Sichere Systeme

Sicherheit und Zuverlässigkeit – Safety and Reliability

Hochschule Esslingen
University of Applied Sciences

"Alles, was schiefgehen kann, wird auch schiefgehen" (nach Edward Murphy, ca. 1949)

Explosion der Raumfähre Challenger 1986 (Quelle: www.flickr.com)

… und das, obwohl es zu Murphys Zeiten doch noch gar keine Software gab …

Webseite:	*www.hs-esslingen.de/mitarbeiter/Werner.Zimmermann*
Material zur Vorlesung:	*Menu Vorlesungen – Sichere Systeme*
Sprechstunden und aktuelle Meldungen:	*Menu Aktuelles*

Prof. Dr.-Ing. Werner Zimmermann, Hochschule für Technik Esslingen - Fakultät Informationstechnik

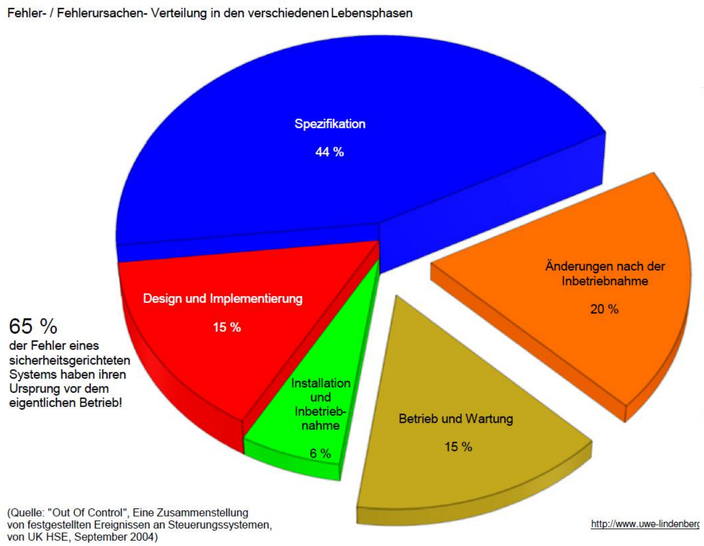
Flugzeuge, Automobile, Fertigungsmaschinen und andere technische Produkte

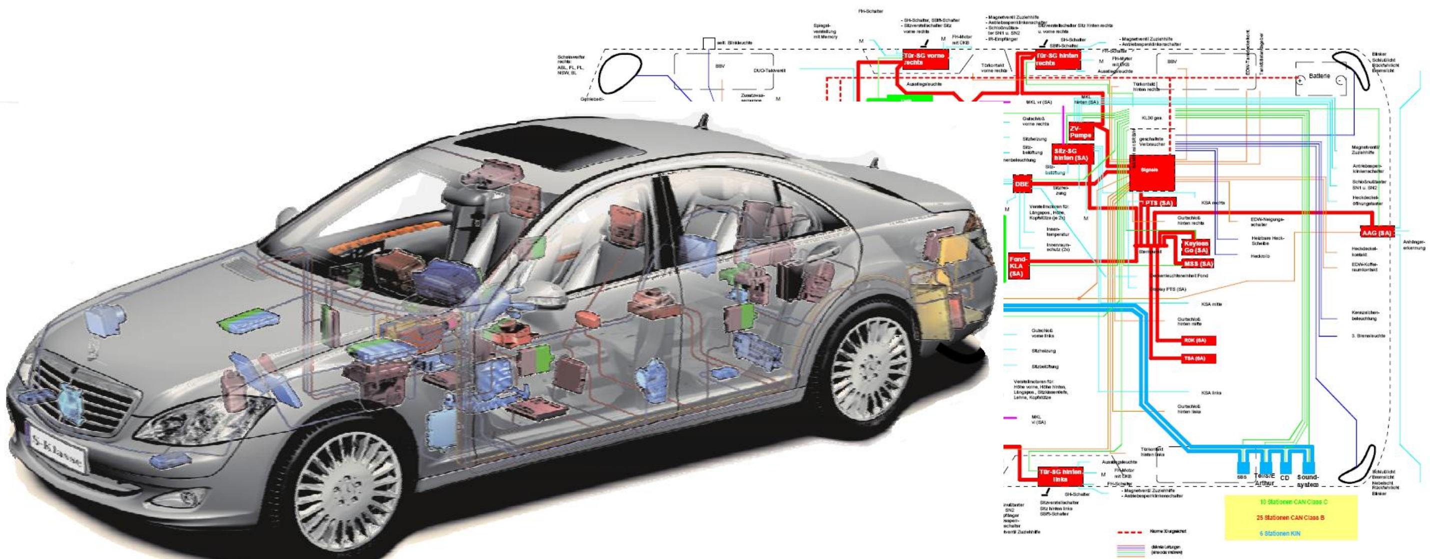
- Komplexe Kombinationen von **mechanischen** und **elektrischen** Komponenten
- Gesteuert durch **vernetzte** Computer (**elektronische** Steuergeräte) durch **Software**
- Fehlfunktionen verärgern: Der Kunde soll wiederkommen, nicht das Produkt → **Reliability**
- Fehlfunktionen beschädigen Güter und verletzen oder töten Menschen → ***Safety critical***

Wegen der Komplexität sind **Fehler** in Hard- und Software **unvermeidbar**:

Systeme müssen Komponentenausfälle beherrschen: Verschleiß mechanischer Komponenten, blockierte Klappen, Ventile und Motoren, Kurzschlüsse und Brüche von Kabeln und Steckern, Über- und Unterspannung, EMV-Störungen UND Software-Fehler:

→ **Fehlertoleranz** in Hardware (**Redundanz**) und Software (**Functional Safety**)



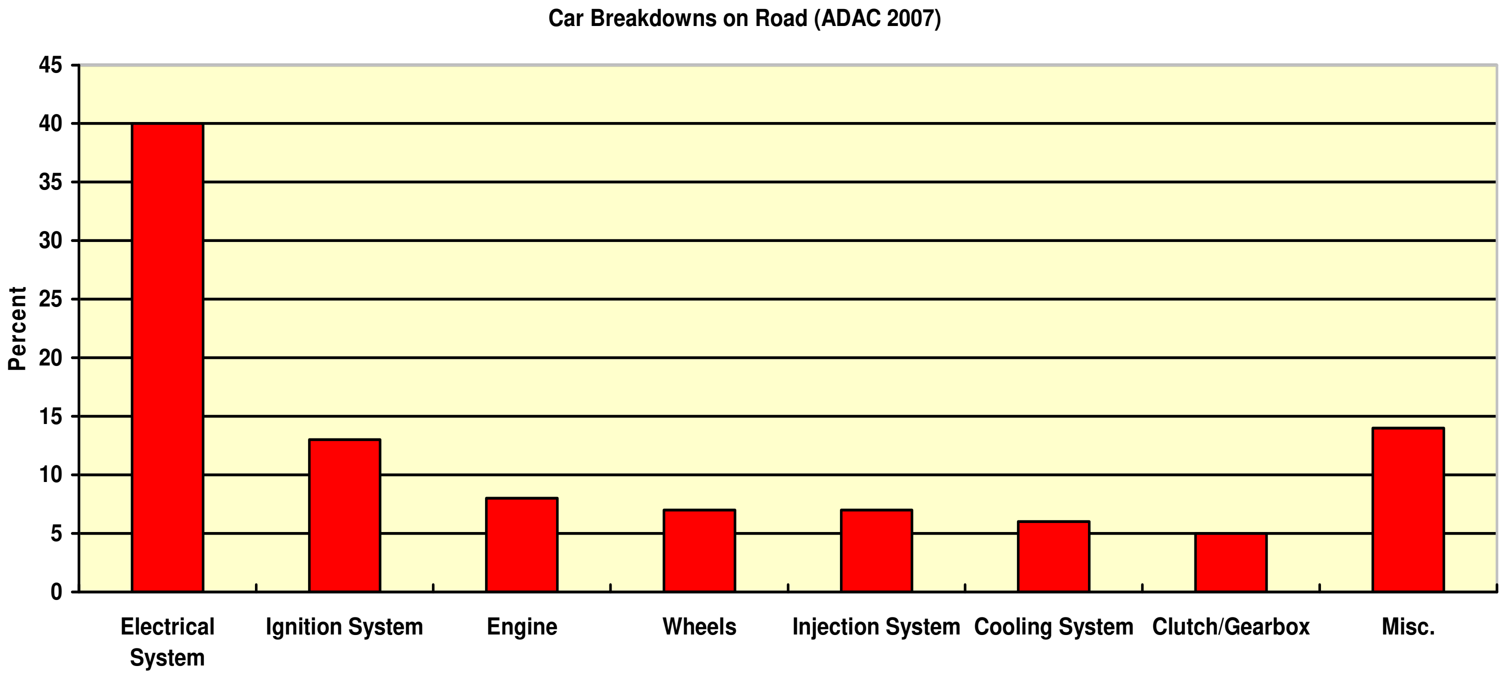


Mercedes-Benz S-Class, Source: Dr. Hofmann, Daimler

System Safety and Reliability

Hochschule Esslingen
University of Applied Sciences

1 Basics



Breakdowns per 1000 units for 2 year old cars:

- Compact cars (VW Golf, MercedesBenz A, BMW 1, …) Best in class: 2 Worst in class: 13
- Lower Luxury class (Audi A6, Mercedes Benz E, BMW 5) Best in class: 4 Worst in class: 8

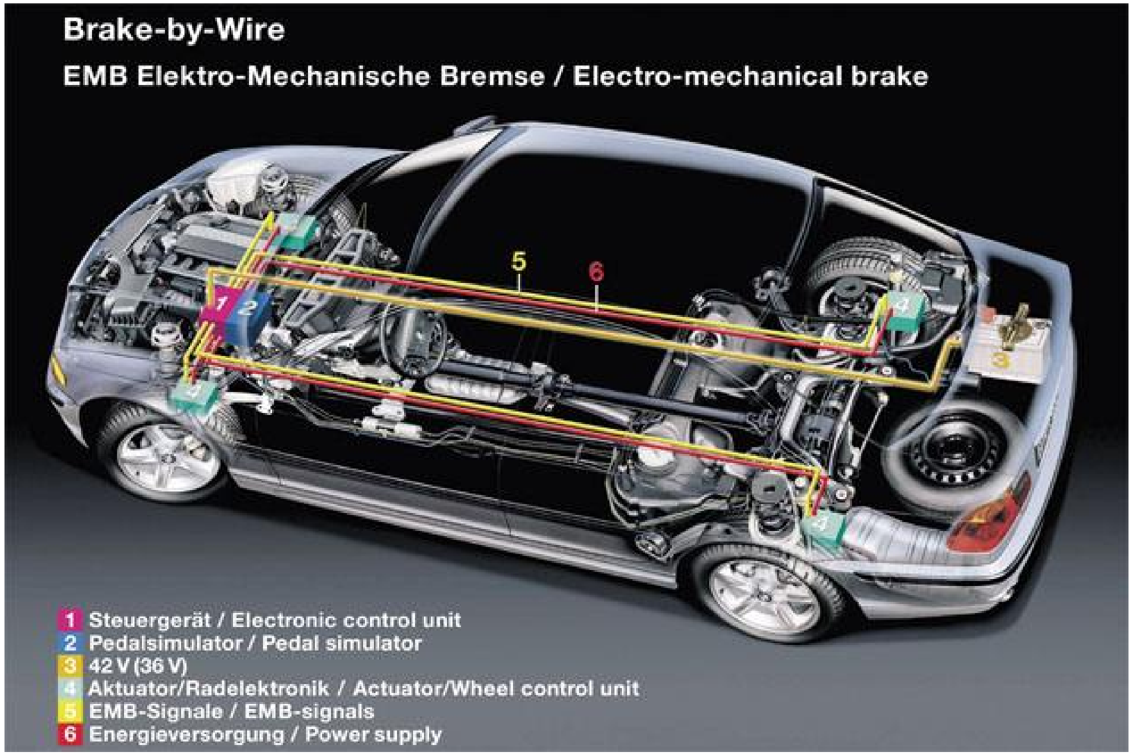
→ Trivial electrical failures (battery, generator, wiring harness, spark plugs, connectors, …) cause > 50% of all problems

→ The more E/E functionality a car has, the more likely E/E failures will occur

→ **Reliability** as part of product quality is an important issue

Concept study for a brake-by-wire system:

- Actuation of the brakes by electromagnetic actuators controlled by an ECU
- Advantage: Avoids the hydraulic system
- Problem: High electrical peak power necessary



How safe is it compared to the proven hydraulical dual-circuit braking system with mechanical parking brake?

- What if the battery is discharged?
- What if one of the wires is broken?
- What if the software in the ECU crashes?

→ **Safety** analysis necessary

Source: Article "by-wire - …", www.heise.de/autos

Safety and Reliability

- driven by customer expectations (quality) and legal requirements (state of the art)
- trade off between cost and technical perfection

Some Technical Terms

- **Reliability** and **Dependability** (in German: Zuverlässigkeit und Verlässlichkeit)
  Capability to perform its function as desired, i.e. without failure in any of its components. In newer publications *Reliability* is used for the quantitative aspect, whereas *Dependability* is used for the qualitative aspect.
- **Safety** (in German: Sicherheit)
  Capability to operate without endangering people, goods or data. When a failure occurs, a safe system may switch into a safe state, e.g. with no or limited functionality (**Fail Safe**), or may continue to work using redundant components (**Failure Tolerant – Fail Operational**).
- **Security** → Not discussed in this module (in German: Zugangsschutz)
  Capability to allow access only to authorized users, keep information confidential, ...

- **Fault** e.g. leakage in brake hydraulics (in German: Fehler, Störung, Defekt)
  A hardware component **defect** or a software **bug**.
- **Failure** e.g. car does not brake (in German: Ausfall, Fehlverhalten)
  A fault's **system level effect**, may occur immediately or delayed until the faulty component or the buggy function is used.
- **Hazard** (in German: Gefährdung)
  The potential of a failure to injure or kill men or damage or destroy goods or data.
- **Risk** (in German: Risiko)
  Assessment of failures based on a combination of hazard potential and failure probability.

