Zuverlässigkeit und Sicherheit

Beispiele

- **Verfügbarkeit von Web-Services „24/7“**
  - **Redundante Hardware (Rechner, Platten), um Verfügbarkeit zu erhöhen**
- **Flugzeugsteuerung, Space Shuttle/Ariane**
  - **Redundante Hardware (Rechner, Platten), um Sicherheit zu erhöhen**

Zuverlässigkeit und Sicherheit

Begriffe

Zuverlässigkeit

"Fähigkeit einer Betrachtungseinheit, innerhalb der vorgegebenen Grenzen denjenigen durch den Verwendungszweck bedingten Anforderungen zu genügen, die an das Verhalten ihrer Eigenschaften während der gegebenen Zeitdauer gestellt sind." [DIN 40041]

Sicherheit

"Sicherheit ist eine Sachlage, bei der das Risiko nicht größer als das Grenzrisiko ist." [DIN/VDE 31000 Teil2] Als Grenzrisiko ist dabei das größte, noch vertretbare Risiko zu verstehen [Hal99].

Zuverlässigkeit und Sicherheit

Begriffe

Verfügbarkeit

Die Wahrscheinlichkeit, dass ein System innerhalb eines spezifizierten Zeitraums funktionstüchtig (verfügbar) ist.

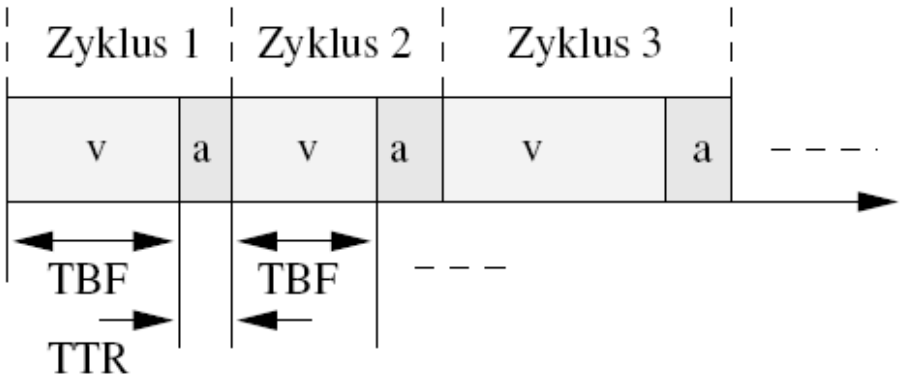
Unverfügbarkeit

Die Wahrscheinlichkeit, dass ein System innerhalb eines spezifizierten Betrachtungszeitraums funktionsuntüchtig (unverfügbar) ist.

Zuverlässigkeit und Sicherheit

Mathematische Grundlagen

Ein Student erwirbt einen Rechner. Leider zeigen sich gleich nach etwa 4 Betriebsstunden erste Mängel. Für die Fehlersuche, Ersatzteilbeschaffung und Reparatur benötigt der Student 1 Stunde. Danach läuft der Rechner weitere 96 Stunden problemlos, um abermals auszufallen. Diesmal dauert die Reparatur 1.5 Stunden. Weitere 96 Stunden später fällt der Rechner erneut aus, ist aber nach bereits einer halben Stunden wieder einsatzbereit. Die Zeit bis zum nächsten Ausfall (TBF, Time Between Failure) beträgt dann 72 Stunden, die Zeit zur Reparatur (TTR, Time To Repair) beträgt eine halbe Stunde. Insgesamt ergeben sich über einen Beobachtungszeitraum von 376 Stunden die folgenden Kennzahlen:



TBF	TTR
96h	1.5h
96h	0.5h
72h	0.5h
48h	0.5h
44h	1h
16h	1h
4h	1h

Zuverlässigkeit und Sicherheit

Mathematische Grundlagen

TBF (*Time Between Failures*):
Zeit bis zum nächsten Ausfall

TTR (*Time To Repair*):
Zeit für die Reparatur

TBF	TTR
96h	1.5h
96h	0.5h
72h	0.5h
48h	0.5h
44h	1h
16h	1h
4h	1h

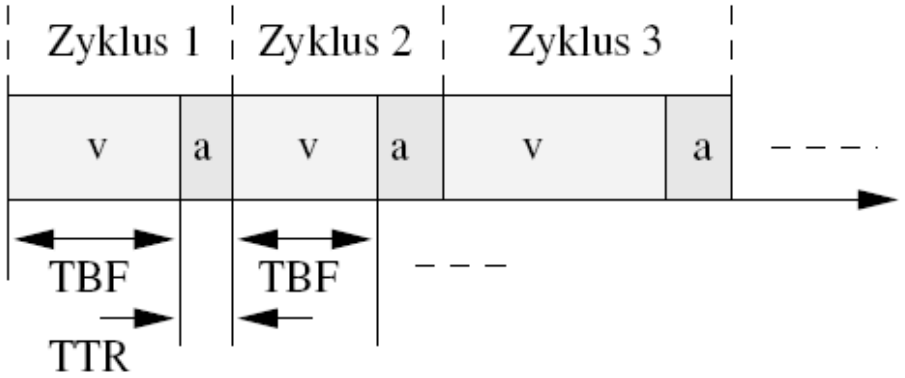
Wichtige statistische Kennzahlen:

- **MTBF (*Mean Time Between Failure*):**
  Durchschnittliche Zeit bis zum nächsten Ausfall

Im Beispiel: MTBF=376h/7=53.7h

- **MTTR (*Mean Time To Repair*):**
  Durchschnittliche Reparaturzeit

Im Beispiel:
MTTR = 6/7 = 0.857h = 51 Minuten



v = verfügbar
a = ausgefallen

TBF: Time Between Failure
TTR: Time To Repair

Zuverlässigkeit und Sicherheit

Mathematische Grundlagen

Allgemein:

MTBF (*Mean Time Between Failure*)

MTTR (*Mean Time To Repair*)

- **Jeweils arithmetisches Mittel aller TBF bzw. TTR**

$$MTBF = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} TBF_i$$

$$MTTR = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} TTR_i$$

Zuverlässigkeit und Sicherheit

Mathematische Grundlagen

Weitere interessante statistische Kennzahlen:

- **Wahrscheinlichkeit, das System verfügbar anzutreffen: (Dauer-) Verfügbarkeit**
  - **Verhältnis aus Zeit, in der das System verfügbar ist (für einen durchschnittlichen Zyklus: MTBF) zur Gesamtbetriebszeit (für einen durchschnittlichen Zyklus: MTBF + MTTR):**

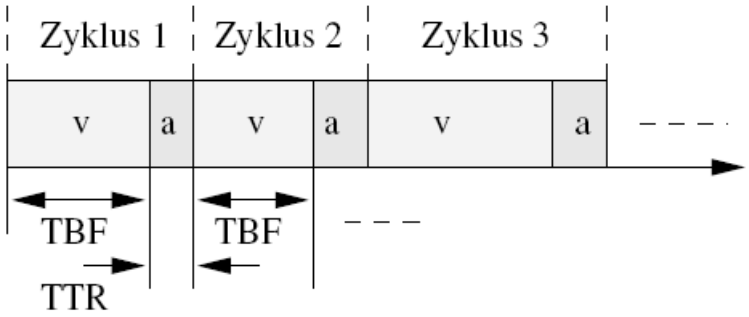
Im Beispiel: p = MTBF/(MTBF+MTTR) = 376/382 = 98.43%

- **Analog: Wahrscheinlichkeit, dass das System unverfügbar ist: Unverfügbarkeit**

