Konstruktive Qualitätssicherung

- Software Richtlinien
- Typisierung
- Vertragsbasierte Programmierung
- Fehlertolerante Programmierung
- Portabilität
- Dokumentation

Fehlertolerante Programmierung

Fehlertolerante Programmierung:

Verbesserung des Reaktionsverhaltens eines Programms im Fall eines Fehlers.

Die Bedeutung der fehlertoleranten Programmierung hängt von der Anwendung ab. Ein Textverarbeitungssystem kann auch mal abstürzen, Flugzeugsteuerungssoftware sollte das nicht.

Fehlertolerante Programmierung

- Software Redundanz
- Selbstüberwachende Systeme
- Ausnahmebehandlung

Software Redundanz – Ausprägungen 1/2

- **Funktionale Redundanz**
  Erweiterung um zusätzliche Funktionen, die ausschließlich der Erhöhung der Fehlertoleranz dienen.

- **Informationelle Redundanz**
  Nutzdaten werden um zusätzliche Informationen angereichert.

Software Redundanz – Ausprägungen 2/2

- **Temporale Redundanz**
  Zeitanforderungen werden übererfüllt, so dass ggfs. eine Wiederholung stattfinden kann.

- **Strukturelle Redundanz**
  Mehrfachauslegung von Komponenten.
  ➔Details nächste Seiten

Strukturelle Redundanz

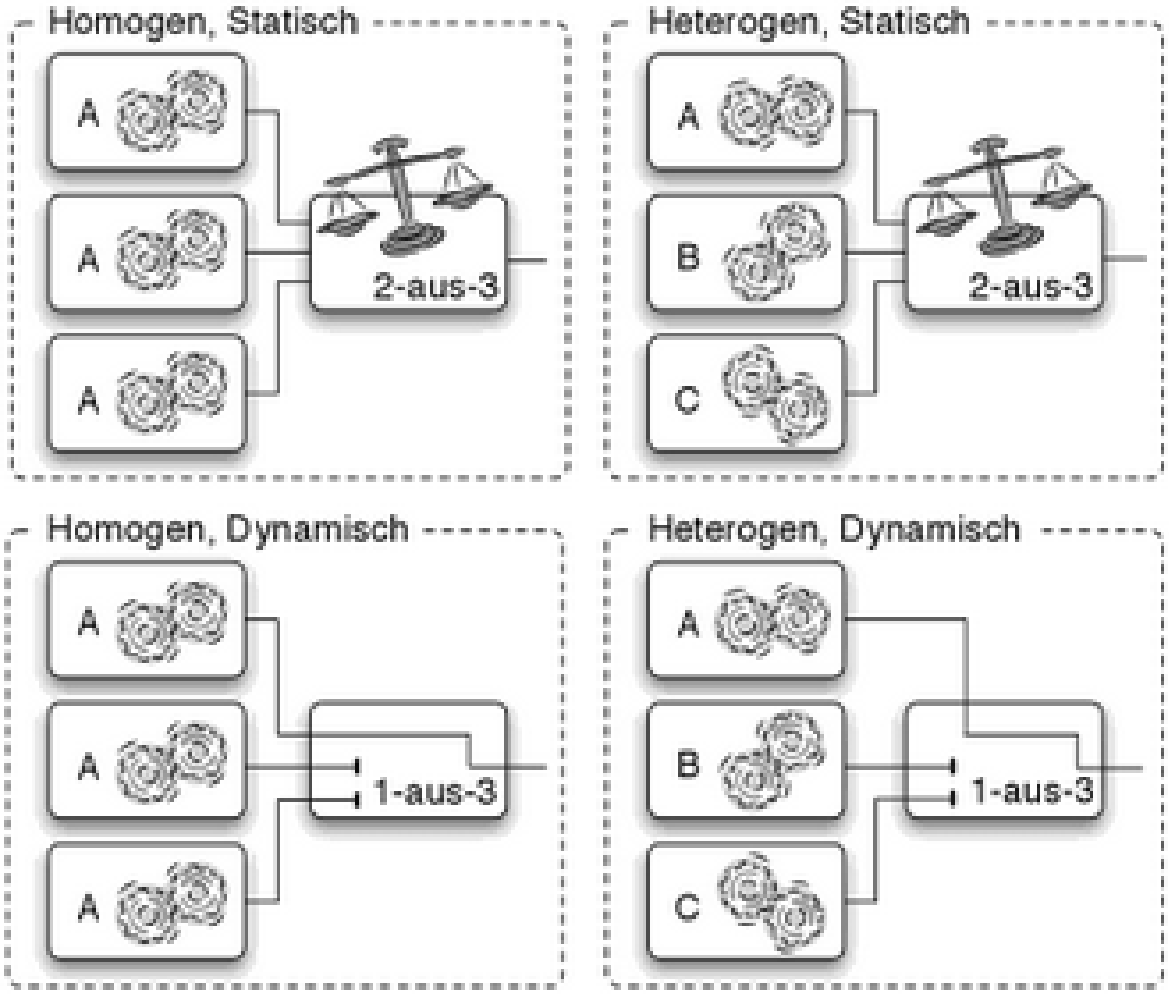
Ausprägungen:

