes Lesen)
    - » Plausibilitätskontrollen
    - » Selbstdiagnose und Abschalten (Einnahme eines sicheren Fehlerzustands)
    - » Einkapselung, Interaktionen nur über „enge“ Schnittstellen

Z5: Fehlermodell und Ausbreitung: Fehlereingrenzung

- Vertikale Eingrenzung in Hardware
  - HW-Fehlerkorrektur verhindert Ausbreitung zu SW
  - Zugriffskontrolle verhindert Fehlerausbreitung von SW auf HW
- Vertikale Fehlereingrenzung in Software
  - Syntax- und Konsistenzprüfungen
  - Modularität (Einkapselung)
  - unteilbare Aktionen, Transaktionen
- Horizontale Fehlereingrenzung in der Hardware und in tieferen Schichten des Betriebssystems
  - Modularität (Einkapselung)
- Horizontale Fehlereingrenzung bei Interaktionen in der Software
  - Syntax- und Konsistenzprüfungen
  - im verteilten System: Nachrichtenaustausch – Nachrichtenfehler

Z5: Fehlermodell und Ausbreitung: Fehlereingrenzung

Nachrichtenfehler und horizontale Fehlereingrenzung

- Verfälschung des Nachrichteninhalts beim Transfer
  - Blockprüfzeichen bzw. Signatur zur Fehlererkennung, negative Quittung und Wiederholung
- Nachrichtenverlust
  - Timeout bei ausbleibender Quittung, Wiederholung
- Nachrichtenvervielfältigung
  - Sequenzzahlen, Duplikaterkennung und Ignorieren
- Reihenfolgevertauschungen
  - Sequenzzahlen und Sortieren
- Spontanes Absenden unerwarteter fehlerhafter Nachrichten
  - Empfänger prüft Nachricht und Kontext, ignoriert oder weist zurück
- Verbindungsverlust
  - Timeout-Anzahl-Überwachung, Wahl alternativer Wege
- Sender erzeugt Nachricht mit fehlerhaftem Inhalt aber „gutem“ Blockprüfzeichen
  - Kontextprüfung beim Empfänger (häufig nicht möglich)
- Verfälschung so, dass Blockprüfzeichen stimmt
  - häufig nicht erkennbar und eingrenzbar