Im Beispiel: q = MTTR/(MTBF+MTTR) = 6/382 = 1.57%

$$p = \frac{MTBF}{MTTR + MTBF} \quad \text{(Verfügbarkeit)} \quad \text{mit} \quad p + q = 1$$

$$q = \frac{MTTR}{MTTR + MTBF} \quad \text{(Unverfügbarkeit)}$$



Zuverlässigkeit und Sicherheit

Mathematische Grundlagen

Ausfallrate: Die (im Mittel) erwartete Zahl von Ausfällen pro Zeiteinheit

$$\lambda = \frac{1}{MTBF}$$

Reparaturrate: Die (im Mittel) mögliche Zahl von Reparaturen pro Zeiteinheit

$$\rho = \frac{1}{MTTR}$$

System wird nie repariert: $\rho = 0$
System nach Ausfall sofort wieder verfügbar (unendlich kurze Reparatur): $\rho = \infty$

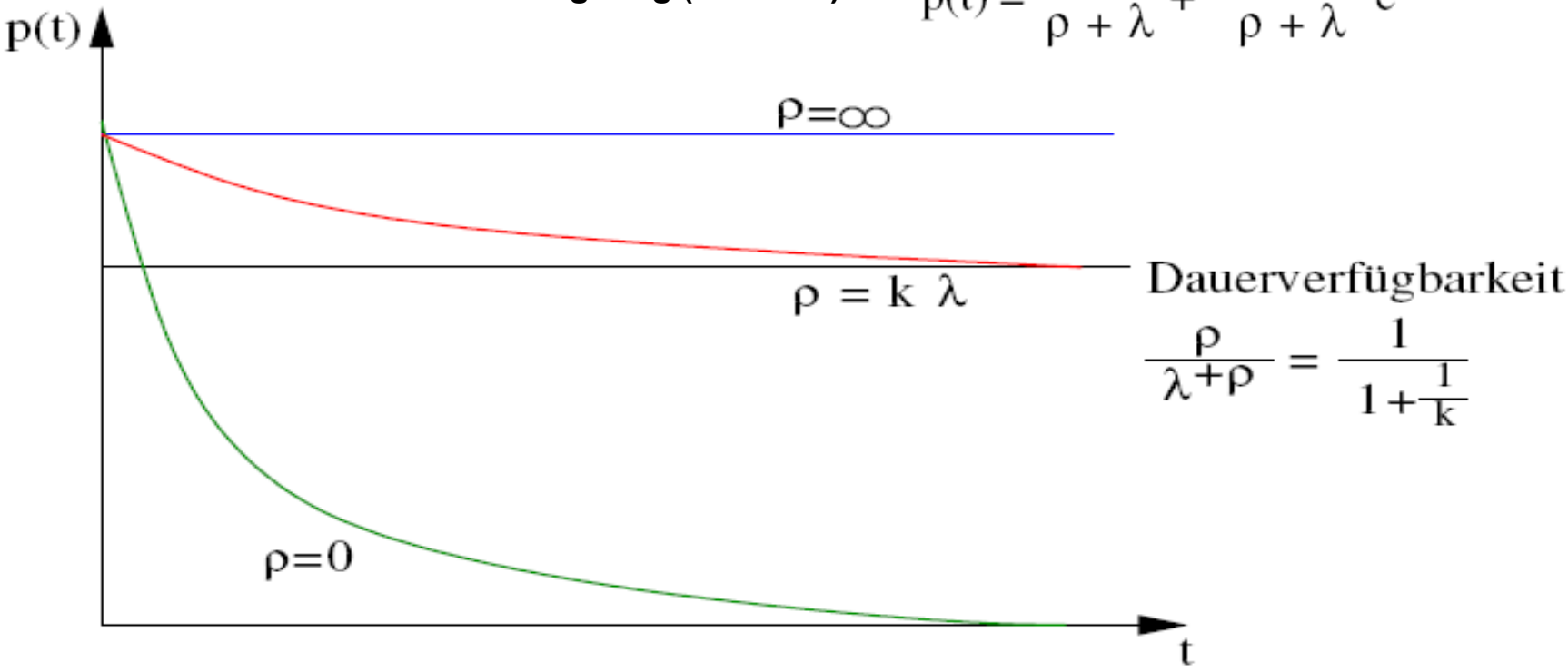
Zuverlässigkeit und Sicherheit

Mathematische Grundlagen

Zeitlich betrachtet sind Ausfälle nicht gleichverteilt

- **Nach einer Reparatur**
  - **Wahrscheinlichkeit, dass System funktionstüchtig ist, ist sehr groß (fast 100%)**
- **Nach langer Zeit (ohne Reparatur)**
  - **Wahrscheinlichkeit sehr gering (nahe 0%)**

$$p(t) = \frac{\rho}{\rho + \lambda} + \frac{\lambda}{\rho + \lambda} e^{-(\lambda+\rho)t}$$



Dauerverfügbarkeit

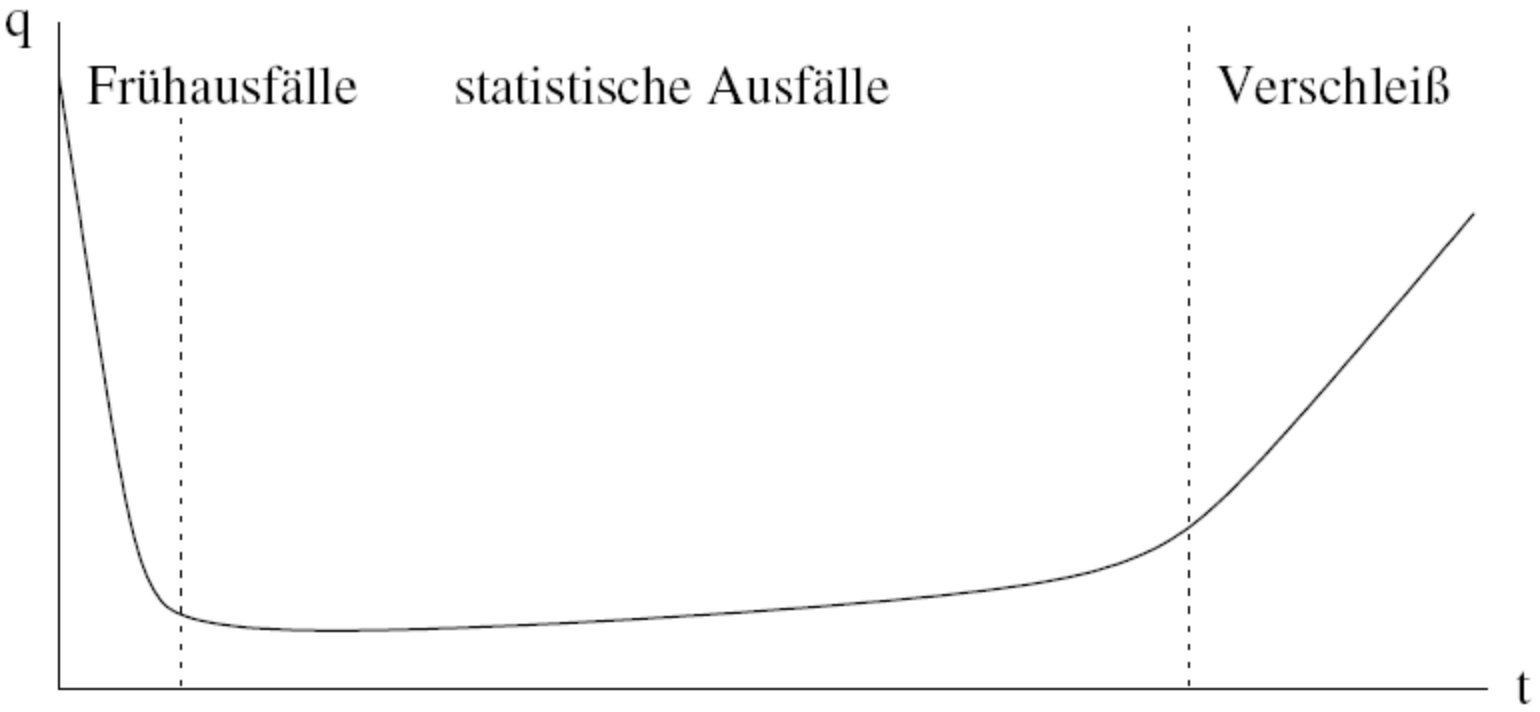
$$\frac{\rho}{\lambda+\rho} = \frac{1}{1+\frac{1}{k}}$$