Classification of Failures

- **Failure types/modes** = What kind of failure?
  - Parameter failure: The basic function is still available, but one or more parameters are outside their specified range — e.g. engine performance low
  - Functional failure: A function does no longer work. — e.g. gear box stuck at 1st gear
  - Total failure: System does no longer operate. — e.g. engine stall
- **System level effect** = Is the failure safety critical?
- **Failure duration** = How long does the failure occur?
  - Permanent failure — e.g. if a fuse did burn out
  - Temporary failure — e.g. intermittent contact
- **Failure probability** = How often does the failure occur?
  - Systematic failure: All devices do contain the same failure, the failure is reproducible if the same operating conditions are applied — e.g. most software bugs
  - Random (stochastic) failure: Failure, which does not occur in all devices due to tolerances in manufacturing processes. — e.g. most hardware failures
- **Failure occurrence** = When does the failure occur?
  - Early failures (in the automotive industry: 0 km failures) — e.g. manufacturing failure
  - Mid of life failure (in the automotive industry: field failures) — e.g. 0.02% die @age 20…30
  - End of life failures: Failure due to natural wear out — e.g. tires

- **Failure cause** =Why does the failure occur?
  - Design error (=systematic) e.g. wrong material used
  - Manufacturing error (=stochastic (in most cases)) e.g. process tolerances
  - Operator error e.g. shifting gears without pressing clutch pedal
  - Overstress, overloading e.g. due to operator error or design error
  - Wear out e.g. operator error (forgot maintenance!), design error or accepted feature

Dealing with failures

- **Design Phase**
  - Analyze reliability and availability, especially considering safety related failures
  - Use proven design methods and processes, avoid complicated designs
  - Increase reliability by using reliable components
  - Increase availability and safety by including redundancy
- **Operation Phase**
  - **Monitor** to detect failures
  - **Act** to switch the system into a redundant operating mode or into a fail safe state
  - **Notify** the user (either immediately or by error logging)

→ **Functional Safety**

Measuring reliability

- **Mean Time to Failure MTTF**: Average time of operation, until a failure will occur

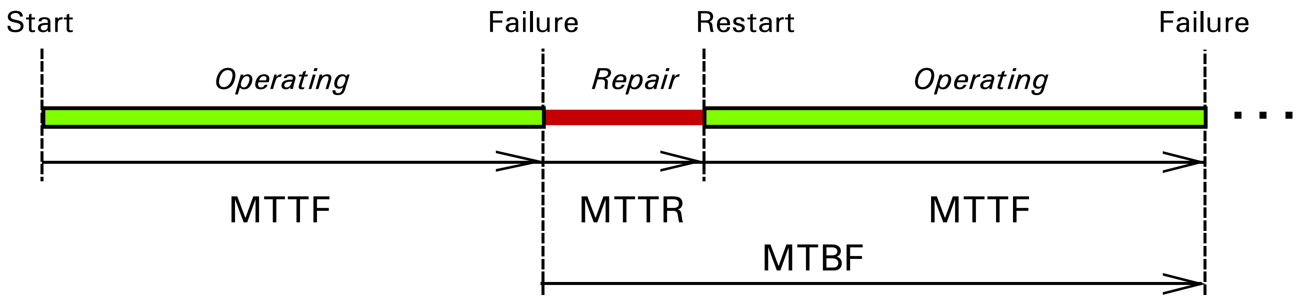
  If n out of N devices fail at times $t_1, t_2, \ldots, t_n$ within test time T:

$$\mathbf{MTTF} = \frac{t_1 + t_2 + \ldots + t_n + (N-n) \cdot T}{N}$$

if n=0: MTTF = T

- **Mean Time to Recovery (Repair) MTTR**: Average time to repair a failed system. Time for preventive maintenance is treated like repair time.

- **Mean Time Between Failures MTBF** = MTTR + MTTF

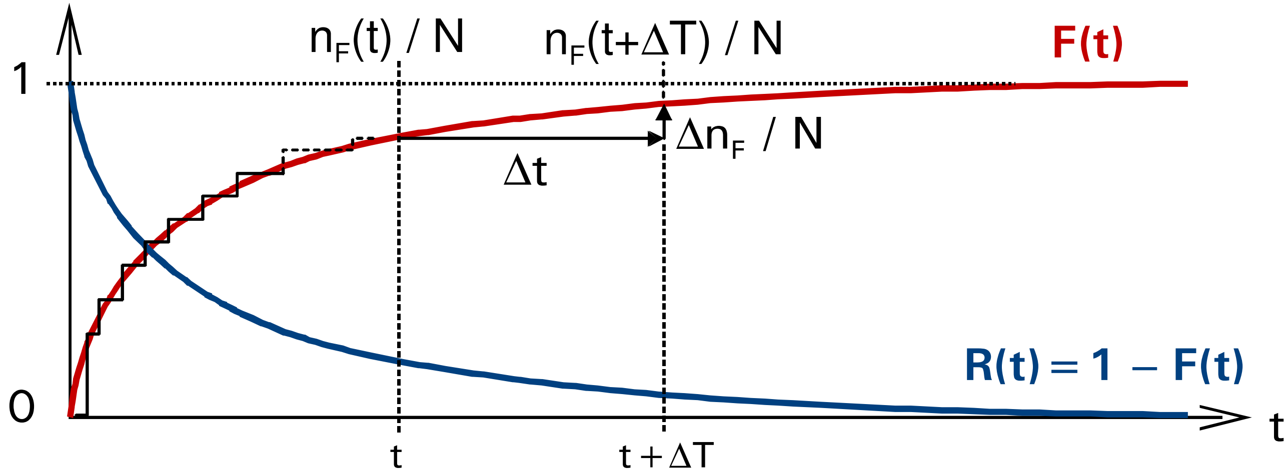


- **Availability** $\quad AV = \frac{MTTF}{MTBF} = 1 - \frac{MTTR}{MTBF} = \frac{MTTF}{MTTF + MTTR}$

Example: A system has AV = 99% → within 1 year the system is not available for 0.01 · 8760 hrs = 87.6 hrs = 3.7 days

2 Reliability Calculus

■ **Failure Probability F**: Relative failures over time



with	N	total number of units
	$n_F(t)$	number of failed units at time
	$n_R(t) = N - n_F(t)$	number of remaining units at time t.

If N is big enough and failures are statistically independent, probabilities can be calculated:

$$F(t) = \frac{n_F(t)}{N}$$ Failure probability with $F(t=0) = 0$ and $0 \le F(t) \le 1$ (1)

$$R(t) = \frac{n_R(t)}{N} = 1 - F(t)$$ Reliability = probability, that a unit is still good at time t (2)

■ **Failure Density**

$$f(t) = \lim_{\Delta T \to 0} \frac{n_F(t+\Delta T) - n(t)}{\Delta T \cdot N} = \frac{dF(t)}{dt} = -\frac{dR(t)}{dt} \qquad (3)$$

■ **Failure Rate**

$$\lambda(t) = \lim_{\Delta T \to 0} \frac{n_F(t+\Delta T) - n(t)}{\Delta T \cdot [N - n_F(t)]} = \lim_{\Delta T \to 0} \frac{n_F(t+\Delta T) - n(t)}{\Delta T \cdot N \cdot [1 - n_F(t)/N]} = \frac{f(t)}{R(t)}$$

$$= \frac{-1}{R(t)} \cdot \frac{dR(t)}{dt} = \frac{1}{1-F(t)} \cdot \frac{dF(t)}{dt} \qquad (4)$$

Probability, that one of the remaining devices fails in time interval t … t+ΔT (PFH … Probability of Failure per Hour)

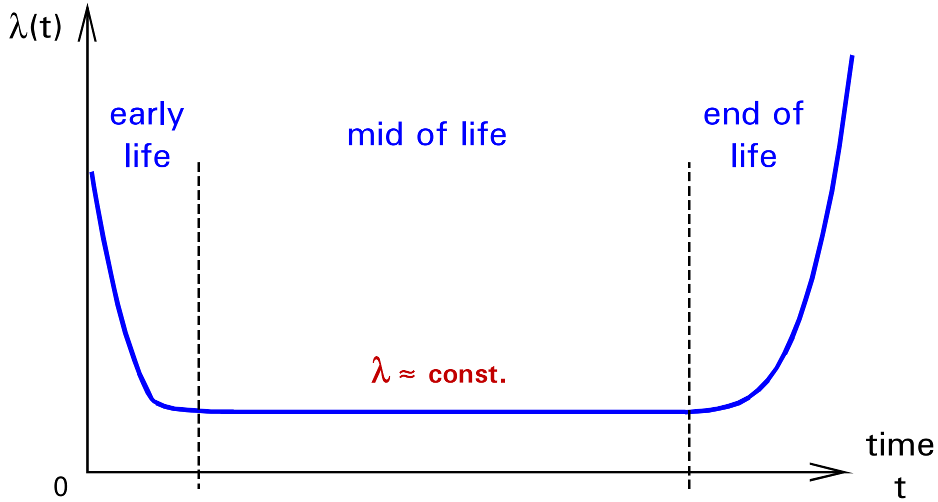
Relation between F(t), R(t) and λ(t):

From eq. (4) $\quad \lambda(t) \cdot R(t) = -\frac{dR(t)}{dt} \rightarrow \lambda(t) \cdot R(t) + \frac{dR(t)}{dt} = 0 \quad$ with $R(t=0) = 1$

$$\rightarrow \quad R(t) = e^{-\int_0^t \lambda(\tau)\, d\tau} \quad (5) \qquad \text{and} \qquad F(t) = 1 - R(t) \quad (6)$$

$$\rightarrow \quad MTTF = \int_0^\infty t \cdot f(t)\, dt = \int_0^\infty R(t)\, dt \qquad (7)$$

Typical failure rate
over product lifetime
(bathtub curve)

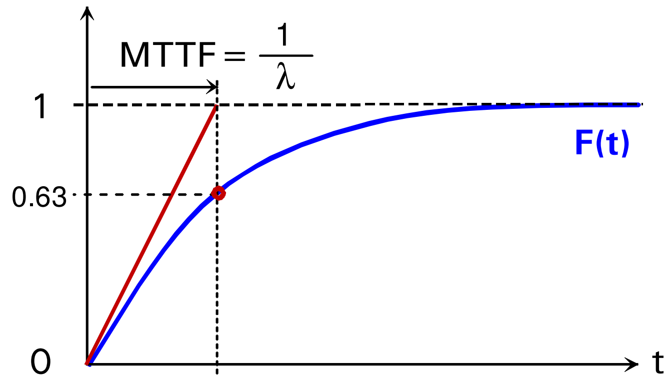


Exponential Distribution

During the **mid of life phase** often the failure rate can be considered to be constant

- if λ **= const.:** $R(t) = e^{-\lambda \cdot t}$ (8a) $F(t) = 1 - e^{-\lambda \cdot t}$ (8b) $MTTF = \frac{1}{\lambda}$ (8c)
- if $\lambda \cdot t \ll 1$ (and λ = const.) this can be simplified to

$R(t) \approx 1 - \lambda \cdot t$ $F(t) \approx \lambda \cdot t$



If you use the linear approximation, please be careful to ensure

$0 \le R(t), F(t) \le 1.$

The approximation error for F(t) is less than 10% for $\lambda \cdot t < 0.1$

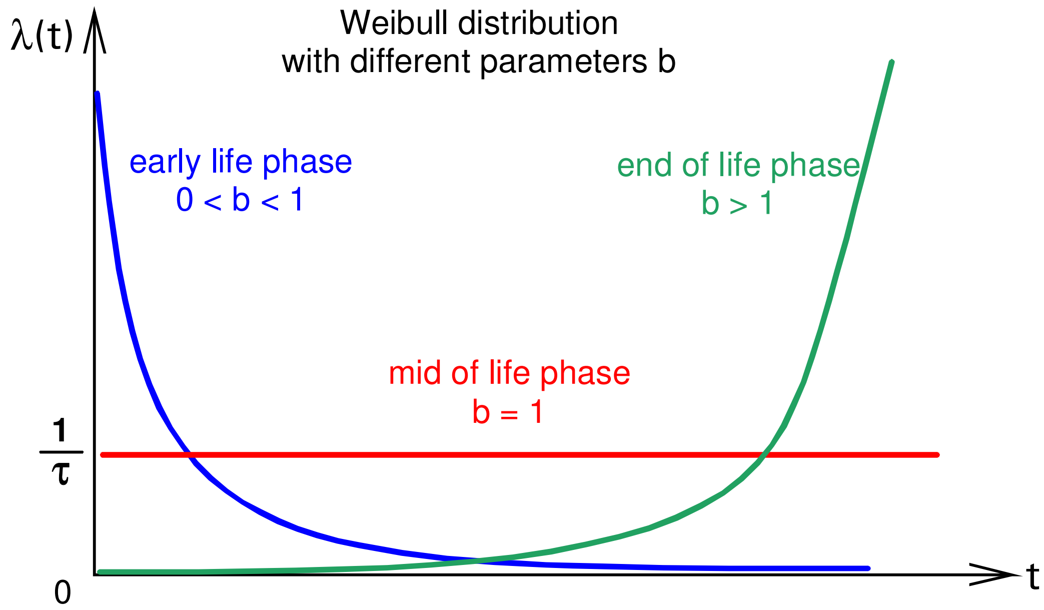
Weibull-Distribution

The 3 segments of the bathtub curve can be approximated by

$$\lambda(t) = \frac{b}{\tau} \cdot \left(\frac{t}{\tau}\right)^{b-1}$$

with parameters b and τ.

Note:
Each of the 3 segments of the bathtub curve has its own set of parameters b, τ.



For an arbitrary Weibull distribution:

$$R(t) = e^{-(t/\tau)^b}$$

$$MTTF = \tau \cdot \Gamma\left(\frac{b+1}{b}\right)$$

with the Gamma-Funktion

b	0.5	1	2	5
$\Gamma(\frac{b+1}{b})$	2	1	0.89	0.92

For b=1 the Weibull distribution is equal to the exponential distribution with λ=const. described by eq. (8a-c).

Typical units . . .

. . . for **failure probability F**:	**1 ppm**	**= 10^{-6} = 1 part per million**
	10000 ppm	= 1%
. . . for **failure rate**:	**1 fit**	**= 10^{-9}/h** = 1 ppm/1000h =1 failure per 10^{9} h

Example:

An electronic control unit (ECU) has a failure rate of λ = 500 fit?