- Homogene Redundanz (n Komponenten gleicher Bauart)
  Ausschließlich im Hardware Bereich
- Heterogene Redundanz (n Komponenten unterschiedlicher Bauart)

Ausprägungen:

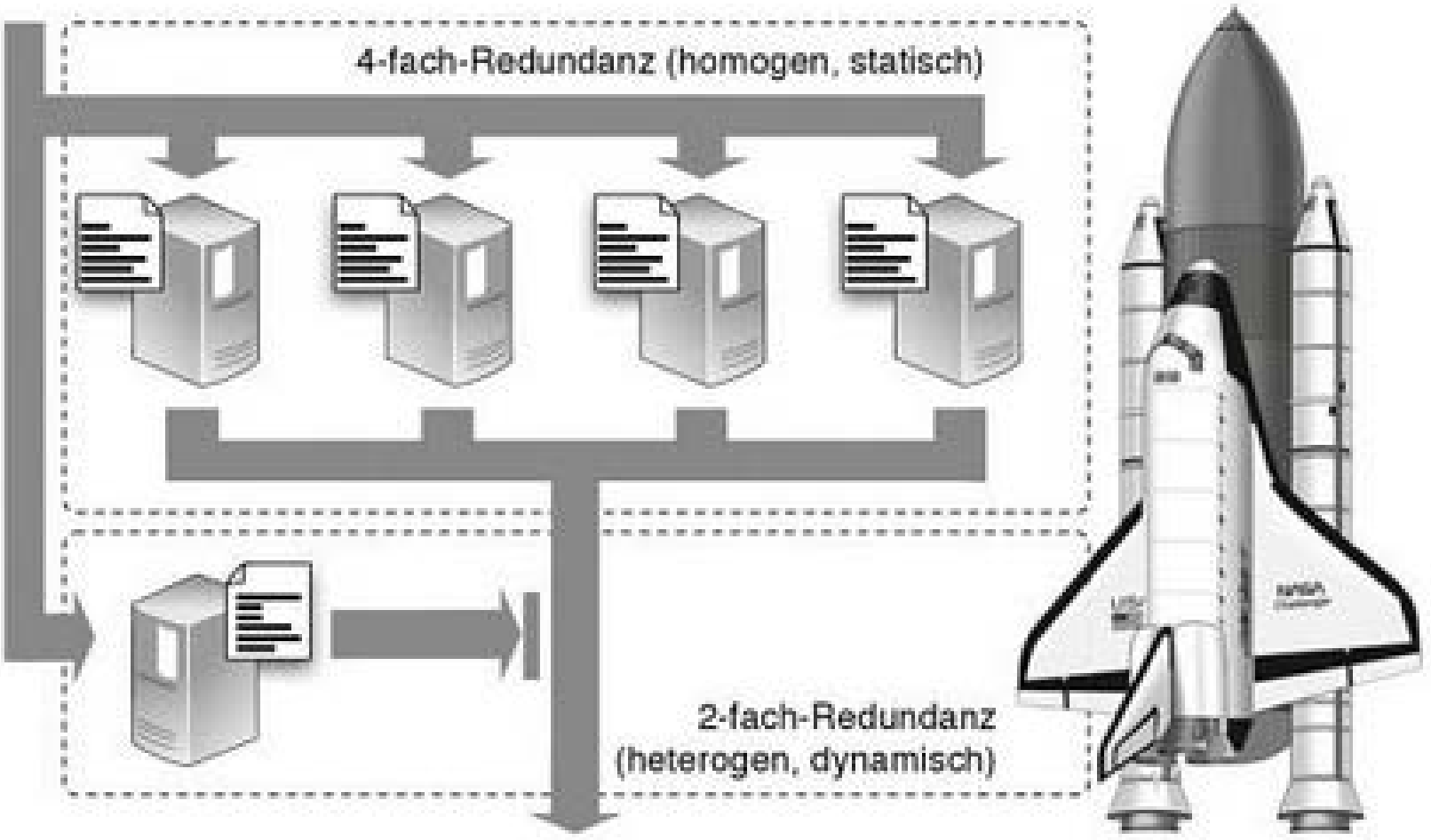
- Statische Redundanz (alle Komponenten aktiv, Ergebnisse werden durch Voter verglichen)
- Dynamische Redundanz (Ersatzkomponenten, die bei Bedarf aktiviert werden)

