Test and Depentability 1: Threads and Countermeasures

Prof. G. Kemnitz

Institute for Informatics, TU Clausthal (TV_F1_engl.pdf)
May 5, 2023

Organisation

Lecture web page: http://techwww.in.tu-clausthal.de/TestVerl_2022

- Slide sets, handouts, homework, video recordings
- Submission of homework by e-mail to ha-tv@in.tu-clausthal.de as pdf. See web page for deadlines.
- Homework will be graded and returned. Additional publication of the number of points on the website.
- Examination admission 50% of the achievable homework points. For higher scores up to 2 bonus points for the exam.
- Questions and comments to: gkemnitz@in.tu-clausthal.de

Examination

- Written examination for 10 or more participants.
- Permitted aids for the written exam: Own elaboration incl. handouts with own comments and own homework, calculator.
- Permitted aids for the oral examination: an A4 sheet (one-sided) with your own elaborations.

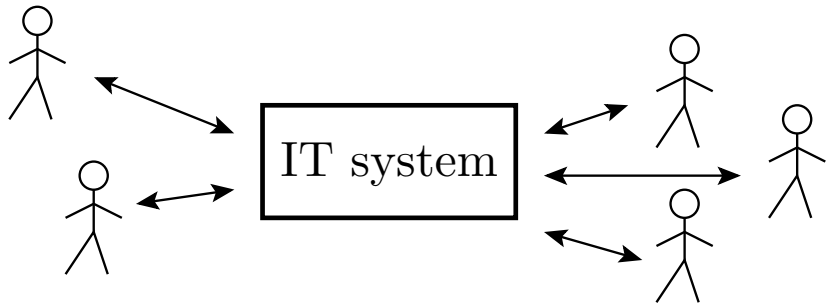
For all further information, see web page.

Contents slide set S1

lecture	1	2	3	4
slide	2	44	107	173

Introduction

Trust and dependability



IT systems automated intellectual tasks:

- operational procedures,
- control of processes and machines,
- design tasks, ...

The prerequisite for use is trust that

- the system works when it is needed,
- performs its services correctly and on time,
- and that it will not cause incalculable damage and costs.

Trust in an IT system presupposes dependability.

Dependability

In colloquial language, dependability (of people, computers, ...) describes the fact that they can be trusted. Different aspects come together (wishes, expectations, ...). Subjective factors influencing the perception of reliability:

- Life experiences, especially from childhood,
- catastrophes or slow changes,
- personality type (optimist, pessimist, conservative, gambler), ...

Objectification by counting positive and negative experiences and descriptive attributes:

- Reliable for what:
  - Lecturer reliable that on time,
  - student reliable, that homework are handed in, ...
- why reliable:
  - Doctor reliable because medical degree,
  - Car reliable because technically approved and technical supervision.

Apparently, monitoring and passed inspections are important for dependability, but also the error culture ...

Error culture

The way societies, cultures and social systems deal with mistakes and their consequences.

Negative view: Hiding mistakes, talking them away, ...

Positive view: Learning from mistakes, eliminating mistakes. ...