Z6: Redundanz

Überfluss

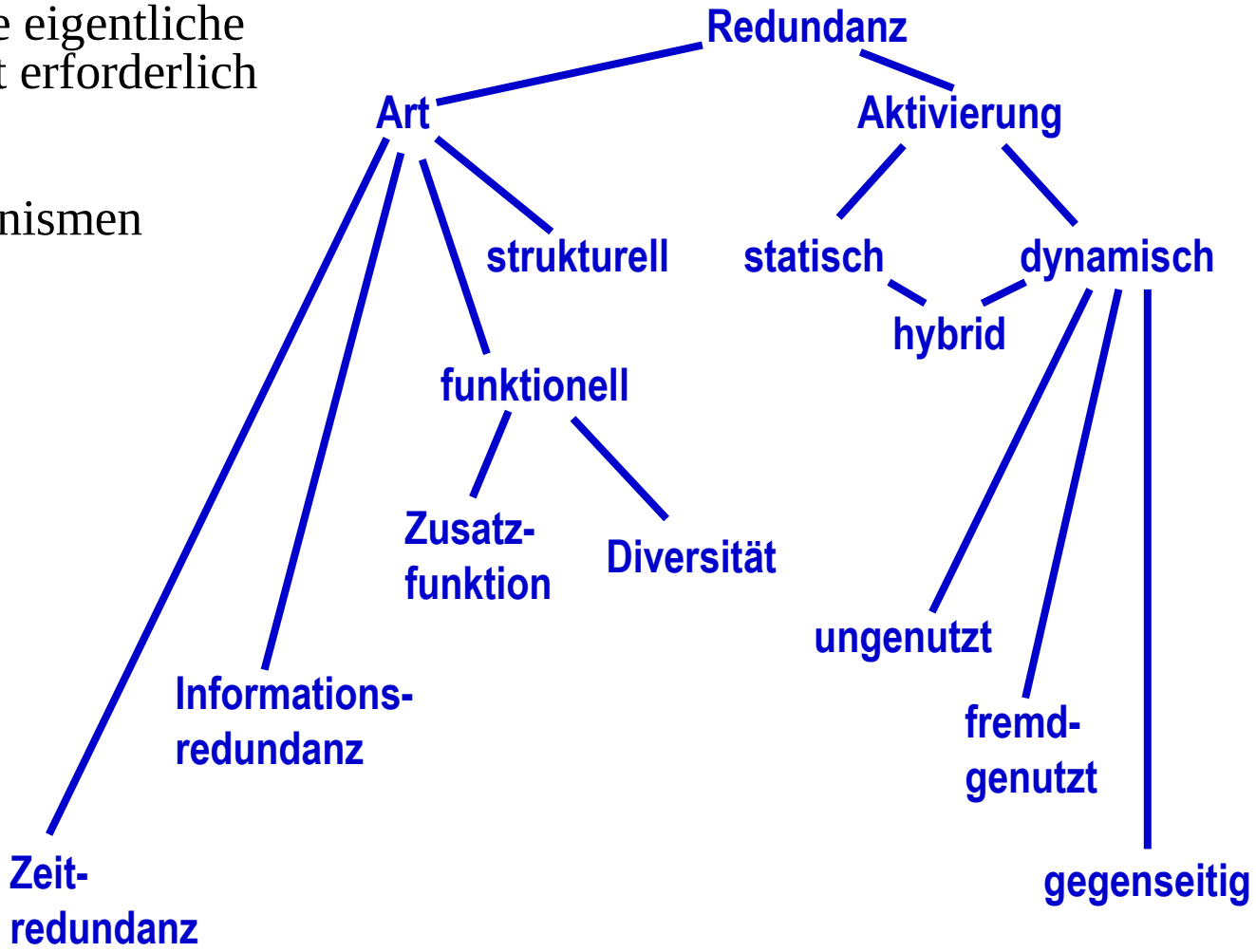
Mittel, welche für die eigentliche Systemfunktion nicht erforderlich sind
→ Basis der Fehlertoleranzmechanismen

Art

- Zeit
- Struktur
- Funktion
- Information

Aktivierung

- statisch, dynamisch, hybrid
- ungenutzt, fremdgenutzt, gegenseitig genutzt



Z6: Redundanz

- ***Strukturelle Redundanz***

  Erweiterung eines Systems um zusätzliche (gleich- oder andersartige), für den Nutzbetrieb entbehrliche Komponenten
  - Mehrrechnersystem
  - Datei-Replikate
  - RAID-Plattenarrays
  - Diagnosekomponente

Z6: Redundanz

- ***Funktionelle Redundanz***

  Erweiterung eines Systems um zusätzliche für den Nutzbetrieb entbehrliche Funktionen

  - Zusatzfunktionen
    - » Testfunktionen
    - » Funktionen zur Verwaltung von Toleranzmechanismen
    - » Parity-Erzeugung
    - » Rekonfigurationsfunktion
  - Diversität
    - » n-Versionsprogramme
      - unabhängiger Entwurf
      - gegensätzlicher Entwurf

Z6: Redundanz

- ***Informationsredundanz***
  zusätzliche Information neben der Nutzinformation
  - fehlererkennende Codes
  - fehlerkorrigierende Codes
  - Nachrichtenwiederholungen
  - Quersummen, Signaturen
  - Doppelverzeigerung

Z6: Redundanz

- ***Zeitredundanz***
  bezeichnet über den Zeitbedarf des *Normalbetriebs* hinausgehende zusätzliche Zeit, die für Fehlertoleranzmechanismen zur Verfügung steht
  - Zeit, um die wiederholte Nachricht zu verarbeiten
  - Zeit, um die Rekonfiguration durchzuführen
  - Zeit, um die Prüfsumme zu berechnen

Z7: Fehlerdiagnose

Liegt in einem *Fehlerbereich* ein Fehler vor?