How many ECUs out of a production batch of 5 Mio. ECUs will fail in the first year of continuous operation?

$$F = \lambda \cdot T = 500 \cdot 10^{-9} / \text{h} \cdot 365 \text{ days} \cdot 24 \text{ h} \approx 4400 \text{ ppm} = 0.44\%$$

$$n_F = F \cdot N = 4380 \text{ ppm} \cdot 5 \text{ Mio.} \approx 22000$$

How long will each ECU survive with 95% likelihood?

$$R = e^{-\lambda \cdot T} = 0.95 \quad \rightarrow \quad T = -\frac{1}{\lambda} \cdot \ln 0.95 = 100\,000 \text{ h} \approx 12 \text{ a}$$

3 Reliability Prediction for Electronic Circuits

Assumptions

- Only **random errors** which occur during normal operation of the device are considered, **no systematic errors** in design or manufacturing.
- Only **mid of life failures** are considered, assuming $\lambda \approx$ const. (no early/end of life failures)

Failure Rate Data for Components

- Failure rates **cannot be precalculated** from geometrical and material properties.
- Failure rates of electronic devices are **too small** (1 … 1000 fit) **to be measured by testing** individual devices, but can be statistically collected for classes of devices.
- Actual **failure rates are company secrets**. Publicly available data comes from US military (**MIL Handbook 217**), the Society of Automotive Engineers (SAE 870050), Bellcore (TR/SR-332) or the International Engineering Consortium (IEC 61709).
- MIL and others do publish **basic device failure rates** λ_B for nominal operating conditions. Different operating conditions (within the specified min – max range) can be taken into account by correction factors (stress factors): $\lambda = \lambda_B \cdot c_\vartheta \cdot c_M \cdot \ldots$

  temperature stress factor $c_\vartheta = 2^{\Delta\vartheta / 10K}$ with $\Delta\vartheta$ … difference between actual and nominal temperature (i.e. a 10K temperature increase doubles the failure rate (Arrhenius law)

  mechanical stress factor c_M = 0.5 … stationary, 1 … ground mobile, 2 … in flight

  voltage or current stress indirectly included via their temperature effect

  additional "stress factors" can be used to assess new technologies or other types of risk

Base failure rates (Estimates, based on MIL HDBK 217E)

Type	*Base rate* λ_B		*Type*	*Base rate* λ_B	
Semiconductors (active components):			*Passive components:*		
CMOS microcontroller	200 fit	+ 0.5 fit per pin [*1]	Metal file resistor	0.3 fit	+ 0.5 fit per pin [*1]
EPROM, RAM	100 fit		Film capacitor	0.5 fit	
CMOS logic IC	20 fit		Ceramic capacitor	0.3 fit	
Operational amplifier (OP)	50 fit		Aluminum capacitor	10 fit	
Small signal transis-tor/diode	0.5 fit		Inductor (coil, trans-former)	5 fit	
Power transistor/diode	50 fit		Quartz crystal	200 fit	
LED	25 fit				
Optocoupler	100 fit				
Electromechanical components (λ not temperature dependent)					
Switch	5 fit				
Relay	30 fit				
Connector per pin	5 fit				

Operating conditions: Temperature 45°C ambient, 85°C junction; ground mobile

[*1] Per pin failure rate and failure rate of mechanical components not temperature dependent.

Typical Operating Times

- Passenger cars 300h/year
- Trucks 2000h/year
- TV sets 1500h/year
- Telephones, fax 8760h/year
- Automation equipment 1 working shift = 2000h/year 3 working shifts = 6000h/year
- If operating time is not known, use calendar time (1d=24h, 1w=168h, 1y = 8760h)

MTTF estimation for circuits: Parts Count Method

- Sum of the failure rates λ_i of all i=1…n components: $\lambda_{circuit} = \sum_{i=0}^{n} \lambda_i$ $MTTF_{circuit} = \frac{1}{\lambda_{circuit}}$
- This method does not take into account, how a component fails (parameter drift, open or short circuit, …), which effect this failure has on the circuits overall function and how this failure propagates to the system level.

Early Life Failures

- Can be reduced by **burn in** or **run in**

4 Reliability Prediction for Software

- Software bugs have a systematic nature, i.e. they are included in each copy of the software. However, often they occur under very special, rare operating conditions only. Thus, they may also be described by statistical methods.
- **Development phase software reliability prediction** is difficult or impossible, as all software bugs are development errors. Development quality is highly dependent on application area, software complexity, developers' experience and available time. Not much freely available statistical data does exist. As rule of thumb the following data has been published:

Quality status	*Bugs per 1000 LOC*	*CMMI Level*
Untested software	250	-
Unusable software	> 10	-
Faulty software	< 10	0
Unstable software	< 6	1
Mature software	< 3	2, 3
Stable software	< 1	4, 5

LOC … programming language source code statement, empty lines and comments not counted.

A typical software test (Code review, inspection or test step does find 30% of all bugs, i.e. 70% of all bugs survive each test step).

CMMI … Capability Maturity Model Integration is a formal method to assess the quality of software development processes, proposed by Carnegie Mellon University, Pittsburg, USA. A similar assessment exists in Europe as ISO 15504 Automotive Software Process Improvement and Capability Determination (SPICE).

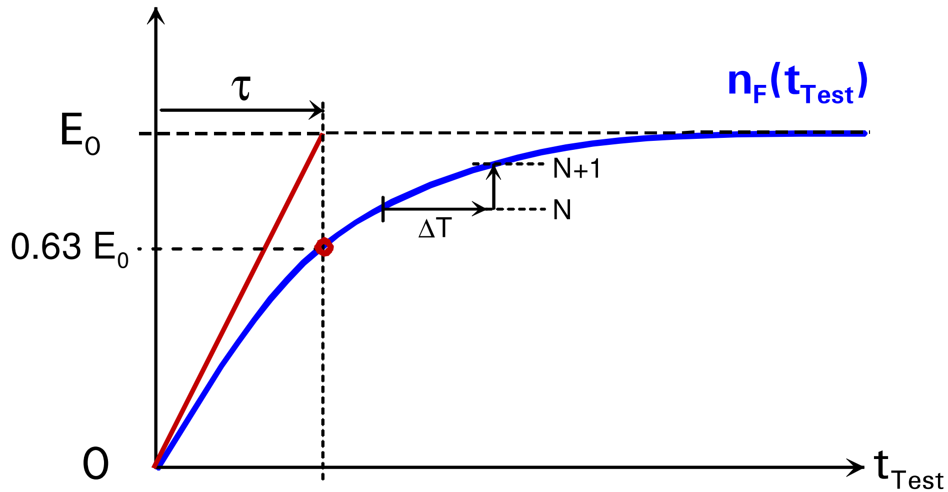
Prediction based on Software Test Statistics: John Musa’s Software Reliability Model

… assumes, that

- at begin of the software tests, the software has a finite, but unknown number of bugs E_0
- during tests bugs are found and removed according to an exponential time function

$$n_F(t_{Test}) = E_0 \left(1 - e^{-t_{Test}/\tau} \right)$$

where τ is an unknown parameter describing the intensity of the tests



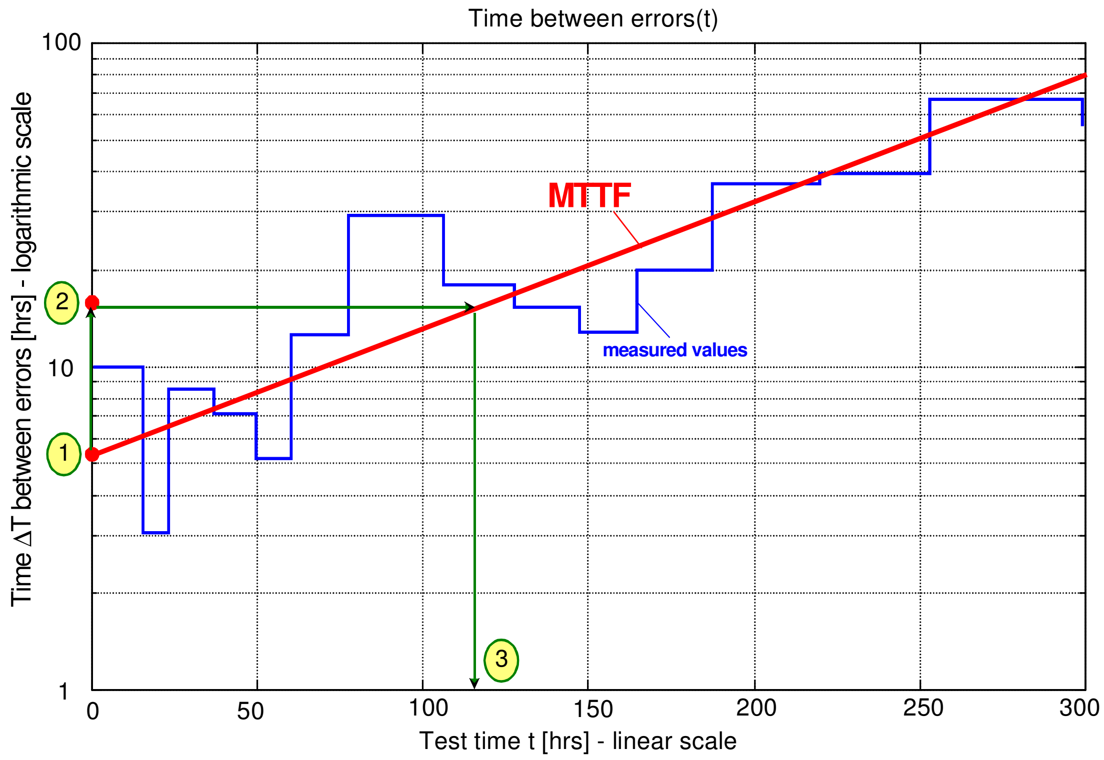
After bug N has been found and fixed, it takes time ΔT to find the next bug N+1. The rate, at which bugs are found, can be calculated as $\dfrac{N+1-N}{\Delta T} = \dfrac{1}{\Delta T} \approx \dfrac{dn_F}{dt_{Test}} = E_0 \cdot \dfrac{1}{\tau} \cdot e^{-t_{Test}/\tau}$

From this equation, we can calculate: $MTBF \approx MTTF \approx \Delta T = \dfrac{\tau}{E_0}\ e^{t_{Test}/\tau}$

i.e. when testing and fixing bugs, the MTTF increases exponentially with test time.

When testing, the times $\Delta T \approx MTBF$ between bugs can be recorded and plotted. When using a diagram with linear axis, the plot should show a step approximation of the above exponential function $MTTF \approx \Delta T = \frac{\tau}{E_0} e^{t_{Test}/\tau}$

By computer-aided curve fitting, the unknown parameters E_0 and τ can be found. For manual analysis, it is better to use a semi-logarithmic plot. In such a semi-logarithmic plot, an exponential function turns into a straight line, which simplifies curve-fitting drastically:



Step 0:

- Draw the straight line approximation for MTTF. Note: The areas above and below the red line should be balanced.

Step 1:

- Get the value $MTTF(t_{Test}=0)$ from the diagram.

Step 2:

- Calculate
  $MTTF(t_{Test}=\tau) = MTTF(t_{Test}=0) \cdot e$

Step 3:

- Get the value τ from the diagram.

Step 4:

- Calculate $E_0 = \dfrac{\tau}{MTTF(t_{Test}=0)}$

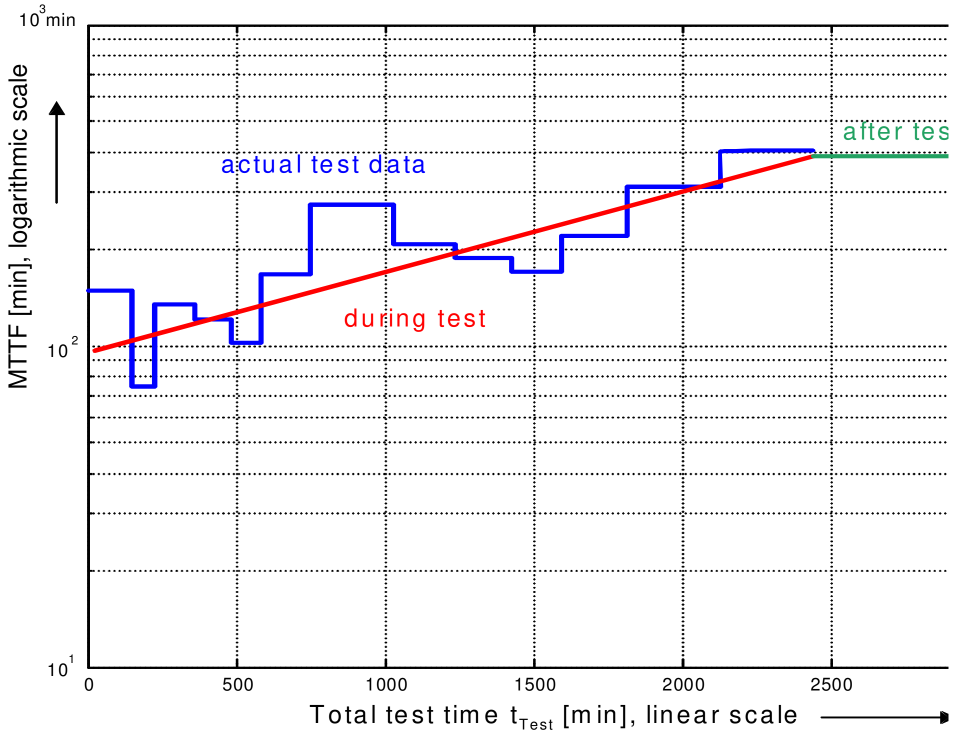
- **Are Musa's assumptions valid?**
  - Testing must be done under conditions similar to later use of the software. Simulate Env.
  - Bugs which were found, must be removed without inserting new bugs. Questionable
  - Test time to find the latest bug is a measurement value of the current MTTF. The more bugs were removed, the more time is necessary to find the next one, i.e. MTTF does increase exponentially with test time. Long test times, need test automation
  - Once the software is delivered, the MTTF = $1/\lambda$ does not change anymore. OK

Example:

Bug #	Bug found after = total test time t_{Test}	Test time between bugs = MTTF
1	149 min	149 min
2	224 min	75 min
3	359 min	135 min
4	480 min	121 min
5	582 min	102 min
6	750 min	168 min
7	1027 min	277 min
8	1235 min	208 min
9	1423 min	188 min
. . .	. . .	. . .

During test: $\text{MTTF} = \tau / E_0 \cdot e^{t_{Test}/\tau}$

E_0, τ are unknown parameters (see appendix)



5 Analyzing System Safety 1: Fault Tree Analysis FTA

- Not all failures are safety related, thus the parts count approach is too pessimistic with respect to safety.
- To get a more reasonable estimate, we should distinguish between "safe" (uncritical) and dangerous failures:

$$\lambda_{total} = \lambda_S + \lambda_D = \lambda_S + \lambda_{DD} + \lambda_{DU}$$

Dangerous failures (λ_D) can be further divided into detectable (λ_{DD}) and undetectable (λ_{DU}) ones. Assuming, that for detectable failures appropriate countermeasures can be taken before a safety critical situation occurs, the remaining safety risk is given by λ_{DU}.

If no better estimate is available, IEC 61508 recommends to assume $\lambda_D = 0.5\ \lambda_{total}$. This standard also defines the **Safe Failure Fraction SFF** $= (\lambda_S + \lambda_{DD}) / \lambda_{total}$.