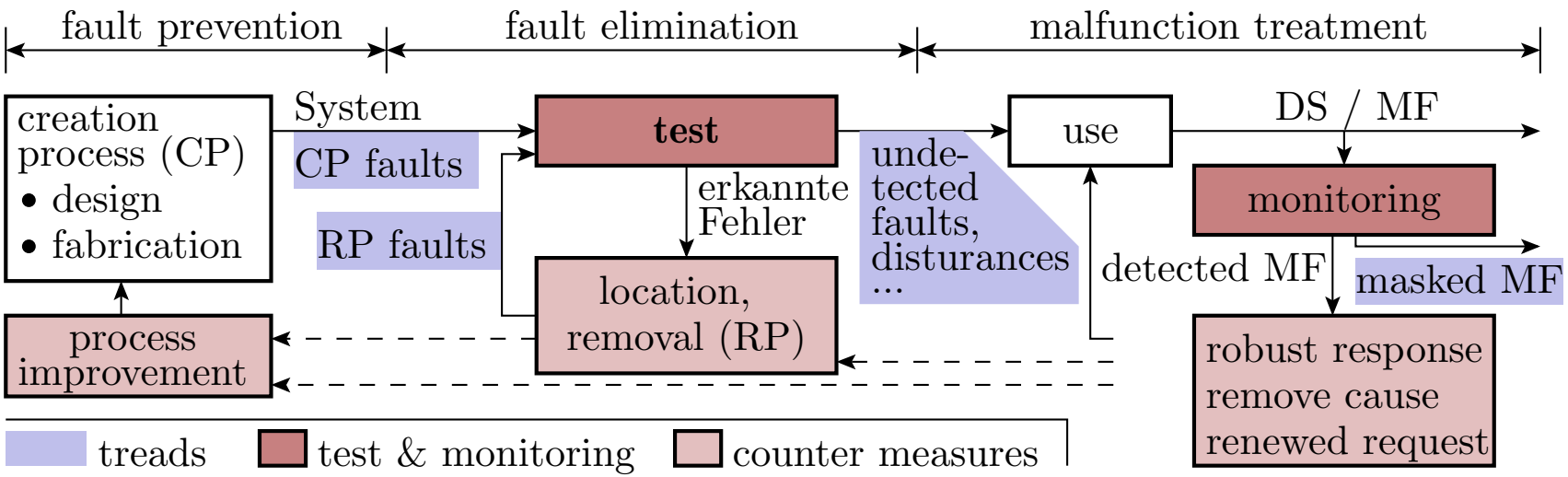
- Pedagogy: positive climate for learning from mistakes.
- Quality management: minimising the costs of mistakes.
- Innovation managers: striving for innovations. Error are seen as an opportunity / productive potential.

The lecture assumes the following idealised error culture:

- All detected problems are eliminated.
- Success of elimination is controlled by test repetition.

Unsuitable for cost optimisation in the IT sector. A less radical error culture is also desirable for dealing with friends, superiors and partners.

Hazards and hazard prevention



Threats to the dependability of IT systems:

- Malfunctions (MF) and their causes and consequences.
- Causes for MF are faults, Disturbances and failures.
- Causes for fault creation: problems in the creation process.

Countermeasures to avert hazards:

1. monitoring and malfunction treatment.
2. testing and fault elimination,
3. fault prevention by improving the processes of creation.

What is the cost of dependability?

The price of dependability is the total cost of all measures for hazard prevention at all three levels:

1. monitoring and response to detected MF: typically $> 50\%$ of total functionality, plus costs for repair, damage mitigation, ...
2. testing, fault elimination: For HW and SW typically more than 50% of total design effort.
3. fault prevention by improvement of the development processes: Costs for quality assurance and further development and improvement of the creation processes.

Dependability is the most expensive product characteristic even for IT systems without increased requirements. With higher demands on reliability the proportionate total costs for ensuring dependability amount to well over 50%.

The price of a lack of dependability

Dependability is expensive, why not waiver? - Damage costs:

- Data loss, backdoors for data misuse[1],
- accidents, self-destruction, production downtime, ...

> *On 3 June 1980, a computer at the North American Air Defence Centre reported the approach of Soviet nuclear missiles. Immediately, retaliatory measures were prepared. A check of the data from radar stations and satellites could not confirm the attack[2]. ...*

Cause of the near-nuclear strike: a defective circuit.