- *Fehlertest*
  möglichst hohe Testgüte

- *Testgüte*
  - Fehlererfassung
    $C_F = p(DF_1 \vee \ldots \vee DF_n) / p(DF_1 \vee \ldots \vee DF_n \vee NF_1 \vee \ldots NF_m)$
    DF_i: richtig als Fehler diagnostizierte Fälle
    NF_i: fälschlich als richtig diagnostizierte Fehler (false positive)
  - Richtigerfassung
    $C_R = p(DR_1 \vee \ldots \vee DR_n) / p(DR_1 \vee \ldots \vee DR_n \vee NR_1 \vee \ldots NR_m)$
    DR_i: richtig diagnostizierte fehlerfreie Fälle
    NR_i: fälschlich als Fehler diagnostizierte fehlerfreie Fälle (false negative)

Z7: Fehlerdiagnose

Testzwecke

- ***Fehlererkennung***
  Sind die Komponenten fehlerfrei?
- ***Fehlerlokalisierung***
  Welche Komponenten sind fehlerfrei?
  Übrige sind u.U. fehlerhaft.
  - Genauigkeit
    - » Möglichkeiten zur Fehlererfassung
    - » Anforderungen der gewünschten Ausgrenzungsmaßnahmen
    - » Aufwand
  - ***Lokalisierungsbereich*** Lb_i
    - » Komponentenmenge Lb_i
    - » Test sagt für einen (vergangenen) Zeitpunkt t aus, ob alle Komponenten aus Lb_i fehlerfrei waren oder Fehler aus der Menge der zu tolerierenden Fehler aufgetreten sind ($\exists\ x \in Lb_i, t \in$ Zeit: FS (x, t) = fehlerhaft).

Z7: Fehlerdiagnose

- *Lokalisierungsbereich*
  Lokalisierungsbereiche LB_i müssen alle Fehlerbereiche FB_i überdecken

$$\bigcup_{i=1}^{f} Fb_i \subset \bigcup_{i=1}^{l} Lb_i$$

- *Behandlungsbereich*
  Komponentenmenge Bb_i, bei welcher ein Fehlerbehandlungs-Verfahren auf alle Komponenten $x \in Bb_i$ stets die gleiche Operation anwendet