Zuverlässigkeit und Sicherheit

Mathematische Grundlagen

Badewannenkurve

- Hohe Ausfallwahrscheinlichkeit bei neuen Systemen
  - Abhilfe: „Voraltern“
- Zunehmende Ausfallwahrscheinlichkeit durch Verschleiß bei älteren Systemen
  - Abhilfe „Wartung, frühzeitiger Austausch“



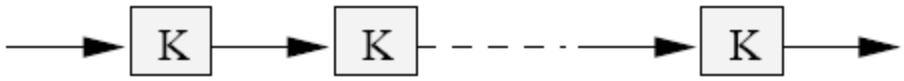
Zuverlässigkeit und Sicherheit

Mathematische Grundlagen, mehrere Komponenten

Serienschaltung (Abhängige Komponenten).

- **System nur dann verfügbar, wenn alle Einzelkomponenten verfügbar sind**
  - **Beispiel: Rechnersystem nur dann verfügbar, wenn CPU, Speicher und Peripheriemodule verfügbar sind**
- **Gesamtverfügbarkeit:**
  - **Produkt der Einzelverfügbarkeiten.**
  - **kleiner als die kleinste Einzelverfügbarkeit.**

$$p_{gesamt} = \prod_{i=1}^{n} p_i$$



Zuverlässigkeit und Sicherheit

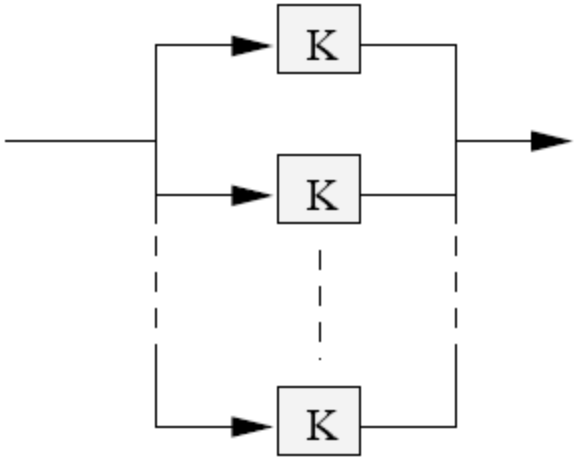
Mathematische Grundlagen, mehrere Komponenten

Parallelschaltung (redundante Komponenten)

- System ist verfügbar, sobald eine von mehreren Komponenten verfügbar ist
- redundantes System
- Gesamtunverfügbarkeit:
  - Produkt der Einzelunverfügbarkeiten
  - deutlich kleiner als die Einzelunverfügbarkeiten

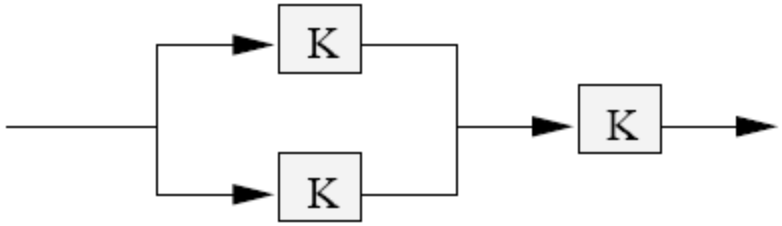
$$q_{gesamt} = \prod_{i=1}^{n} q_i$$

$$p_{gesamt} = 1 - q_{gesamt}$$



Zuverlässigkeit und Sicherheit

Redundante Systeme



Komplexe Systeme:

- **Meist Mischungen aus Serienschaltung und Parallelschaltung**

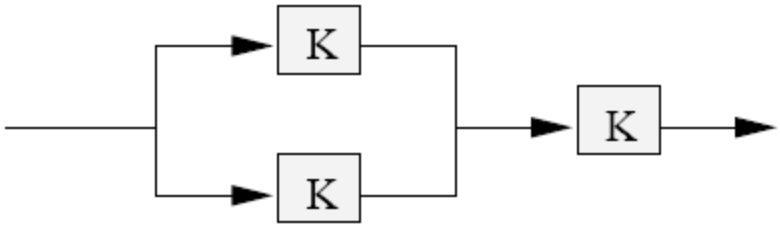
z.B. Parallelschaltung: Koppeleinrichtung (Arbiter) erforderlich, Anforderung an Arbiter: Hochverfügbar

Ermittlung der Verfügbarkeit bei komplexen Systemen: Verfügbarkeits-Ersatzschaltbild, daraus Verfügbarkeit ermitteln

Zuverlässigkeit und Sicherheit

Redundante Systeme

Betriebsarten redundanter Systeme



- Hohe Verfügbarkeit
- Hohe Sicherheit

z.B. 2 (redundante) Rechner

- **Hoch-verfügbar: 1:2 (1 aus 2)- System: mindestens einer von zwei Rechnern muss verfügbar sein**
  - **Ersatzschaltbild: Rechner parallel, in Reihe mit Koppelelement**
- **Hoch-sicher: 2:2**
  - **Ersatzschaltbild: beide Rechner und Koppelelement in Reihe**

Zuverlässigkeit und Sicherheit

Zuverlässigkeitssteigerung

Erforderlich: Maßnahmen in den Bereichen

- **Infrastruktur**
- **Hardware**
- **Software**
- **Management**

Infrastruktur

- **Sicherung der Stromversorgung.**
  - **unterbrechungsfreie Stromversorgung (USV)**
  - **Powerfail-Interrupt(mit kurzer Energiereserve): Systeme in sicheren, konsistenten Zustand bringen**
- **Klimatisierung.**
  - **Problem: Überhitzung: Vorzeitige Alterung, sporadische oder Totalausfälle**
- **Schutz vor Feuer, Nässe, Blitzschlag und Einbruch**

Zuverlässigkeit und Sicherheit

Redundante Systeme

Redundanzarten zur Erhöhung der Verfügbarkeit

- statische Redundanz (cold standby)
  - Redundante Komponente wird bei Systemausfall aktiviert, kompletter Neuanlauf
  - Problem: Vorübergehende Unverfügbarkeit
- dynamische Redundanz (hot standby)
  - Redundante Komponente ist aktiviert, kennt den aktuellen Systemzustand des Gesamtsystems
  - Ausgaben nur von der Hauptkomponente
  - Im Fehlerfall (Hauptkomponente): Umschalten auf die redundante Komponente
- Doppelsystem
  - Die redundanten Komponenten bearbeiten jeweils Teilaufgaben (keine Hauptkomponente)
  - Im Fehlerfall übernimmt die redundante Komponente die Aufgaben der defekten Komponente

Zuverlässigkeit und Sicherheit

Zuverlässigkeitssteigerung

Hardware

Redundanz der Systemkomponenten

Probleme:

Detektion des Ausfalls (welches Werkzeug kennen wir da schon?)