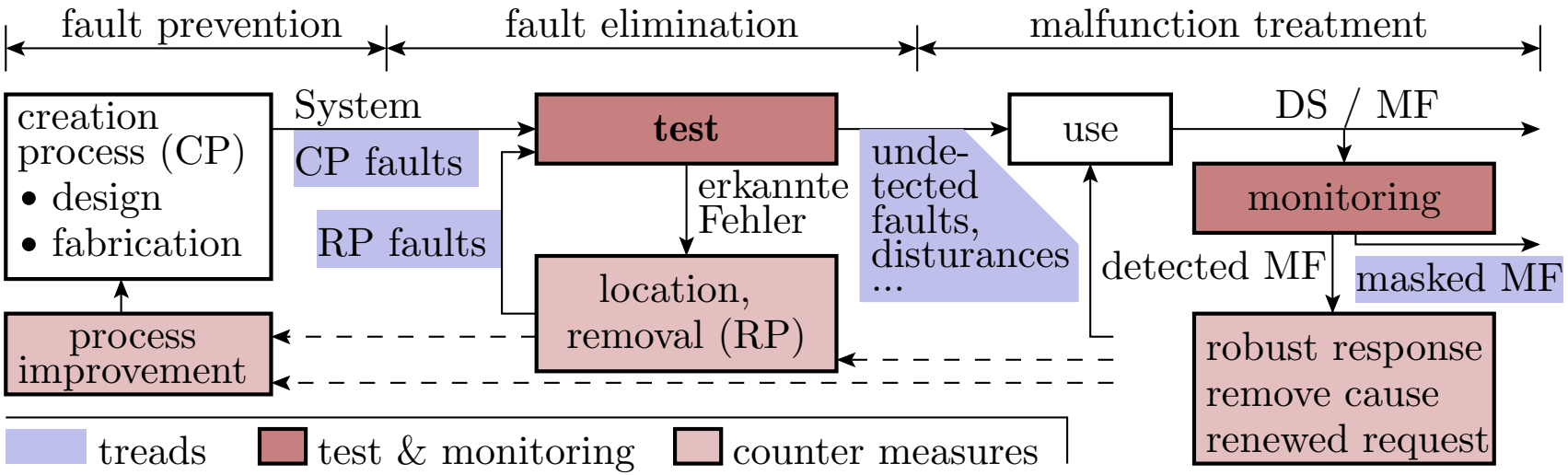
Unreliable IT systems are not usable.

---

[1] https://www.faz.net/aktuell/wirtschaft/diginomics/43-milliarden-euro-schaden-durch-hackerangriffe-15786660.html

[2] Hartmann, J., Analyse und Verbesserung der probabilistischen Testbar- keit kombinatorischer Schaltungen, Diss. Universität des Saarlandes, 1992

Why the lecture is named »Test and Dependability«



Reliability is ensured by iterations of controls, the elimination of identified hazards and success controls. With the assumed error culture "elimination of all identified hazards (MF, errors, ...)", the reliability of the systems in use depends mainly on the tests and controls at all three levels.

MF malfunction.

DS delivered service.

Learning objectives and content

Learning objects

Hazards and countermeasures to avert hazards. Focus on tests and monitoring, and how their effectiveness affects the dependability of IT systems in use.

Emergence and averting of hazards as well as the influence of unavoided hazards on dependability are stochastic in nature. For this purpose, a topic-specific introduction to stochastics is included.

Slide sets:

1. Threats and countermeasures: Introduction, overview, ...
2. Probabilities: error trees, Markov chains, ...
3. Distributions: counts, range estimation, ..
4. Monitoring: mixed signal, inspection, digital formats und values, ...
5. HW: fault modelling, test selection, self-test.
6. SW: fault avoidance, test selection.
7. Failures and failure tolerance.

Dependability

Description of dependability

The dependability of IT systems is described by the emergence and averting of threats on 3 levels:

	emerging hazards	Hazard aversion
Fault prevention	Faults from creation process	Reduction rate of occurrence
Test and fault elimination	Faults due to repair	Elimination of existing faults
monitoring under operation and MF treatment	MF, damage through MF	Damage prevention and correction of MF

A quantitative estimate of dependability aspects requires

- count values for created, avoided, ..., not recognised MF and
- the same for faults and their causes.

Therefore we have to describe IT systems in such a way that the correct services, the malfunctions, the faults etc. can be counted.

Service model

Service model

A service or service provider is a system that in response to

- SR: service requests

generates outputs from inputs. Classification of the service results:

- NS: not provided (no service),
- DS: delivered service:
  - CS: correct service result,
  - MF: malfunction.