To find out, which failures are safety related, a cause and effects analysis is needed:

- **Fault Tree analysis FTA**

  Top down approach: Identify system level safety critical events and track them down to component failures

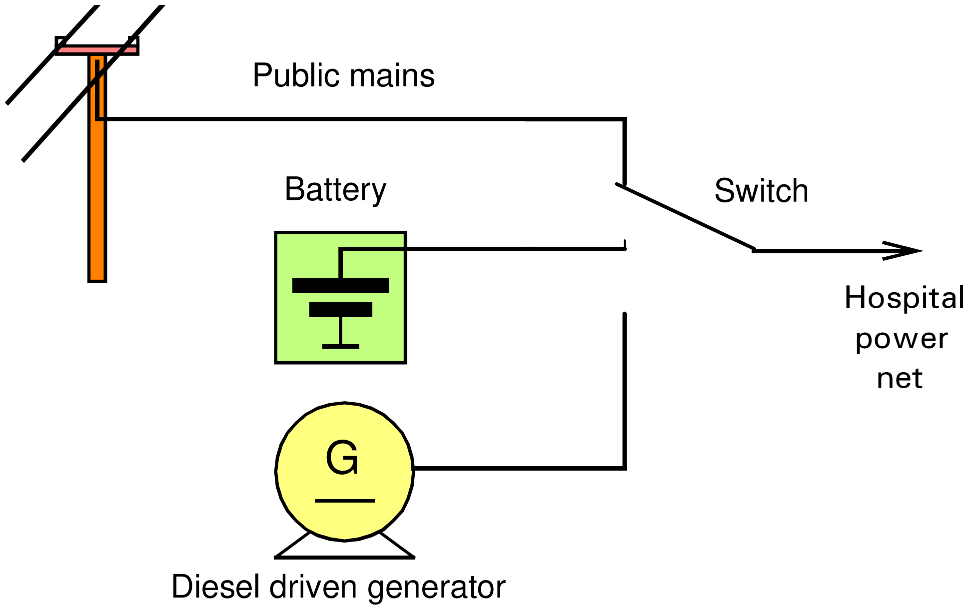
  Standardized by IEC 61025, DIN 25424 and various industry standards, e.g. SAE ARP 4761

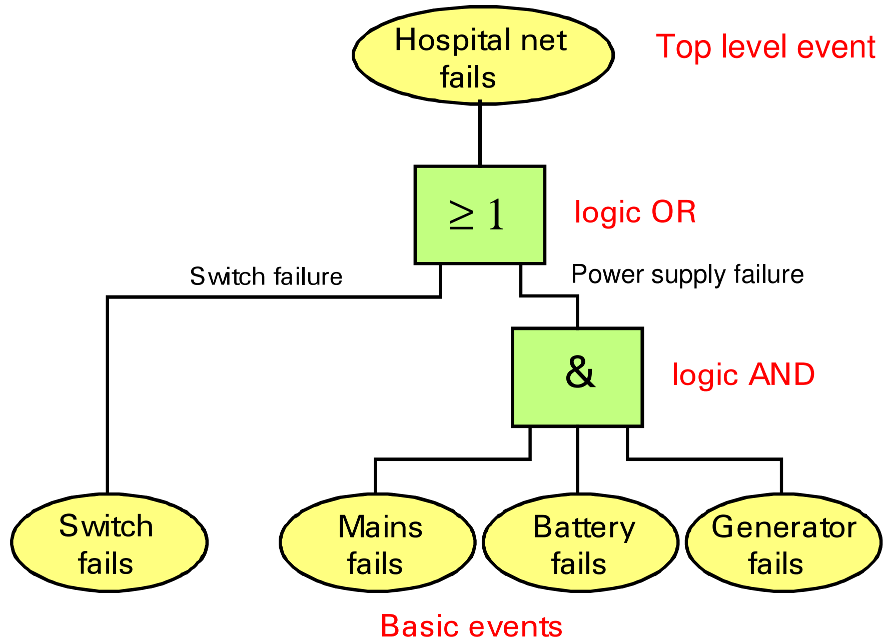
- **Failure Modes and Effects Analysis FMEA**

  Bottom up approach: Identify component failures & track their effect up to system level

  Standardized by IEC 60812, DIN 25448 and various industry standards, e.g. SAE J1739

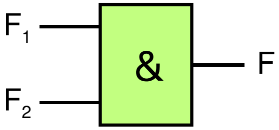
FTA example: Electrical power supply for a hospital





- **Fault trees describe**, which **system failure (=top level event)** does have which **root cause(s) (=basic event)**. Basic events typically are component failures.
- **Basic events must be independent** on each other
- If a failure has more then one cause,
  - **causes** are **logically ANDed**, if the system fails only, when all causes occur,
  - **causes** are **logically ORed**, if the system fails already, when one of the causes occurs.
- A typical fault tree has several top level events and a multi-level hierarchy of ANDs and ORs

- **System level failure probability** as a function of component failure probabilities:



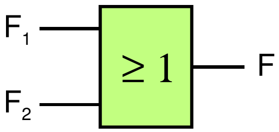
Logical AND

Failure event F occurs, if failure F_1 AND failure F_2 do occur

$F = F_1 \cdot F_2$

$R = 1 - F_1 \cdot F_2 = 1 - (1-R_1) \cdot (1-R_2)$
$= R_1 + R_2 - R_1 \cdot R_2$

Special case $F_1 \approx 1$, $F_2 << F_1$ → $F \approx F_2$



Logical OR

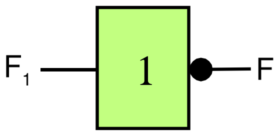
Failure event F occurs, if failure F_1 OR failure F_2 or both do occur

$F = 1 - R_1 \cdot R_2 = 1 - (1-F_1) \cdot (1-F_2)$
$= F_1 + F_2 - F_1 \cdot F_2$

$R = R_1 \cdot R_2$

Special case F_1 or $F_2 \approx 1$ → $F \approx 1$

Special case F_1, $F_2 << 1$ → $F \approx F_1 + F_2$



Logical NOT

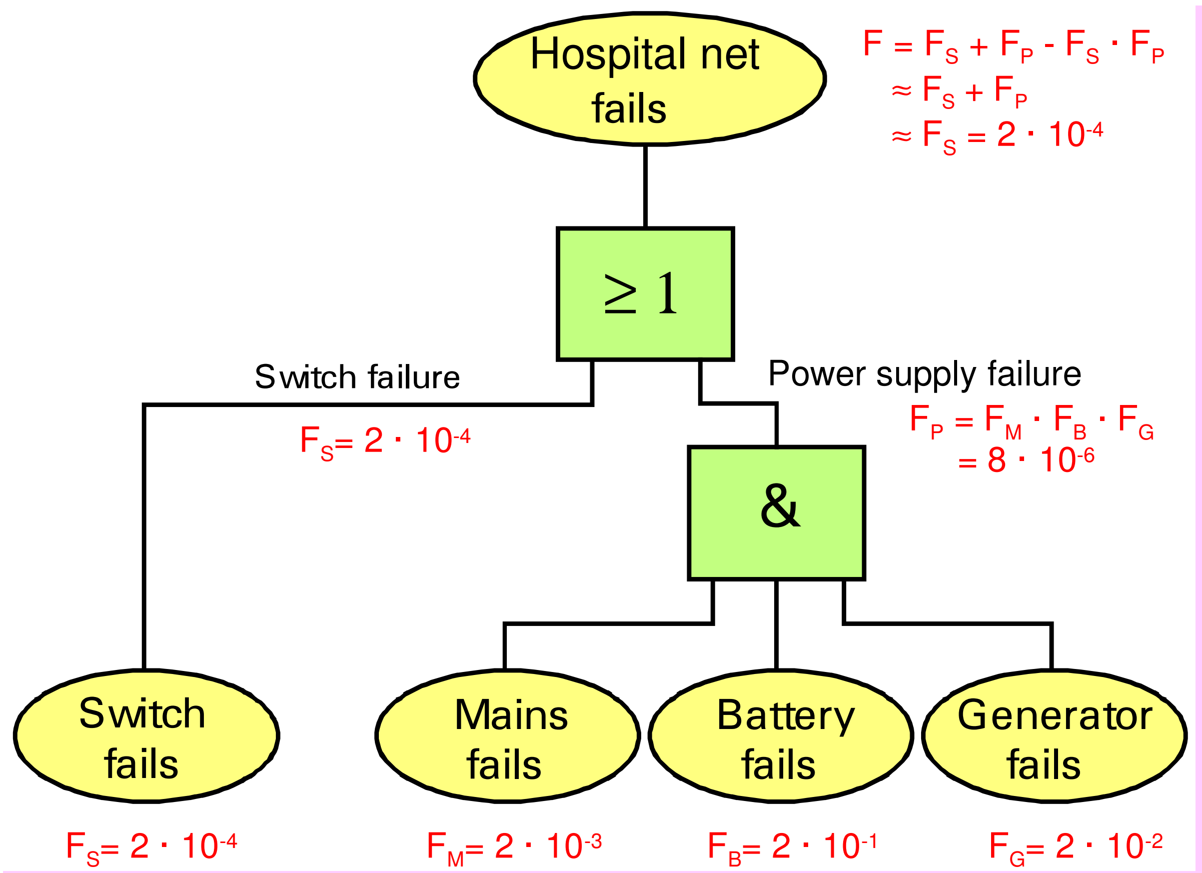
Event F occurs, if event F_1 does not occur

(NOT should not be used in FTAs, if possible)

$F = 1 - F_1$

$R = 1 - R_1$

Example continued:

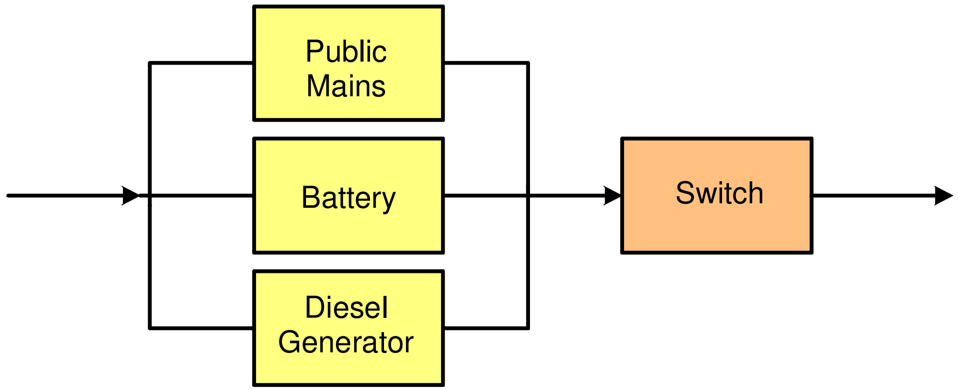


Example Data	λ	$F=1 - e^{-\lambda \cdot T}$ for T=168h (1w)
Switch	10^{-6}/h = 1 kfit	200ppm = $2 \cdot 10^{-4}$
Mains	10^{-5}/h = 10 kfit	2000ppm = $2 \cdot 10^{-3}$
Generator	10^{-4}/h = 100 kfit	20000 ppm = $2 \cdot 10^{-2}$
Battery	10^{-3}/h = 1 Mfit	200000ppm = $2 \cdot 10^{-1}$

Reliability Block Diagrams

- Used as an alternative to fault trees FTA
- Describes which blocks of a system are involved in providing a certain functionality. Required blocks are connected in series, alternative blocks are connected in parallel.

  E.g.: Reliability block diagram for the electrical power supply of a hospital



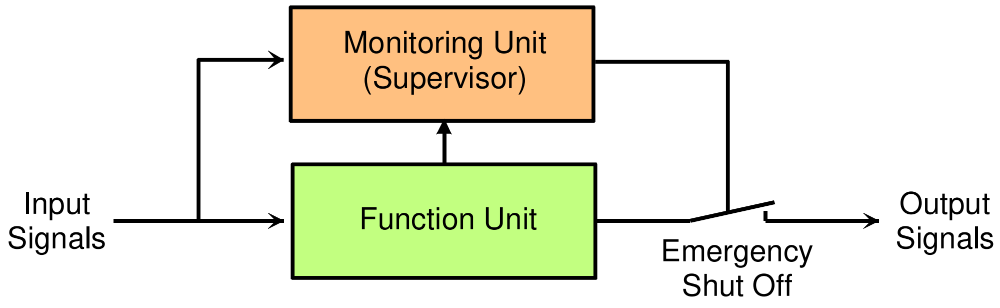
Note: In a reliability block diagram the same block can occur several times if the functional logic requires it.

- The same mathematics as for FTA applies:

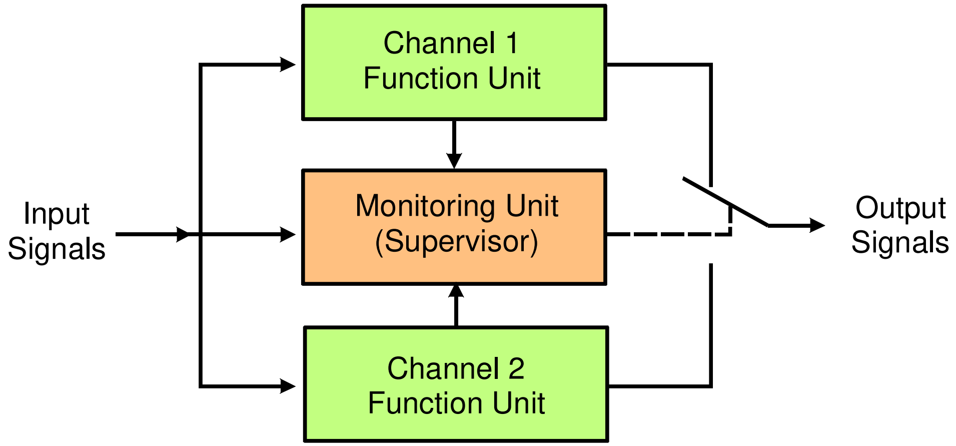
	Fault Tree	Reliability Block Diagram
System fails, if any of the blocks fails	OR gate	Series connection
System fails, if all of the blocks fail	AND gate	Parallel connection

6 System Monitoring, Redundancy and Diversity

Structure of Safety Critical Systems



1 out of 1 system (1oo1): 1 functional unit



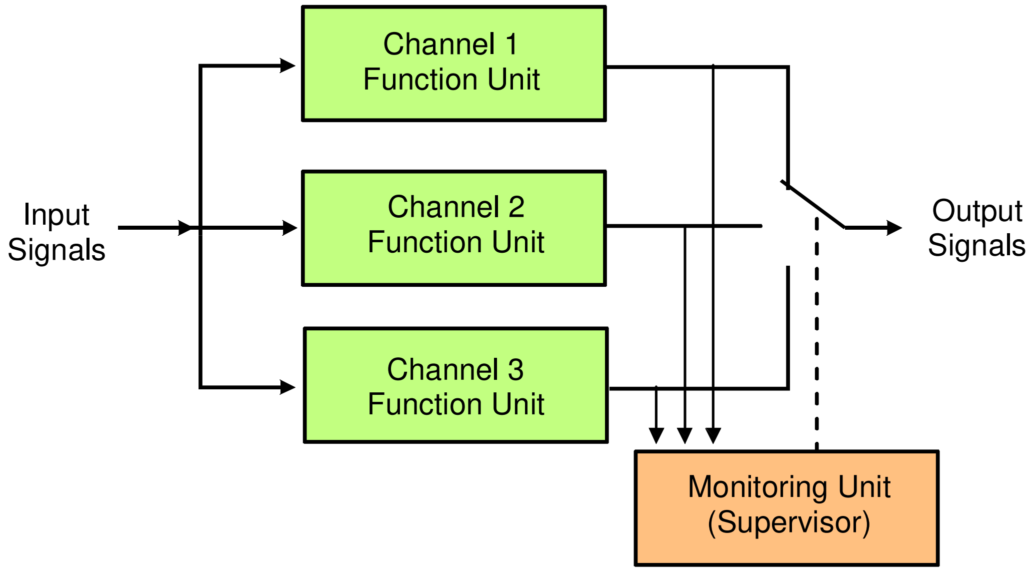
1 out of 2 system (1oo2):
2 functional units, 1 required for operation
(DMR Dual Modular Redundancy)

Independent monitoring unit:

- Detect failures
- Act in case of failures, e.g. shut down (**fail safe**), reduce functionality (**graceful degradation**, **limp home**) or switch to redundant unit (**fault tolerant – fail operational**)

Redundant channels operation:

- **Cold stand-by**
  - Redundant channel 2 normally is off, will be turned on only in case of failure of channel 1
- **Hot stand-by**
  - Redundant channel 2 is permanently operating in parallel to channel 2
  - Improved failure detection, because channel 1 and channel 2 outputs can be compared. However: Comparison alone does not allow to find out, which channel failed.