Übertragung des Systemzustands auf redundante Komponenten

- **Hot Standby: Redundante Systeme arbeiten parallel mit derselben Eingabeinformation**
- **Cold Standby: Aktive Komponente muss ihren Systemzustand bei konsistenten Systemzuständen sichern; Redundante Komponente verwendet den zuletzt gesicherten Systemzustand**

Redundanz der Einzelkomponenten

- **CPU**
- **Speicher**
- **Festplatten (RAID)**
- **Netzteile**

Zuverlässigkeit und Sicherheit

Zuverlässigkeitssteigerung

Hardware

Weitere Maßnahmen die nicht auf Redundanz beruhen

Steigerung der Verfügbarkeit der Einzelkomponenten durch

- **Auswahl zuverlässiger Bauelemente (z.B. „ruggedized“ oder MIL-Varianten).**
- **Steigerung der mechanischen Festigkeit (Vibrationsschutz; Vergiessen, Schäumen).**
- **Fixierung löslicher Elemente (Lackierung von Schrauben, „Coaten“).**
- **Vermeidung von Steckkontakten (bzw. Überwachung der Steck-Zyklen)**
- **Wärmeschutz (Kühlung)**
- **Überwachung und Regelung von Spannungen und Temperaturen**

Zuverlässigkeit und Sicherheit

Zuverlässigkeitssteigerung

Software

- **Überwachung der Systemkomponenten bzw. auch der Diagnose der Fehlerursache (S.M.A.R.T)**
- **Umschaltung und Abgleich zwischen dem aktiven System und dem Ersatzsystem**
- **Backupstrategien für Daten und Systemzustände, um Informationsverlust bei Ausfall eines Systems vorzubeugen**

Zuverlässigkeit und Sicherheit

Zuverlässigkeitssteigerung

Management

- **Wartung**
  - **regelmäßiger Austausch alternder Komponenten**
  - **Problem: Verfügbarkeit von Ersatzkomponenten (NASA, 2002: 8086er Aktion für Boosterraketen-Diagnosesystem, eBay, Yahoo,...)**
- **Staffing**
  - **Schnelle Verfügbarkeit von kompetentem Personal**
- **Notfallpläne**
  - **Minimieren von Stillstands- bzw. Ausfallzeiten**
  - **Dokumentation möglicher Fehlersituationen und detaillierte Aktionspläne**

Zuverlässigkeit und Sicherheit

Harte Realzeitsysteme: Space-Shuttle

Frühere NASA-Projekte: *fail-operational/fail-operational/fail-safe*

Wegen geringer Fehlerrate für Space-Shuttle: *fail-operational/fail-safe*

Fail-operational: ein Fehler kann toleriert werden, Betrieb ohne funktionale Einbussen

Fail-safe: Bei Auftritt eines zweiten Fehlers kann das Shuttle immer noch sicher zur Erde kommen, auch wenn nicht mehr alle Systeme an Board verwendet werden können

Zuverlässigkeit und Sicherheit

Harte Realzeitsysteme: Space-Shuttle

Diskutierte High-Availability-Strategien:

- **Unabhängige Systeme**
  - **Klassisches Verfahren in der Luftfahrt (z.B. Boeing 767)**
  - **Unabhängige Sensor-, Computer- und Aktorkreise**
  - **Problem mangelnde Redundanz. Fällt eine Komponente aus, ist der gesamte Kreis nicht mehr verwendungsfähig**
- **Master/Slave Konzept**
  - **Ein Master bedient ständig die Sensorik, während die anderen Rechner Informationen „im Mithör-Modus“ erhalten**
  - **Ausfall des Masters: redundanter Ersatzrechner übernimmt seine Aufgabe**
  - **Problematisch: sicherer automatischer Switch-Over zum Ersatzsystem ist eine sehr komplexe Aufgabe (in kritischen Flugphasen nur 400ms Reaktionszeit)**
- **Verteiltes Kommando**
  - **Realisiertes Konzept:**
  - **Alle Steuerrechner hören auf alle Sensoren und erzeugen die gleichen Ausgaben**