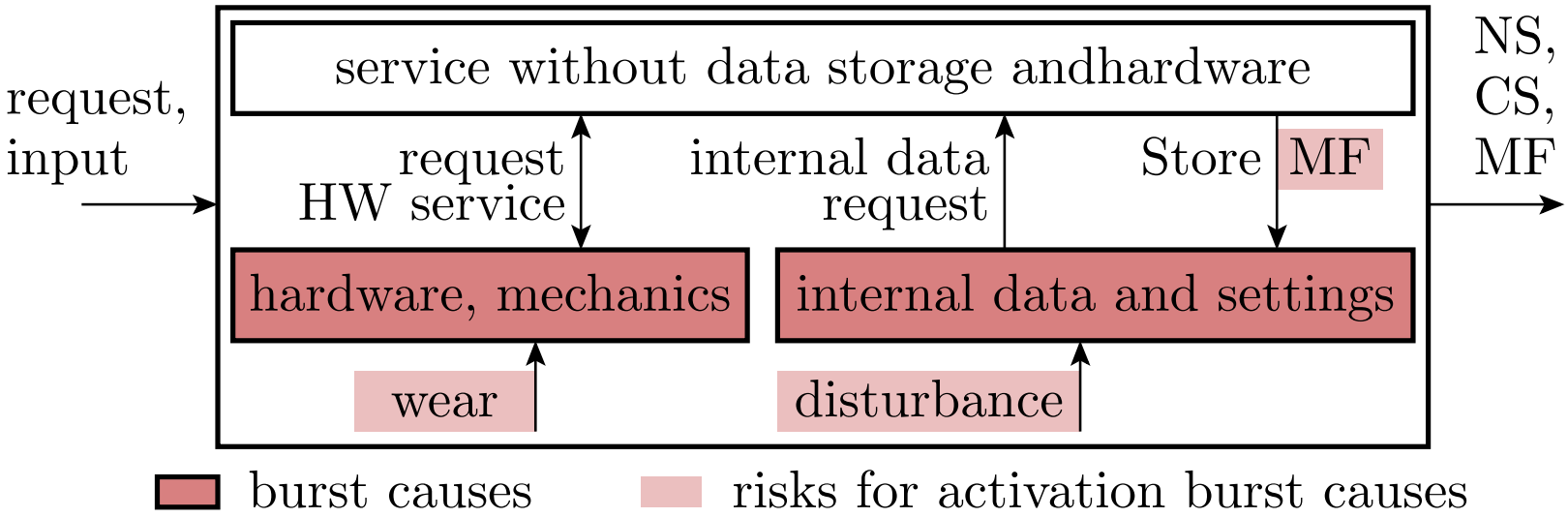
Estimable parameters for the dependability :

- Availability: proportion of time service delivers results on demand,
- Reliability*: expected number of DS per MF and
- Safety*: expected number of DS per HM.

HM **h**azzardous **m**alfunction.

* Usefull definitions, but still uncommon in the professional community.

Malfunction burst



HW failures and corruption of stored data respectively affect not only one, but all following service results (SR) and cause a burst of

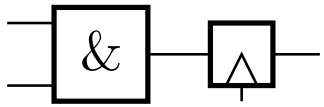
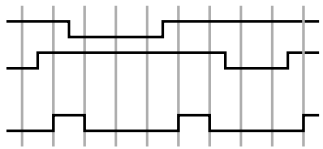
- no further service execution /result delivery,
- noticeably increased malfunction rate or
- not recognisably increased MF rate

until defective HW is repaired and distored data are corrected.

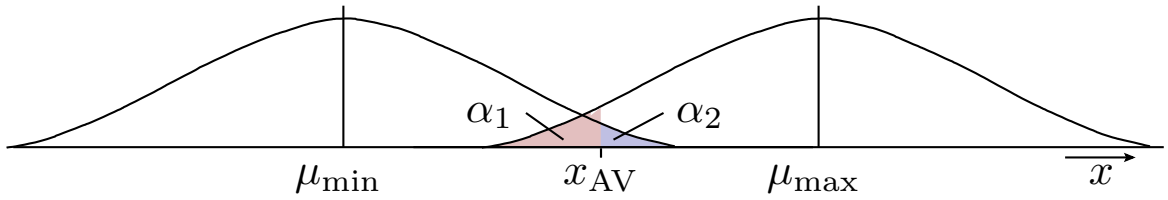
Detected MF bursts count as only one MF and DS and the system is considered unavailable untile cause is eliminated.