In hot stand-by operation, **monitoring by majority voting** possible:

- Channel outputs are compared
- If one channel's output differs from the two others, this channel is considered to be faulty

1 out of 3 system (1oo3):
3 functional units, 1 required for operation
(TMR Triple Modular Redundancy)

HFT = 2

2 out of 3 system (2oo3):
3 functional units, 2 required for operation

HFT = 1

Hardware Failure Tolerance HFT: Number of hardware failures a system can tolerate without becoming unsafe.

Analysis of these structures: see next page

⇒ Switch causes a single point failure
⇒ Monitoring does improve safety
⇒ Redundancy does improve availability

Failure Rates of Redundant Structures

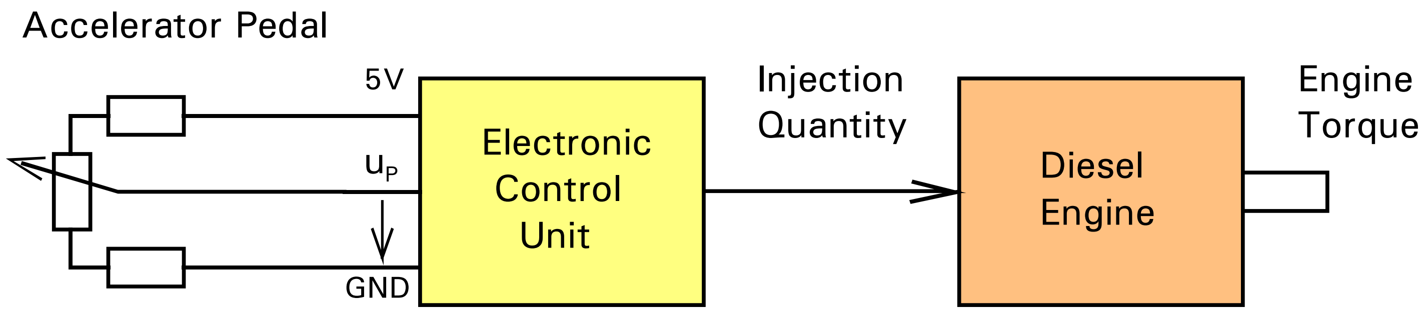
- Assumptions: Identical channels, no common mode failures, $F_{monitor}$, $F_{switch} << F_{channel} << 1$
- F_{REL} Probability, that some failure occurs, no matter how severe
- F_{AV} Probability, that the system cannot be used
- F_{SAF} Probability of an undetected failure (assume: system switched off not safety critical)

1oo1 without monitoring and switch	$F_{REL} = F_{AV} = F_{SAF} = F_{channel}$
1oo1 with monitoring and switch	$F_{REL} \approx F_{AV} = F_{channel} + F_{monitor} + F_{switch}$ $F_{SAF} \approx F_{channel} \cdot F_{monitor} + F_{switch}$
1oo2	$F_{REL} \approx 2\ F_{channel} + F_{monitor} + F_{switch}$ $F_{AV} \approx F_{channel}^2 + F_{SAF}$ … no common cause failures $F_{SAF} \approx F_{channel} \cdot F_{monitor} + F_{switch}$
1oo3	$F_{REL} \approx 3\ F_{channel} + F_{monitor} + F_{switch}$ $F_{AV} \approx F_{channel}^3 + F_{SAF}$ $F_{SAF} \approx F_{channel} \cdot F_{monitor} + F_{switch}$
2oo3	$F_{REL} \approx 3\ F_{channel} + F_{monitor} + F_{switch}$ $F_{AV} \approx 3\ F_{channel}^2 + F_{SAF}$ $F_{SAF} \approx F_{channel} \cdot F_{monitor} + F_{switch}$

Monitoring Methods

- **Signal range check** SRC: Check if a signal is within its physical limits (normal operating range) ← feasible for all signals with know signal range

  Example: Engine control system

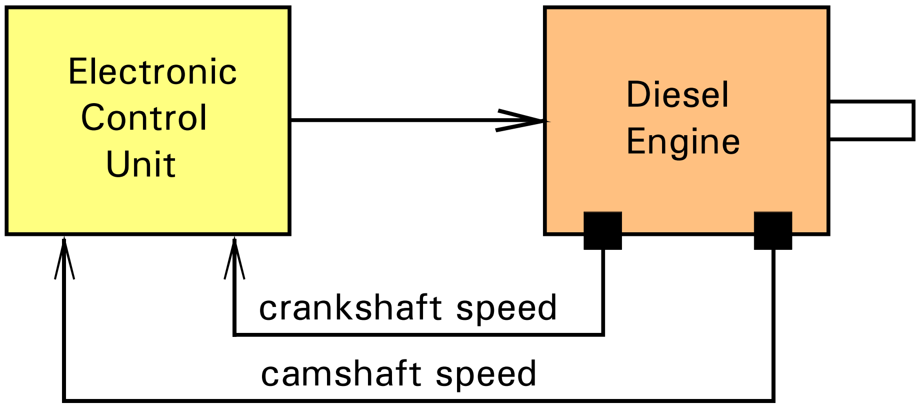


Normal operation: 0V min < **0.5V** idling ≤ u_p ≤ **4.5V** max. torque < 5V max

|← normal operating range →|

Possible Failure	u_p	Comment
5V wire broken	0	<0.5V failure can be detected
u_p wire broken	0	<0.5V failure can be detected
GND wire broken	5V	>4.5V failure can be detected
. . .	. . .	. . .

- **Static plausibility check**: Compare a signal with another signal, which has the same or a similar information content, e.g.



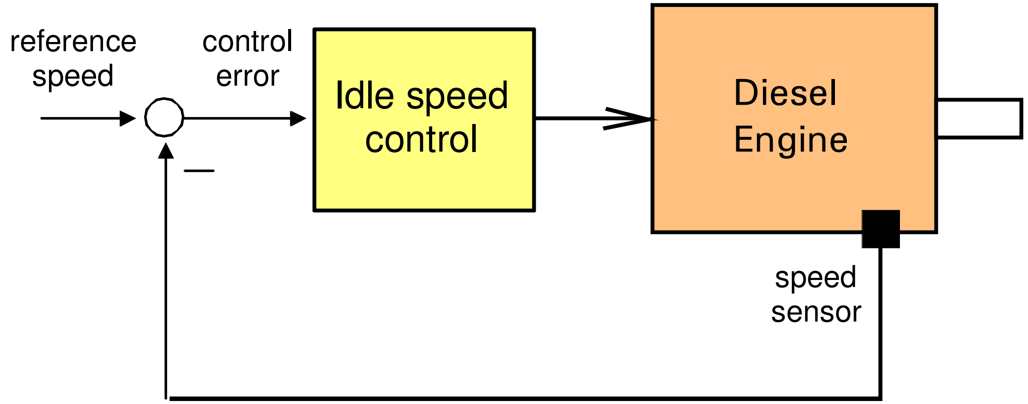
In a 4 stroke engine the camshaft speed is ½ of the (average) crankshaft speed.

If |camshaft speed – crankshaft speed/2| > 10% → failure of one of the speed sensors

- **Dynamic plausibility check**: Compare a signal's rate of change (slew rate) with the physical limits of its slew rate, e.g.
  - Typically the water temperature is monitored with 1 sample/sec
  - Typically the engine heats up or cools down with < 10°C / min
  - If the measured water temperature changes faster → sensor or wire failure

- **Short circuit and broken wire detection** for ECU output drivers and signal inputs

- **Steady state error check**: Check if steady state control error in closed loop systems is within tolerance, e.g.



- **Event sequence check**: Check the sequence of events and/or operator actions, e.g. to start a car with automatic transmission, the driver must
  - put the transmission in park position
  - start engine by turning the ignition key
  - put foot on the brake pedal
  - engage gear into D(rive) position

  . . .

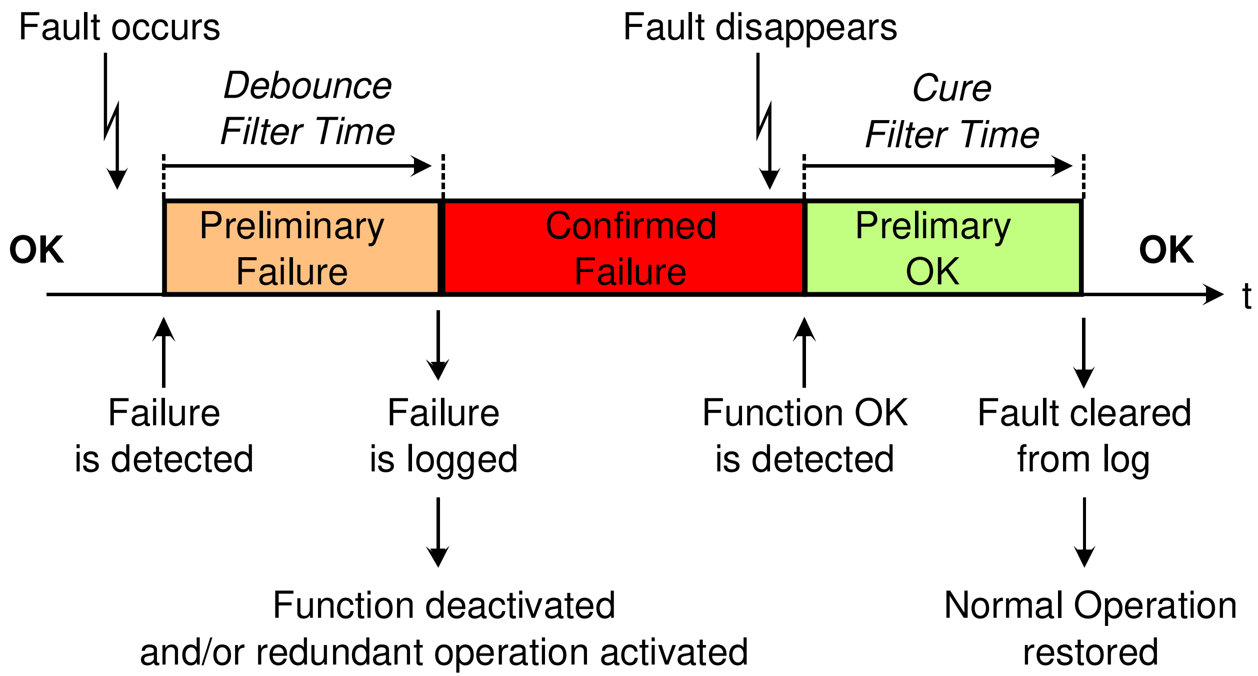
- **Runtime check**, e.g. watchdog
  - The monitored item, e.g. a microprocessor or a human operator, must periodically trigger the monitoring system ("watchdog").
  - If it fails to do so, the "watchdog" switches the system into a safe state and/or re-sets/restarts the monitored device.

- **Message timeouts and information redundancy** in data communication, e.g.
  - Parity and CRC checksums for digital data
  - Timeout monitoring for bus messages

    . . .

Typically, failures are detected and localized by a combination of several monitoring methods.

Because wrong detection is possible, debounce time filters and error counters are used:



Filter times for Function activation/deactivation and Error logging may be different.

