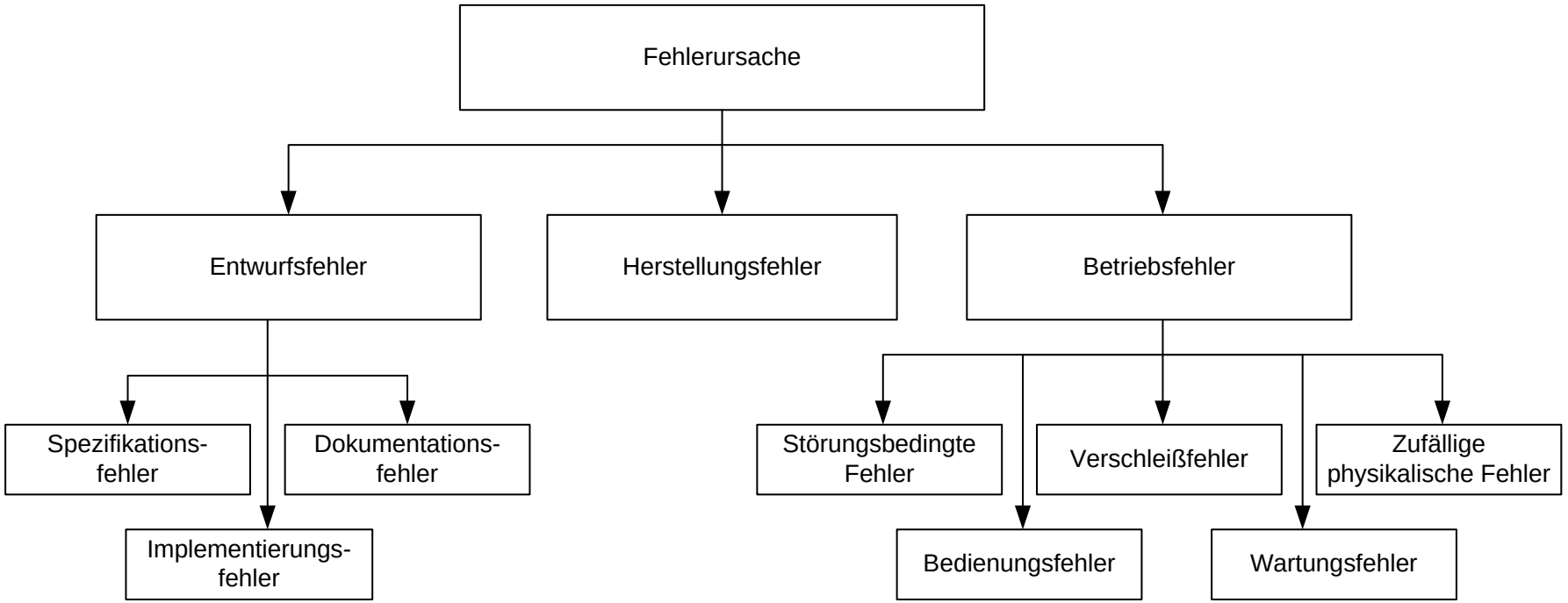
Fehlertoleranz

Fehlermodell

Fehlerursachen



Klassifizierung von Fehlern

- Unterscheidung nach Entstehungsort:
  - Hardware
  - Software

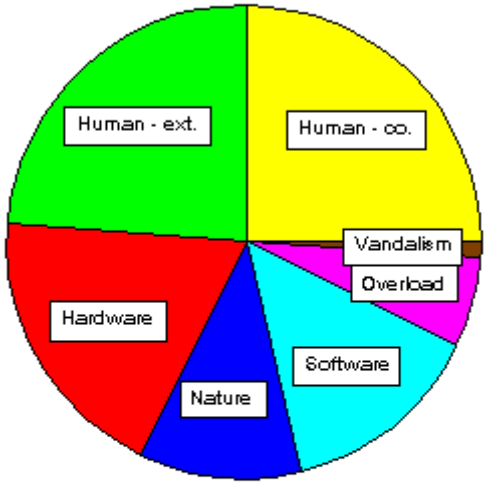
- Unterscheidung nach Fehlerdauer:
  - permanent
  - intermittierend (flüchtig)
    - periodisch
    - wiederkehrend
    - einmalig

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.

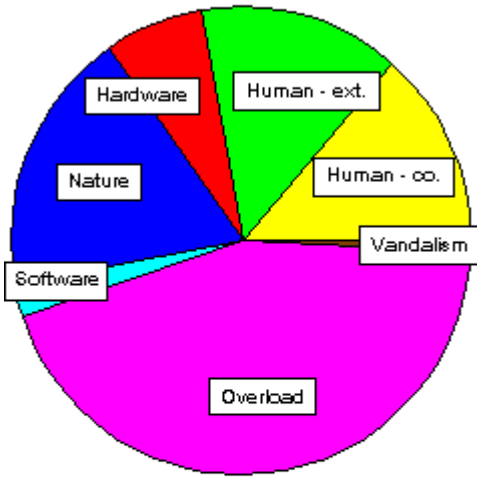
Ursachen und Wirkung

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 eror - 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
  - $Fb_1 \cup ... \cup Fb_n \neq S$ ($\Rightarrow$ es gibt einen Perfektionskern $S \setminus (Fb_1 \cup ... \cup Fb_n)$
  - $\exists i,j \in \{1...n\}: Fb_i \cap Fb_j \neq \emptyset$ ($\Rightarrow$ 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)
  - Anhalte-Ausfall: sobald ein Fehler aufgetreten ist, gibt das System kein Ergebnis mehr aus (fail-stop) ! jedes ausgegebenen Ergebniss 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.