Areas of application of the service model

The service model is very universal and applicable to different levels of abstraction for IT systems, but also human services, technical controls, manufacturing processes, design processes, ...

clocked digital circuit		I:  O:
program with IPO-structure	`uint8_t up(uint8_t a){ return 23 * a; }`	I: 10 101 ... O: 320 19 ...
server	I: e.g. database request O: result data set	
manufacturing process	I: production order, material O: manufactured product	
design process	I: design order O: design result	

Minimum count values for parameter estimation



Many of the parameters introduced below are defined on the basis of expected values for count values:

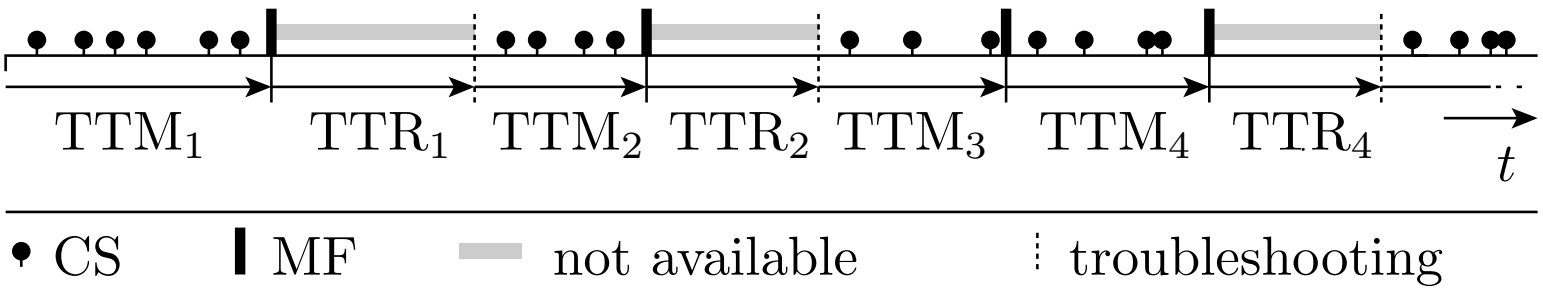
- malfunctions (MF): occurred, avoided, ...,
- Faults: existing, potential, modelled, detectable, avoided, ...

Experimentally determined actual values only allow range statements about the expected value that are subject to error probabilities and that are only meaningful if the actual values are within appropriate counting ranges (see sec. 3.2.7 *Range estimation count values*).

x_{AV}	actual value.
$\mu_{\min}$	minimum expectation value.
$\mu_{\max}$	maximum expectation value.
α_1, α_2	Probability of error, that the values are below or above the estimated range.
ACR	**a**ppropriate **c**ounting **r**anges.

Availability

Times between MF and until repair



- Mean time to next MF (**m**ean **t**ime **t**o **m**alfunction):

$$MTTM = \frac{1}{\#MF} \cdot \sum_{i=1}^{\#MF} TTM_i \Big|_{\text{ACR}}$$

- Mean repair time until replacement of failed hardware and/or reinitialisation (**m**ean **t**ime **t**o **r**epair):

$$MTTR = \frac{1}{\#MF} \cdot \sum_{i=1}^{\#MF} TTR_i \Big|_{\text{ACR}}$$

CS	correct service.
MF	malfunction.

Availability and PFD

Availability is the proportion of time during which the system is usable, i.e. neither failed nor crashed nor busy with a MF treatment:

$$A = \frac{MTTM}{MTTM + MTTR} = \left. \frac{\sum_{i=1}^{\#MF} TTF_i}{\sum_{i=1}^{\#MF} TTM_i + \sum_{i=1}^{\#MF} TTR_i} \right|_{\text{ACR}} \quad (1)$$

$$= 1 - PFD \quad (2)$$

A	**a**vailability.
$MTTM$	**m**ean **t**ime **t**o **m**alfunction.
$MTTR$	**m**ean **t**ime **t**o **r**epair.
$\#MF$	number of **m**al**f**unctions.
TTM_i	**t**ime **t**o **m**alfunction i.
TTR_i	**t**ime **t**o **r**repair after MF i.
ACR	**a**ppropriate **c**ounting **r**anges.
PFD	**p**robability of **f**ailure on **d**emand.

Repair times for highly available systems

Availability A	PFD	Sum of all repair times	
		per month	per year
99%	1%	7.2 h	87.6 h
99.9%	0.1%	43 min	8.8 h
99.99%	0.01%	4.3 min	53 min

$A \approx 99\%$ is common. High availabilities from 99.9% require special measures:

- uninterrupted power supply,
- RAID (**R**edundant **A**rray of **I**ndependent **D**isks),
- Mirrored servers, preventive maintenance, ...