Monitoring methods can be active

- during system start (power-on self-test)
- periodically (cyclic test, background test)
- continuously during normal operation
- on demand (test mode)

Quantitative description of monitoring quality:

Diagnostic Coverage for dangerous (safety critical) failures $DC = \frac{\lambda_{DD}}{\lambda_D} = \frac{\lambda_{DD}}{\lambda_{DD} + \lambda_{DU}}$

Diversity

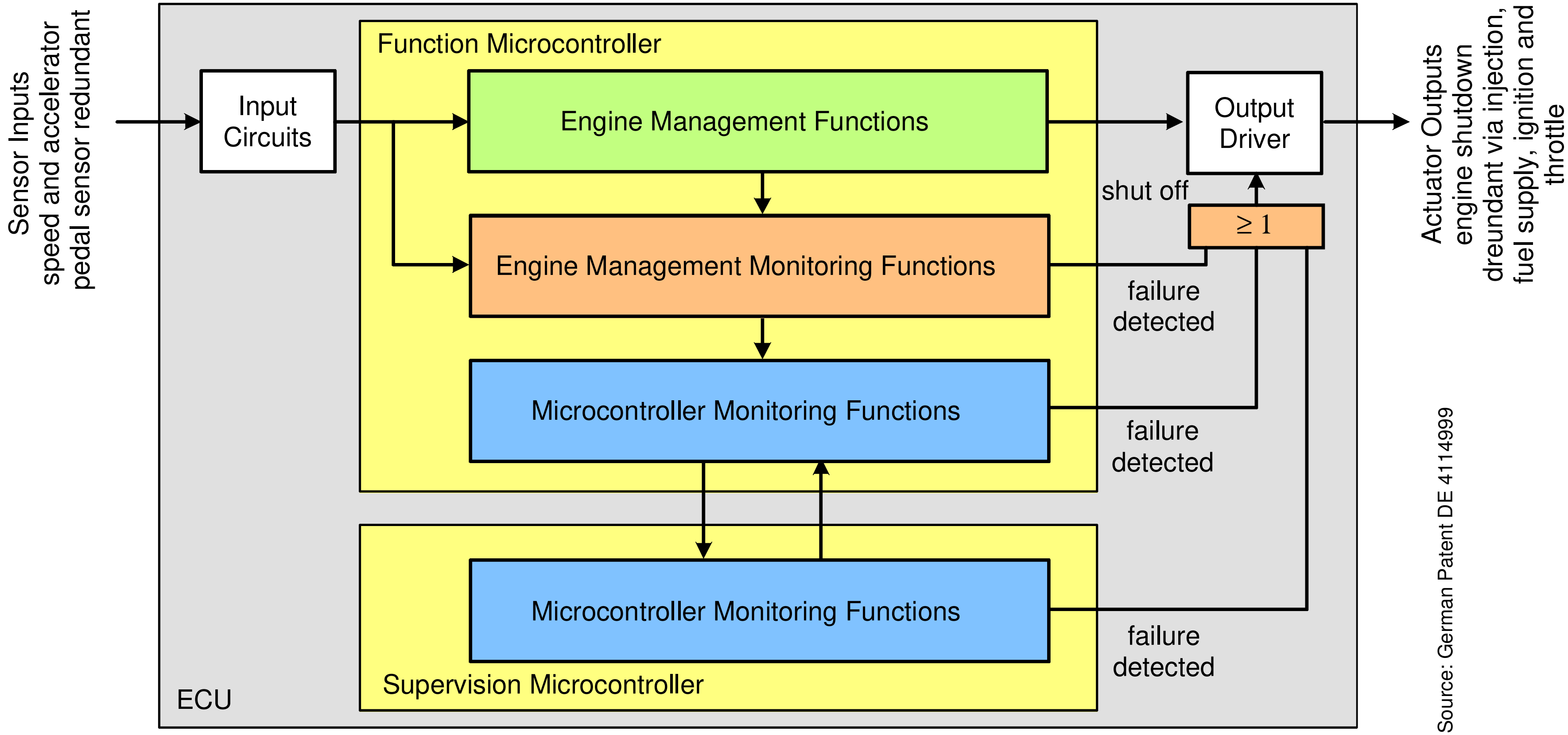
- Common cause failures can be avoided/reduced, if the redundant channels are built with
  - different technologies (e.g. electronic control with mechanical backup system)
  - different hardware and/or software and are
  - developed with different development tools and by different people.

e.g. elevators use an electrical drive but have a mechanical emergency brake

US Space Shuttle uses 4 different computer systems in main flight control system

- Disadvantage:
  very expensive: development effort for multiple systems, reduced economy of scale
- Remaining risk:
  Even diverse systems will be developed according to a common subset of requirement specifications

Redundant structure of Bosch engine management ECUs



- Supervision of engine management function via function software monitoring (signal range check, static and dynamic plausibility, timeouts, …)
- Basic self monitoring of microcontroller operation via software (watchdog, RAM check, …)
- Diverse, redundant monitoring of microcontroller hardware via supervisor microcontroller

7 Analyzing System Safety 2: Failure Modes and Effects Analysis FMEA

FTA is a **top-down** approach, which

- is of limited value, when failure probabilities are not know
- can't guarantee, that all component failures are investigated
- does not systematically look for failure detection and avoidance methods

FMEA is a **bottom-u**p approach, which

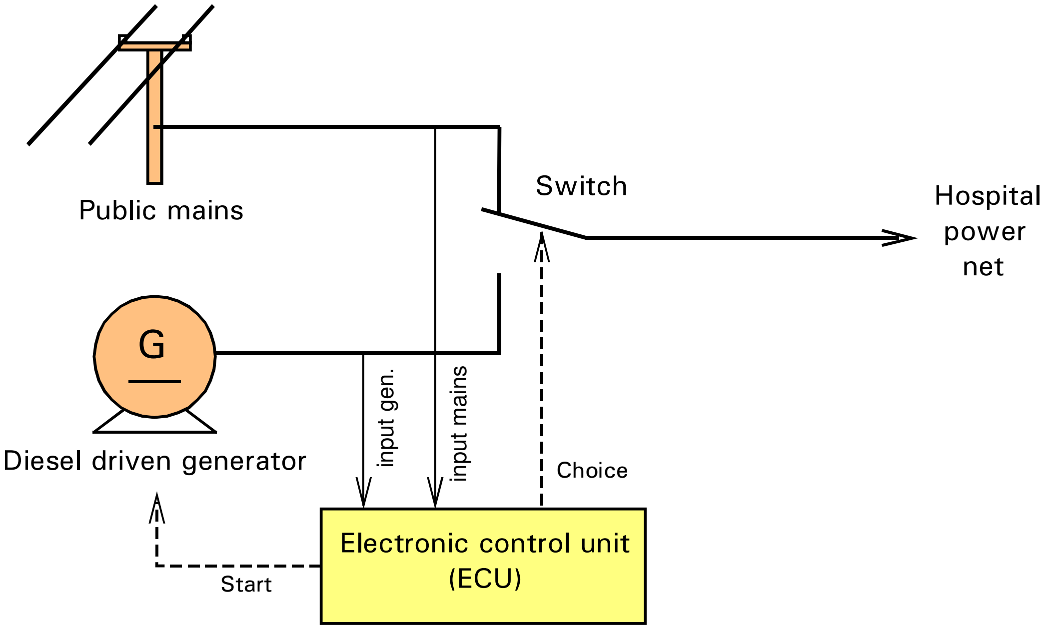
- systematically **tracks all component failures** up to their system level effect
- inherently **requires to develop failure detection** and avoidance methods
- bridges the gap between quantitative analysis and qualitative engineering know how
- allows to **assess failure risk** as a combination of probability and criticality

FMEAs can be used on various levels:

- **System Design FMEA**: Used when defining the system architecture, investigates subsystem and/or component interaction, discusses interface failures, but treats subsystems and/or components as black box
- **Component Design FMEA**: Used when designing a component, tracks internal failures up to the component interface
- **Process FMEA**: Used to analyze manufacturing (and other) processes

FMEAs require a recursive or iterative process all over the development cycle

FMEA example: Electrical power supply for a hospital



System function

- Normally the hospital net is powered by the mains.
- An ECU monitors the mains voltage. If the voltage is too high or too low, the ECU starts up a diesel driven generator and switches the hospital net to the generator.
- The generator's output voltage is also monitored. If this voltage is too high, the generator is stopped.

Step 1: Which **components** do we have?
Step 2: Which component failures can occur? (**Failure Modes**)
Step 3: What may **cause** these failures? → What can be done to avoid these failures?
Step 4: What is the system level **effect** of these failures?
Step 5: How can failures be **detected** and which **countermeasures** can be taken?
Step 6: What is the **rest effect** of these failures, when detected & countermeasures taken?
Step 7: Assess **criticality C**, **probability P** and **detectability D** of these failures on a scale from 1 … 10 and compute the **risk priority R = C · P · D**
Step 8: **Address** all **failures with high risk prority R**

#	Component $\boxed{1}$	Failure $\boxed{2}$	Possible cause $\boxed{3}$	Effect when not detected $\boxed{4}$	Detection / Countermeasure $\boxed{5}$	Effect when detected $\boxed{6}$	Risk priority $\boxed{7}$			
							C	P	D	R
1.1	Mains	voltage too high	bad control quality by mains operator	destroys electrical equipment connected	by voltage measurement in ECU → switch to generator	redundant power supply by generator	3	6	3	54
1.2		voltage too low	same as 1.1 broken wire	power loss in hospital net	same as 1.1	same as 1.1	3	10	3	90
2.1	Generator	voltage too high	failure in generator voltage controller	same as 1.1 (but only when generator is needed)	by voltage measurement in ECU → switch off generator, call service personnel	power loss in hospital net	10	5	3	150
2.2		voltage too low	same as 2.2	same as 1.2 (but only when generator is needed)	by voltage measurement in ECU → call service personnel	malfunction of electrical equipment, in worst case: power loss	9	5	3	135

#	Component $\boxed{1}$	Failure $\boxed{2}$	Possible cause $\boxed{3}$	Effect when not detected $\boxed{4}$	Detection / Countermeasure $\boxed{5}$	Effect when detected $\boxed{6}$	Risk priority assignment $\boxed{7}$			
							C	P	D	R
3	Switch	stick in pos. 1	wear out	cannot switch to generator (only when generator is needed)	None → ECU must measure switch output voltage too	same as 2.1	10	3	10	300
4.1	Diesel	does not start	• no fuel • starter battery not loaded • . . .	same as 2.2	same as 2.1	same as 2.1	10	7	3	210
4.2		runs too fast or too slow	failure in diesel engine speed control	hospital net frequency too high or too low → malfunction of electrical equipment	monitor frequency by ECU → call service personnel	same as 2.2	9	3	3	81
5.1	ECU mains in	open circuit	corrosion vandalism	ECU assumes mains voltage too low and switches to generator	signal range check in ECU → call service personnel	same as 1.1	3	5	3	45

# 2a	Component 1	Failure 2	Possible cause 3	Effect when not detected 4	Detection / Countermeasure 5	Effect when detected 6	Risk priority assignment 7			
							C	P	D	R
5.2	ECU gen. in	open circuit	same as 5.1	generator voltage and frequency cannot be monitored (only when generator is needed)	same as 5.1	none	2	5	3	30
5.3	start out	open circuit	same as 5.1	same as 4.1	same as 4.1	same as 4.1	10	5	3	150
5.4	choice out	open circuit	same as 5.1	same as 3	same as 3	same as 3	10	5	10	500

Top 3 problems (Pareto chart)	0	...	100	200	300	400	500
5.4 ECU signal 'choice out'							
3 switch sticky							
4.1 diesel does not start							

Please note: To introduce the basic FMEA principles, only selected components and failure modes where analyzed here. In a real world scenario more components (e.g. wires, connectors, …) and more failure modes (open circuit, short circuit, intermittent contacts, …) need to be analyzed.

Assessing criticality, probability and detectability

A scale of 1 lowest risk / probability, best detectability . . . 10 highest risk / probability, worst detectability

can be used. C, P and D should be assessed independently on a project specific basis, e.g.

Criticality *when available countermeasure has been taken*

1, 2	No effect on system function	user will not notice the failure
3, 4	Only unimportant functions are affected	only slightly annoying
5, 6	Systems works with limited function or performance	annoying, user will be unhappy
7, 8	System does not work any more (not safety critical)	user will be severely annoyed
9, 10	Safety critical	may damage life or goods

Probability *for automotive systems*

1, 2	Failure will never occur	Experience from similar systems	<10 ppm
3, 4	Small		<100 ppm
5, 6	Moderate		<1000 ppm
7, 8	High	In similar systems this failure showed up often	>10 000 ppm
9, 10	Failure will definitely occur		>100 000 ppm

Detectability

1, 2	automatically by the system's monitoring function, before the failure's effect shows up.
3, 4	automatically at the same time or shortly after the failure's effect shows up
5, 6	automatically, but long after the failure's effect shows up
7, 8	not automatically detectable, only detectable by human operator
9, 10	same as 7 or 8, but operator cannot take any countermeasures

8 Safety Integrity Levels and Functional Safety

How much safety do we need? → Accepted or unavoidable risk

Risk	Percentage of death persons per year = accepted (?) failure rate
Total mid of lifetime death risk (10…15 year old men)	10^{-3} / a
- death by all types of accidents	$0.5 \cdot 10^{-3}$ / a
- accidents at home	$0.4 \cdot 10^{-3}$ / a
- traffic accidents	$0.06 \cdot 10^{-3}$ / a
- natural disasters	$0.002 \cdot 10^{-3}$ / a

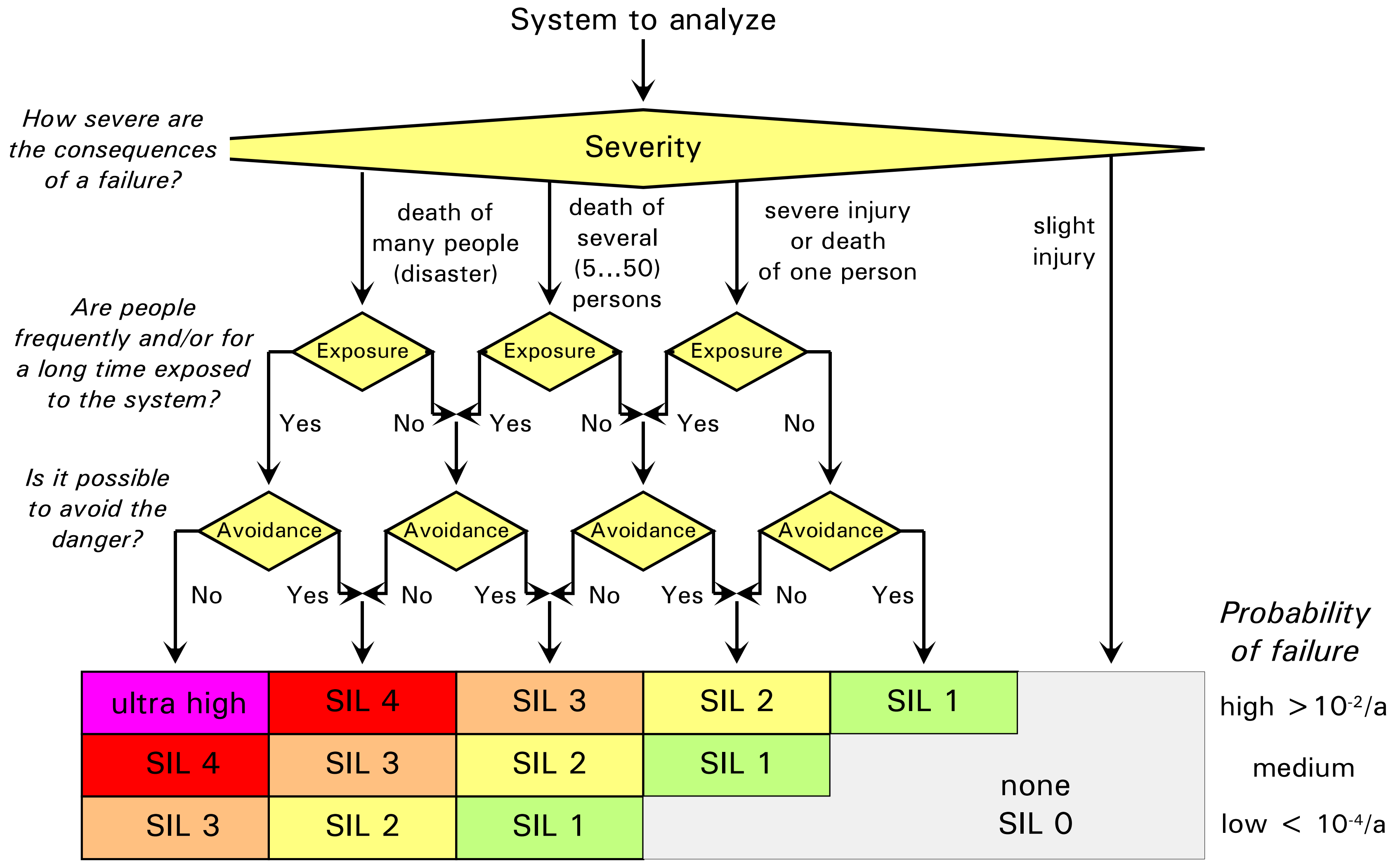
Source: David J. Smith & Kenneth G. L. Simpson, Functional Safety, 2nd Edition, Elsevier, 2004

Principles to define "acceptable risk"

- Technical systems should not considerable increase the human risk, i.e. $< 10^{-5}$ /a for general purpose technical systems. More risk may be accepted for individuals, who are free to decide, whether to expose to it or not, e.g. gliding with $2 \cdot 10^{-3}$ /a.
- A new technical system (e.g. electrical brakes) must not have a higher risk than an existing solution (e.g. mechanical/hydraulical brakes)
- Technical risks should be as low as reasonably possible (ALARP). Sad but true, assurance companies do calculate with "cost per life", typical 1 … 10 Mio $ per life (source: US Department of Transport report DOT HS 809 835, 2004).