  Behandlungsbereiche Bb_i müssen alle Fehlerbereiche FB_i überdecken

$$\bigcup_{i=1}^{f} Fb_i \subset \bigcup_{i=1}^{b} Bb_i$$

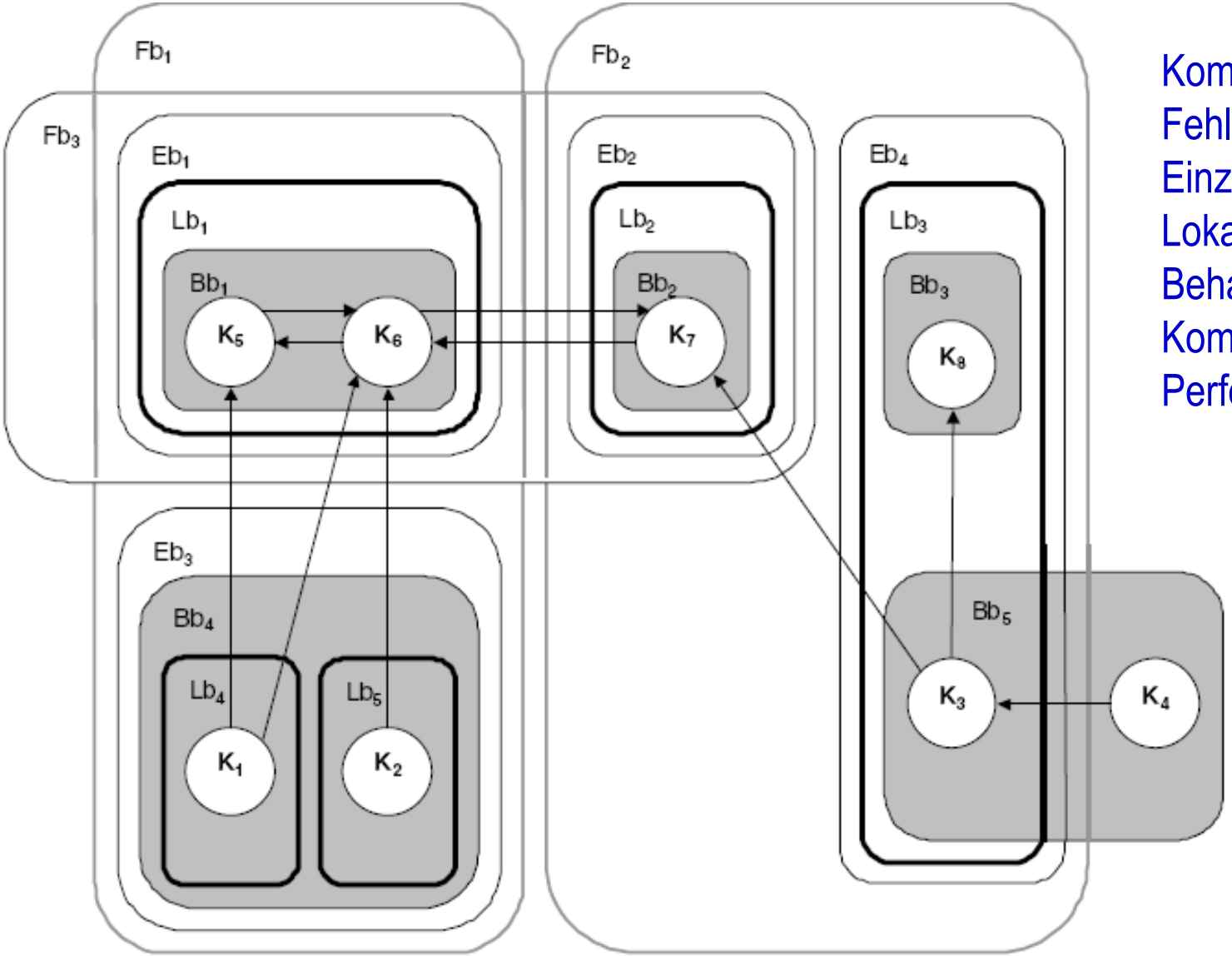
Z7: Fehlerdiagnose

Sinnvolle Einschränkungen

- Jeder Behandlungsbereich ist Teilmenge eines Fehlerbereichs
  - Fehlerbehandlung auf nicht zu viele Komponenten anwenden
- Jeder Lokalisierungsbereich ist im Komplement des Perfektionskerns Teilmenge eines Behandlungsbereichs
  - Entscheidung in welchem Behandlungsbereich Fehler bearbeitet wird
- Behandlungsbereiche sollen möglichst wenig Komponenten des Perfektionskerns enthalten
  - Behandlung verursacht unnötigen Aufwand
- Jeder Behandlungsbereich soll in einem Einzelfehlerbereich enthalten sein
  - Zielgerichtete Behandlung
- Jeder Lokalisierungsbereich umfasst einen Behandlungsbereich
  - nicht genauer lokalisieren als behandelt werden kann

Einfachster Fall

Jedem Einzelfehlerbereich entspricht genau ein Lokalisierungs- und ein Behandlungsbereich



Komponenten K_i
Fehlerbereiche Fb_i
Einzelfehlerbereiche Eb_i
Lokalisierungsbereiche Lb_i
Behandlungsbereichen Bb_i
Komponente K_4 im Perfektionskern

Beispiel aus K. Echtle: Fehlertoleranzverfahren

Der Behandlungsbereich Bb_4 wird durch zwei Lokalisierungsbereiche Lb_4 und Lb_5 gemeinsam abgedeckt (sie lokalisieren genauer als behandelt werden kann).
Im Einzelfehlerbereich Eb_4 ruft der Lokalisierungsbereich Lb_3 bei Fehlermeldung eine Behandlung in Bb_3 und Bb_5 hervor, wobei Bb_5 auch eine Komponente des Perfektionskerns enthält