(see sec. 4.3 *Failure tolerance*).

Rate of provided services

$$\eta_{\mathrm{DS}} = \left.\frac{\#DS}{\#SR}\right|_{\mathrm{ACR}}$$

Mean time to next MF:

$$MTTM = \frac{MTS}{1-\eta_{\mathrm{DS}}}$$

Used in Eq. 1.1:

$$A = \frac{MTTM}{MTTM+MTTR} = \frac{\frac{MTS}{1-\eta_{\mathrm{DS}}}}{\frac{MTS}{1-\eta_{\mathrm{DS}}}+MTTR} = \frac{1}{1-(1-\eta_{\mathrm{DS}})\cdot\frac{MTTR}{MTS}}$$

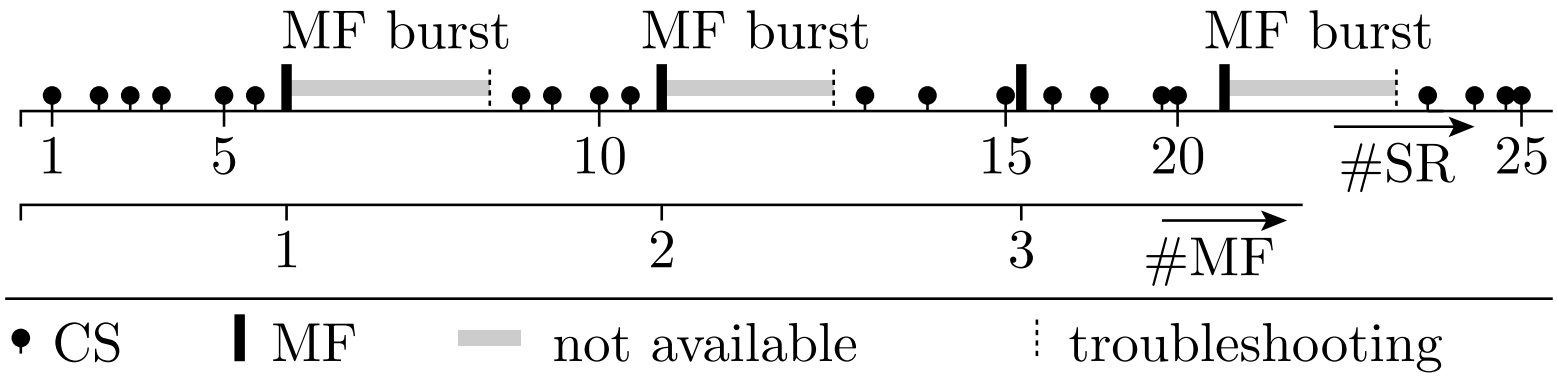
For the normal case A near 1, $x = (1-\eta_{\mathrm{DS}})\cdot\frac{MTTR}{MTS} \ll 1$:

$$A = \frac{1}{1-x} = \frac{1-x}{1-x^2} = 1-x = 1-(1-\eta_{\mathrm{DS}})\cdot\frac{MTTR}{MTS} \qquad (3)$$

η_{DS}	rate of **d**elivered **s**ervice results.
$\#DS$	number of **d**elivered **s**ervices.
$\#DS$	number of **d**elivered **s**ervices.
$MTTM$	**m**ean **t**ime **to** **m**alfunction.
MTS	**m**ean **t**ime to **s**ervice.

Reliability

Malfunction rate and Reliability



The malfunction rate is the relative number of MF per delivered services:

$$\zeta = \left.\frac{\#MF}{\#DS}\right|_{\text{ACR}} \tag{4}$$

and the ratio of mean time to service to mean time to next MF:

$$\zeta = \frac{MTS}{MTTM} \tag{5}$$

SR during a MF burst and other reasons for unavailability do not count, also to mitigate dependencies between counts (see sec. 3.2.7).