Homogene und Heterogene Redundanz



Quelle: D. Hoffmann:
Software Qualität

SW Redundanz, Bsp Space Shuttle



Fehlertolerante Programmierung

- Software Redundanz
- Selbstüberwachende Systeme
- Ausnahmebehandlung

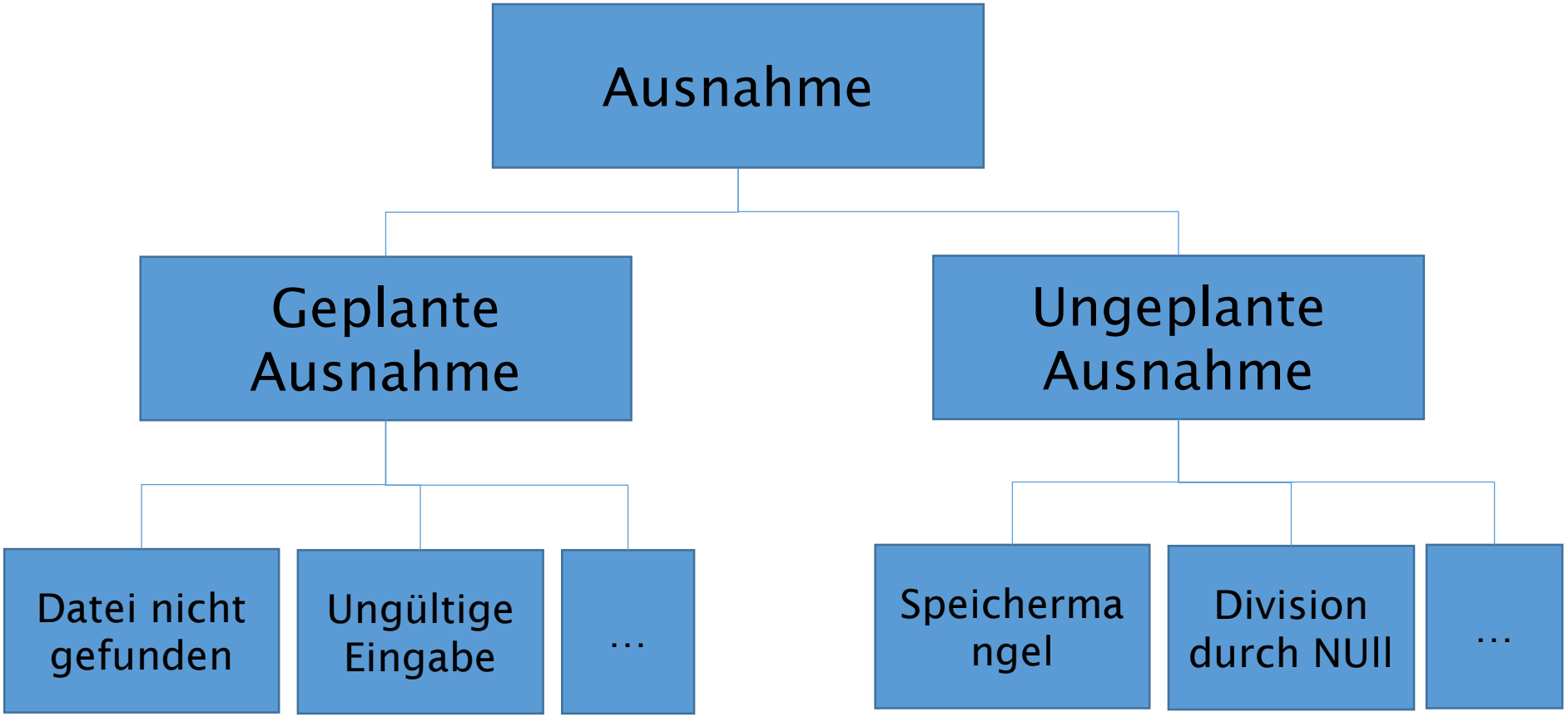
Reaktionsszenarien

- **Fail-Safe Reaktion**
  Wechsel in einen sicheren Zustand (z.B. Selbstabschaltung)
  Bsp: Selbstaktivierende Notbremse eines Aufzugs,
  Sitzplatzverriegelung einer Achterbahn.
- **Selbstreparatur**
  Selbstdiagnose und Reparatur
  Bsp: automatische Wiederherstellung von Dateien
- **Reaktivierung**
  Watchdog Logik, Bsp: Explorer Prozess in Windows XP

Fehlertolerante Programmierung

- Software Redundanz
- Selbstüberwachende Systeme
- Ausnahmebehandlung

Ausnahmen


Ausnahme
Geplante Ausnahme
Ungeplante Ausnahme
Datei nicht gefunden
Ungültige Eingabe
…
Speicherma ngel
Division durch NUll
…

Unterschiedliche Philosophien

- Delegation der Ausnahmebehandlung ans Betriebssystem
- Behandlung der Ausnahmen in dem Konzept der Programmiersprache verankert.

Ausnahmebehandlung

- In C: Ausnahmebehandlung mit geschachtelten If Strukturen
- In Java: Ausnahmebehandlung mit Exceptions

Geschachtelte If-Strukturen

```