IEC 61508 *Functional safety of electrical/electronic/programmable safety-related systems* (www.iec.ch/zone/fsafety) and its automotive specific version **ISO 26262** *Road Vehicles – Functional Safety* (under preparation, www.iso.org) define four **Safety Integrity Levels SIL** 4 to 1

IEC 61508 Risk Graph: Mapping risk to SIL



IEC 61508 and the pre-release version of ISO 26262 are not completely compliant with respect to SIL and other items.

IEC 61508 Required Reliability depends on SIL

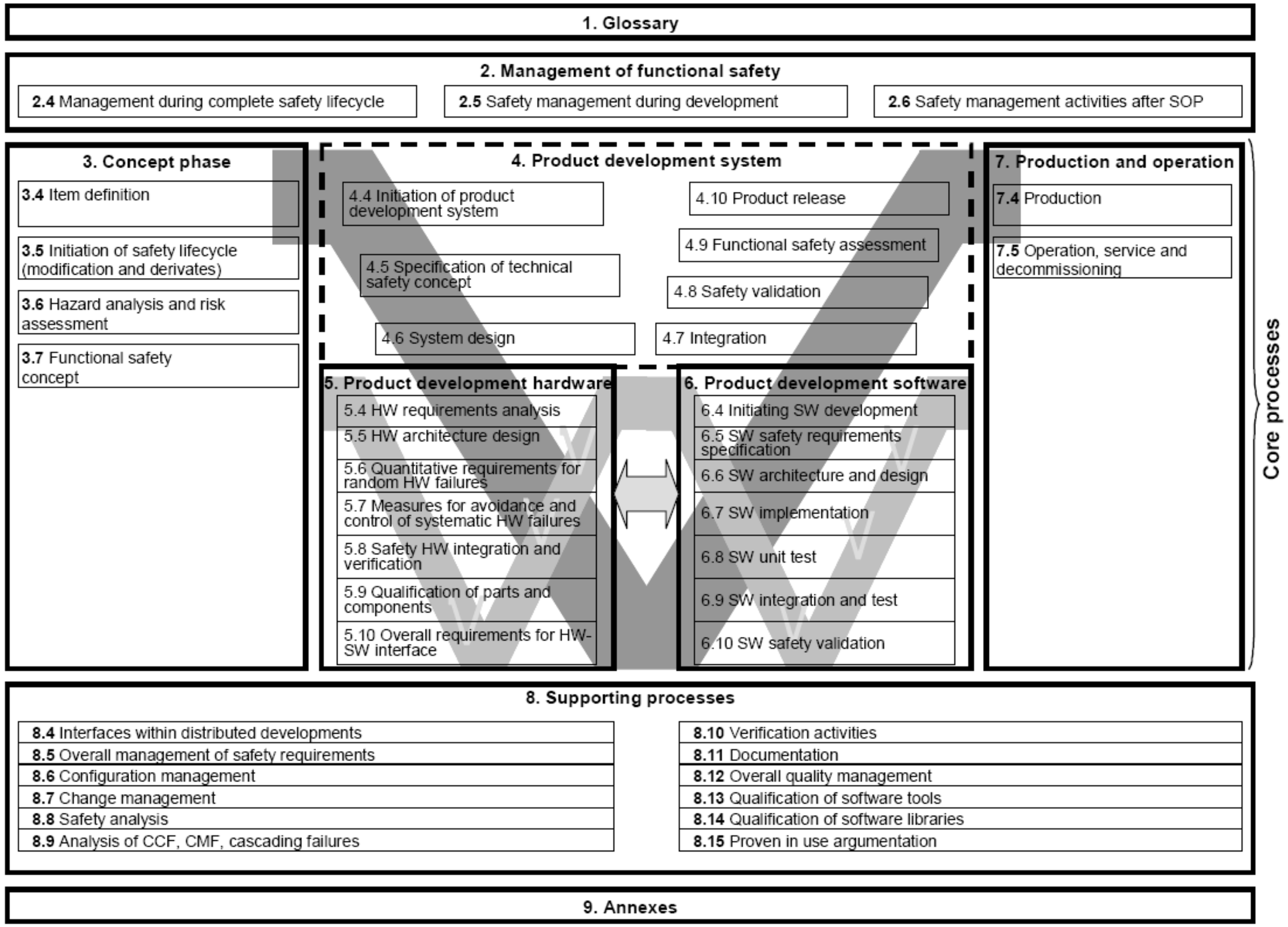
	Acceptable failure rate λ_D for dangerous failures (PFH Failures per Hour)	Acceptable failure probability F_D for dangerous failures (PFD Failure on Demand)
SIL 4 (Highest level)	$< 10^{-8}$ / h = 10 fit	$< 10^{-4}$
SIL 3	$< 10^{-7}$ / h = 100 fit	$< 10^{-3}$
SIL 2	$< 10^{-6}$ / h = 1000 fit	$< 10^{-2}$
SIL 1 (Lowest level)	$< 10^{-5}$ / h = 10000 fit	$< 10^{-1}$
Applies to	High demand rate systems, i.e. systems, which are frequently used, e.g. brakes of a car, so that faults will show up immediately as failures.	Low demand rate systems, i.e. systems, which are rarely used, e.g. airbags, so that faults can be dormant. In case that the system has a built in self test, the test period T is used when calculating F.

IEC 61508 Required Hardware Failure Tolerance HFT dependent on SIL

Safe Failure Fraction SFF $= (\lambda_S + \lambda_{DD}) / \lambda_{total}$	< 60%	60 … 90%	90 … 99%	> 99%
SIL 4 (Highest level)	Not allowed	Not allowed	≥ 2	≥ 1
SIL 3	Not allowed	≥ 2	≥ 1	Not required
SIL 2	≥ 2	≥ 1	Not required	Not required
SIL 1 (Lowest level)	≥ 1	Not required	Not required	Not required

For IEC 61508 type B systems. Type A systems without microcontrollers and software have more relaxed requirements.

IEC 61508 (and ISO 26262) define **requirements for the complete life cycle** of a product, e.g. a development process according to the V-model is required:



In many points IEC 61508 uses a very general approach and often gives recommendations, what to do, but not how to do it. Thus many additional or competing industry and/or company standards do exist, which go much more into detail, but are not fully compatible.

DO-178B, which is **used in** the **aerospace industry, and various SAE and VDA standards**, which are **used in** the **automotive industry**, use a classification like this:

Risk level	*Effect of failure*		
	Aerospace	*Automotive*	*Automotive Example*
A	Can't fly or land safely	Danger to life for many people	Loss of brakes in a bus
B	Major impact on ability to fly and/or land	Danger to life for few people	Unintended acceleration of a passenger car
C	Impact on ability to fly and/or land	Danger to goods, people may be hurt	Engine overspeed in a car
D	Minor impact on ability to fly and/or land	Danger to environment, Shutdown	Excessive pollution Engine stall
E	No impact on ability to fly and/or land	Reduced performance	Reduced vehicle speed
-		No effect on normal operation	Failure in error monitoring system

ISO 26262, the automotive version of IEC 61508, uses the following scheme:

- **Automotive Safety Integrity Levels ASIL D** (highest) **to ASIL A** (lowest)
- Required ASIL defined by **Severity** (classified as S0 … S3) and the combination **Exposure Time** x **Controllability**

		E (exposure time) * C (controllability)					
		1	0.1	0.01	0.001	0.0001	0.00001
Severity	S0 - no injuries	QM	QM	QM	QM	QM	QM
	S1 - slight and moderate injuries	ASIL B	ASIL A	QM	QM	QM	QM
	S2 - serious, including life-threatening, injuries, survival probable	ASIL C	ASIL B	ASIL A	QM	QM	QM
	S3 - life-threatening injuries (survival uncertain) or fatal injuries	ASIL D	ASIL C	ASIL B	ASIL A	QM	QM

QM … Non-safety related systems, only normal quality management required

E and C classification see below

Example: Steering system of a car

Severity: S3 Exposure time: E4 → E = 1 Controllability: C3 → C = 1

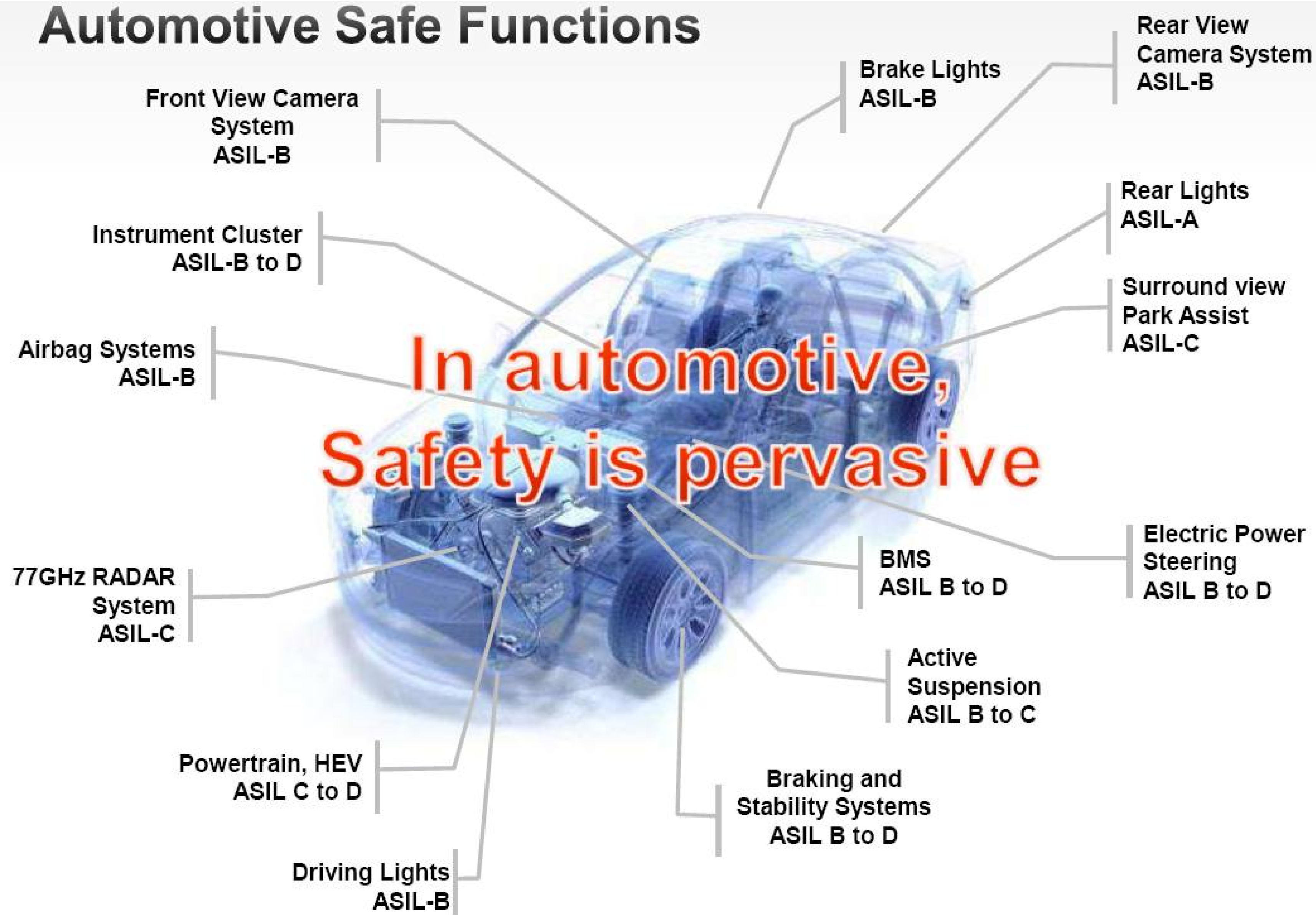
$\Rightarrow$ E · C = 1 + S3 $\Rightarrow$ ASIL D

ISO 26262 Exposure Time Classification

•Class	E1	E2	E3	E4
Description	Rare events	Sometimes	Quite often	Often
•Informative examples	Accident situation that requires release of the airbag Stop at railway crossing, which requires start of engine Towing, jump start.	Pulling a trailer, driving with roof rack Driving on a mountain pass with unsecured steep slope Driving situation with deviation from desired path	Fuelling, passing, tunnels, hill hold, car wash Night driving on roads without streetlights, wet roads, snow and ice, congestion	Starting, shifting gears, accelerating, braking, steering, using indicators, parking, driving backwards Driving on highways, driving on secondary roads, city driving
Value E	**0.001**	**0.01**	**0.1**	**1**