Z7: Fehlerdiagnose – Testmethoden

- **Normalbetrieb: Fehlererkennung**
  Häufig Folgefehler-Erkennung, z.B.
  - *Adressraumüberschreitung*
  - *Zeitüberschreitung*
  - *Zugriffsrechtverletzung*
- **Ausnahmebetrieb: Fehlerlokalisierung**
  Testmodus mit speziellen Testdaten und dafür spezifiziertem Soll-Verhalten
  - Testprogramm
  - Konsistenzprüfung
  - Probenachrichten

Testarten

- *Strukturtest*
  Basis: strukturelles Fehlermodell
  Liegt die Soll-Struktur, bestehend aus fehlerfreien Komponenten und ihren Funktionszuordnungen vor?
- *Funktionstest*
  Basis: funktionelles Fehlermodell
  Wird die spezifizierte Soll-Funktion erbracht?

Z7: Fehlerdiagnose – Testmethoden

Testarten

- *Absoluttest / Akzeptanztest*
  Soll-Ist-Vergleich, ob a priori gegebene Konsistenzbedingungen über dem internen Zustand eines Testobjekts und / oder die von ihm errechneten Ergebnisse erfüllt sind
  ➔ „Testobjekt ist fehlerfrei“ / „Testobjekt ist fehlerhaft“
- *Relativtest*
  Ist-Ist-Vergleich mehrfach ausgeführter Berechnungen
  ➔ Mehrheitsentscheid

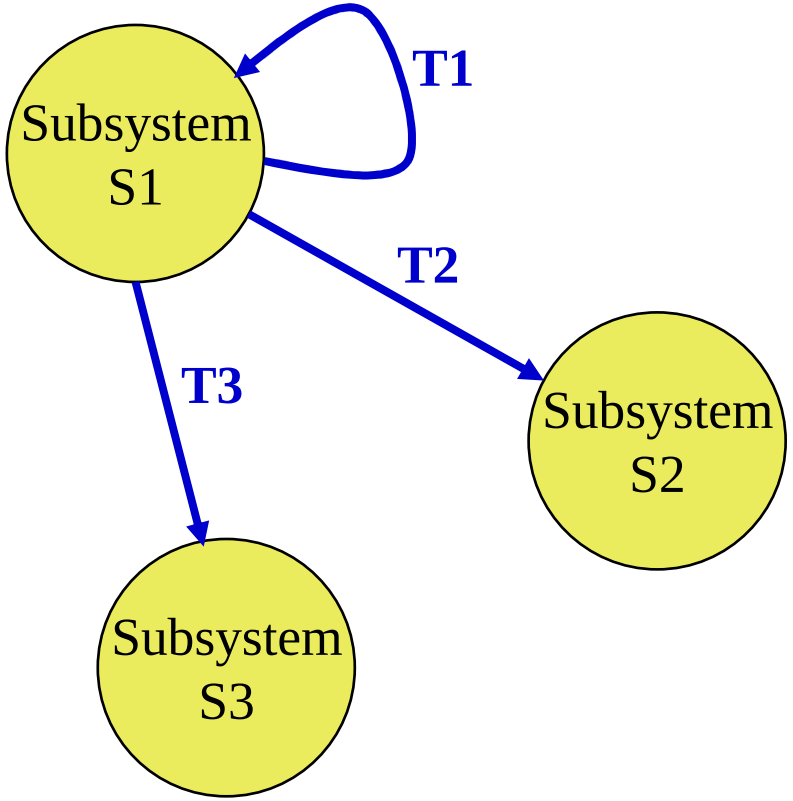
Informationsredundanz als Testbasis

– z.B. CRC, Quersumme, Signatur

Z7: Fehlerdiagnose – Diagnosegraph

Diagnosegraph

- Knoten:
  Testsubjekte und –objekte
- Kanten (gerichtet von Subjekt auf Objekt):
  Tests



Selbsttests

z.B. T1,
Prozessorselbsttest

Fremdtests

z.B, T2, T3
Speichertest
Verbindungstest