int readFile(String name) {
  int error_code;
      <Öffne Datei>
      if (<Datei erfolgreich geöffnet>) {
          <Ermittle Dateigröße>
          if (<Dateigröße erfolgreich ermittelt>) {
              <Belege Speicher>
              if (<Speicher erfolgreich belegt>) {
                  <Lese Dateiinhalt ein>
                  if (<Dateiinhalt erfolgreich gelesen>) {
                      error_code = 0;
                  } else
                      error_code = 1;
              } else
                  error_code = 2
          } else
              error_code = 3
          <Schließe Datei>
          if (<Datei nicht schließbar>)
              error_code = 4
      } else
          error_code = 5
    return error_code;
}
```

Exception Handling in Java

```
void readFile(String name) {
  try {
      <Öffne Datei>
      <Ermittle Dateigröße>
      <Belege Speicher>
      <Lese Dateiinhalt ein>
      <Schließe Datei>
      }
  catch (DateiÖffnenFehler ...) { <Fehlerreaktion> }
  catch (DateiGrößenFehler ...) { <Fehlerreaktion> }
  catch (SpeicherBelegenFehler ...) { <Fehlerreaktion> }
  catch (DateiLesenFehler ...) { <Fehlerreaktion> }
  catch (DateiSchliessenFehler ...) { <Fehlerreaktion> }
}
```