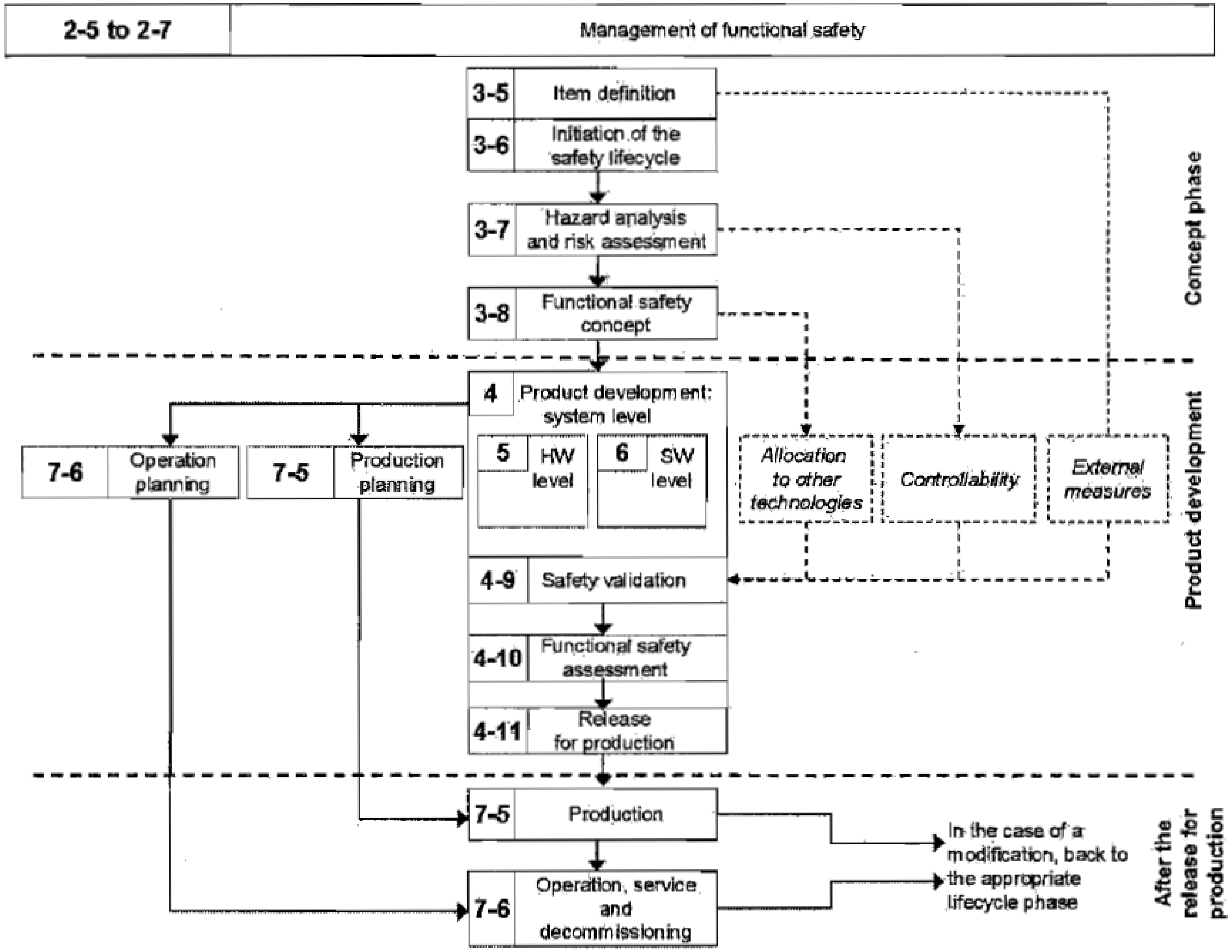
ISO 26262 Controllability Classification

Class	**C1**	**C2**	**C3**
Description	Simply controllable	Normally controllable	Difficult to control or uncontrollable
Definition	Less than 1% of average drivers or other traffic participants are usually unable to control the damage.	Less than 10% of average drivers or other traffic participants are usually unable to control the damage.	The average driver or other traffic participant is usually unable, or barely able, to control the damage.
Value C	**0.01**	**0.1**	**1**
Informative examples	When starting the vehicle with blocked steering column, the car can be brought to stop by almost all drivers early enough to avoid harm to persons nearby. Faulty adjustment of seats while driving can be controlled by almost all drivers through adjustment of seats and bringing the vehicle to a stop.	Driver can normally avoid departing from the lane: • in case of a failure of ABS during emergency braking. • on snow or ice in a curve in case of a failure of ABS during emergency braking. • in case of a motor failure at high lateral acceleration (motorway exit). Driver is normally able to bring the vehicle to a stop in case of a total light failure at medium or high speed on an unlighted country road without departing from the lane in an uncontrolled manner.	Self-steering with high angular speed at medium or high vehicle speed can hardly be controlled by the driver. Driver cannot bring the vehicle to a stop if a total loss of braking performance occurs. In case of faulty airbag release at high or moderate vehicle speed, driver usually cannot prevent vehicle from departing from the lane. Quelle: ISO/WD 26262-3



Source: Freescale

ISO 26262 Safety Lifecycle



Major Steps:

- Hazard Analysis and Risk Assessment

  Which hazards may occur and how risky are they?

- Functional Safety Concept

  Desig appropriate failure detection and handling functions for risky haz-ards?

- Safety Validation

  Validate by theoretical proofs and practical tests that the safety functions do manage risky hazards correctly.

- Functional Safety Assessment

  Formally assess your safety concept using FMEA, FTA etc.

Further Reading

- Books and Papers

[1] A. Birolini: Reliability Engineering. Springer, 6th Edition, 2010

[2] J. Börcsök: Electronic Safety Systems. Hüthig, 1st Edition, 2004

[3] J.W. Evans, J.Y. Evans: Product Integrity and Reliability in Design. Springer, 2012

[4] W. Denson, Mary Priore: Automotive Electronic Reliability Prediction, SAE Paper 870050, www.sae.org

[5] D. Hermann: Software Safety and Reliability. IEEE Computer Society, 3rd Edition, 2000

[6] N. Leveson: Safeware, System Safety and Computers. Addison-Wesley, 1995

[7] M.R. Lyu (editor): Handbook of Software Reliability Engineering. McGraw-Hill, 1998

[8] C. Jones: Software estimating rules of thumb, IEEE Computer 3/1996, page 116

[9] L. Rosenberg, T. Hammer: Software Metrics and Reliability. 1998, satc.gsfc.nasa.gov

[10] W. Vesely: NASA Fault Tree Handbook with Aerospace Applications, 2002, http://www.hq.nasa.gov

[11] D. Smith, K. Simpson: Safety Critical Systems Handbook. Elsevier, 3rd Edition, 2010

[12] J. Börcsök: Functional Safety. Hüthig, 1st Edition, 2007

[13] Exida: IEC 61508 Overview Report. 2006. www.exida.com

[14] D. Smith: Reliability, Maintainability and Risk. Elsevier, 8th edition, 2011

- Standards and Guidelines

[S1] MIL Handbook (HDBK) 217F Reliability Prediction of Electronic Equipment, US Department of Defense

[S2] MIL Handbook 338B Electronic Reliability Design Handbook

Further Reading

[S3] IEC 61709 Electronic Component Reliability. www.iec.ch

[S4] Bellcore TR-332: Reliability Prediction for Electronic Equipment

[S5] IEC 61709: Electronic Components Reliability. www.iec.ch

[S6] DO 178B Software Considerations in Airborne Systems and Equipment Certification. www.rtca.org

[S7] NASA Technical Standard 8719.13A: Software Safety. www.nasa.gov

[S8] MISRA Motor Industry Software Reliability Association, www.misra.org.uk:

Guidelines for the use of the C language in critical systems.

Guidelines for the use of the C++ language in critical systems.

Development guidelines for vehicle based software.

[S9] IEC 61508 Functional Safety of Electrical, Electronic and Programmable Safety-Related Systems. www.iec.ch/functionalsafety

[S10] ISO 26262 Road Vehciles – Functional Safety, www.iso.org

[S11] DIN V 19250: Grundlegende Sicherheitsbetrachtungen für MSR-Schutzeinrichtungen (withdrawn standard in favor of IEC 61508). www.din.de

[S12] SAE ARP 4754: Certification Considerations for Highly-Integrated Or Complex Aircraft Systems, www.sae.org

[S13] SAE ARP 4761: Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment www.sae.org

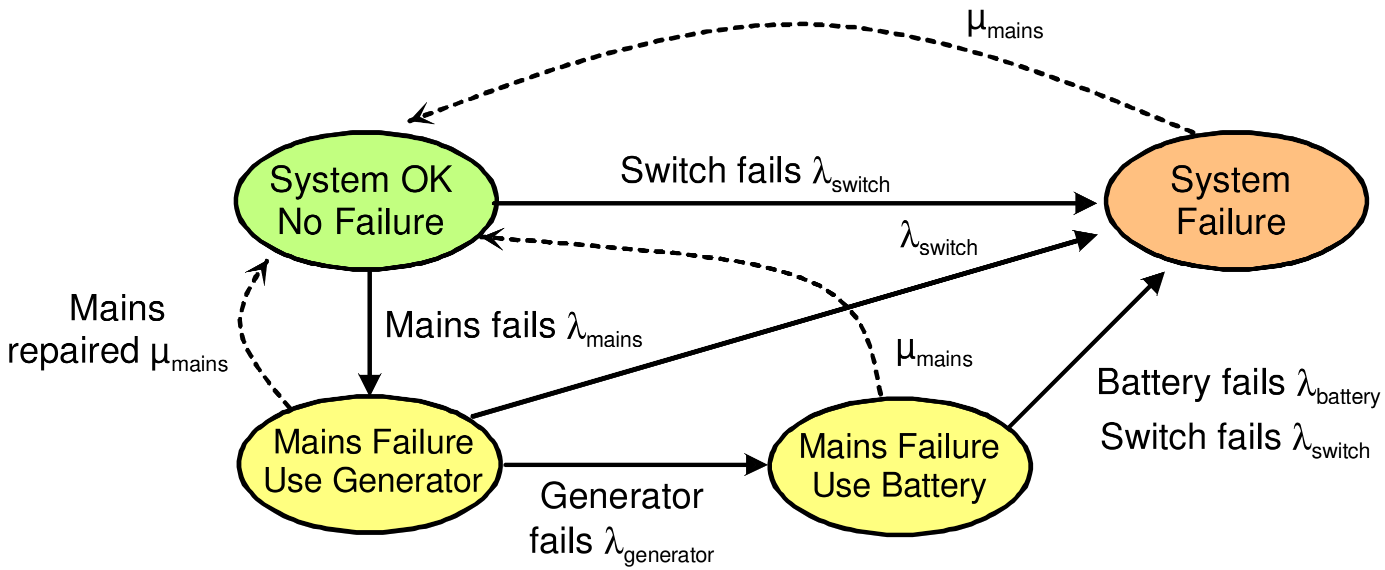
[S14] VDA FAKRA (Society of German Automobile Manufacturers): Automotive SPICE Prozessassessment. 2007, www.vda-qmc.de

Appendix: Methods not discussed in detail

Markov Models

- Similar to state charts, where working and failed components are described by states and failure events by state transitions. Failure rates λ describe the probability of the transitions.

  E.g.: Markov model for the electrical power supply of a hospital (incomplete)



- Can describe the dynamic behaviour of a system when failure events do occur in sequence, compared to FTA, which can only describe static behaviour.
- Allows to analyze systems, where all or some of the component failures are repaired, described by their respective repair rate μ (similar to failure rate λ)

MTTF for 1ooN redundant structures

$F = (F_{channel})^N \rightarrow R = 1 - (1 - R_{channel})^N = 1 - (1 - e^{-\lambda \cdot t})^N$

$$\rightarrow \quad MTTF = \int_0^\infty R(t)\,dt = \left\{ 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{N} \right\} \cdot \frac{1}{\lambda}$$

E.g.:

N=1: $MTTF = \frac{1}{\lambda}$

N=2: $MTTF = \frac{3}{2} \cdot \frac{1}{\lambda}$

50% improvement
100% cost increase

N=3: $MTTF = \frac{11}{6} \cdot \frac{1}{\lambda}$

83% improvement
200% cost increase

English – German Glossary (in German: Fachbegriffe)

Englisch	*Deutsch*	*Englisch*	*Deutsch*
Availability AV	Verfügbarkeit, Funktionsfähigkeit	Failure Rate λ (Probability of Failure per Hour PFH)	Ausfallrate, Fehlerrate
Bug	Softwarefehler	Fault	Störung, Fehler
Component	Bauteil	Fault Tolerant	Ausfalltolerantes System, das auch bei Ausfall/Störung einer Komponente funktionsfähig bleibt
Defect	Mangel, Defekt	Fault Tree	Fehlerbaum
Dependability	Verläßlichkeit, Überbegriff für Zuverlässigkeit	Functional Safety	Funktionale Sicherheit
Device	Gerät, Bauteil	Hardware Fault/Failure Tolerance HFT	Hardware-Fehlertoleranz
Dual Modular Redundancy DMR	Zwei-kanalige Redundanz	Harm	Schaden
Electronic Control Unit ECU	Steuergerät	Hazard	Gefährdung, gefährlicher Fehler
Error	Fehler im Sinn von Abweichung	Injury	Verletzung
Event	Ereignis	Line of Code LOC	Programmzeile
Fail Safe	System geht bei Ausfall einer oder mehrere Komponenten in einen sicheren Zustand	Maintenance	Wartung
Failure	Ausfall, Fehlverhalten	Mean Time Between Failure MTBF = MTTF + MTTR	Mittlere Zeit zwischen zwei Ausfällen (Betriebdauer plus Reparaturdauer)

Glossary

Englisch	*Deutsch*	*Englisch*	*Deutsch*
Failure Cause	Ausfallursache	Risk	Risiko = Kombination von Ausfallschwere und Ausfall-häufigkeit
Failure Duration	Ausfalldauer	Root Cause	Grundursache, auslösende Ursache
Failure Frequency	Ausfallhäufigkeit	Safe Failure Fraction SFF	Prozentualer Anteil der Ausfälle, die nicht sicherheitskritisch sind.
Failure Mode	Ausfallart	Safe State	Sicherer Zustand
Failure Occurrence	Auftretenszeitpunkt des Ausfalls	Safety	Sicherheit im Sinne von Gefahrlosigkeit
Failure Probability $F = 1 - R$ (Probability of Failure PF)	Ausfallwahrscheinlichkeit	Safety Critical	Sicherheitskritisch, gefährdend
Mean Time To Failure MTTF	Mittlere Betriebszeit bis zum Auftreten eines Ausfalls (ohne Reparaturdauer)	Safety Integrity Level	Sicherheitsstufe
Mean Time To Repair MTTR	Mittlere Reparaturdauer	Security	Sicherheit im Sinne von Zugriffs- und Datenschutz
Module	Baugruppe	Signal Range Check SRC	Signalbereichsüberprüfung
Monitoring	Überwachung	Stochastic	Zufällig
Notification	Benachrichtigung	Systematic	Systematisch
Overload, Overstress	Überlastung	Triple Modular Redundancy TMR	Drei-kanalige Redundanz
Plausibility	Plausibilität	Wear Out	Verschleiß
Redundancy	Redundanz		
Reliability $R = 1 - F$	Zuverlässigkeit, Überlebenswahrscheinlichkeit		