Reliability

Furthermore, let reliability be the expected number of services per MF or the reciprocal of the MF rate:

$$R = \left.\frac{\#DS}{\#MF}\right|_{\mathrm{ACR}} \tag{6}$$

$$= \frac{MTTM}{MTS} \tag{7}$$

$$= 1/\zeta \tag{8}$$

ζ	malfunction rate during operation.
R	**r**eliability.
$\#MF$	number of **m**al**f**unctions, MF bursts and failures count as one MF.
$\#DS$	number of **d**elivered **s**ervices.
ACR	**a**ppropriate **c**ounting **r**anges.
MTS	**m**ean **t**ime to **s**ervice.
$MTTM$	**m**ean **t**ime **t**o **m**alfunction.

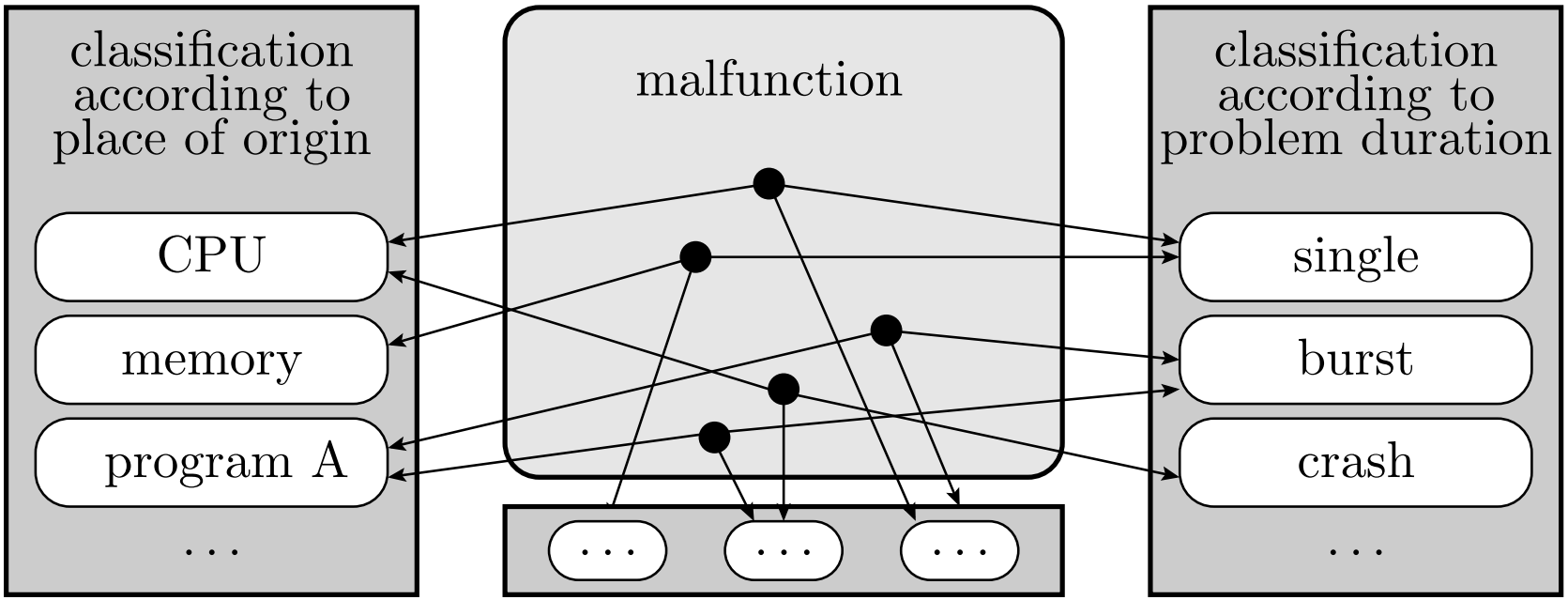
Example 1.1: Reliability and MF rate

Within 300 h of programme use 30 MF, $MTS = 0.1\,\mathrm{h}$. What is the reliability and the malfunction rate?

$$MTTM = \frac{300\,\mathrm{h}}{30\,[\mathrm{MF}]} = 10\,\mathrm{h}$$
$$R = \frac{10\,\mathrm{h}}{0.1\,\mathrm{h}} = 100\ \left[\frac{\mathrm{DS}}{\mathrm{MF}}\right]$$
$$\zeta = R^{-1} = 10^{-2}\ \left[\frac{\mathrm{MF}}{\mathrm{DS}}\right]$$

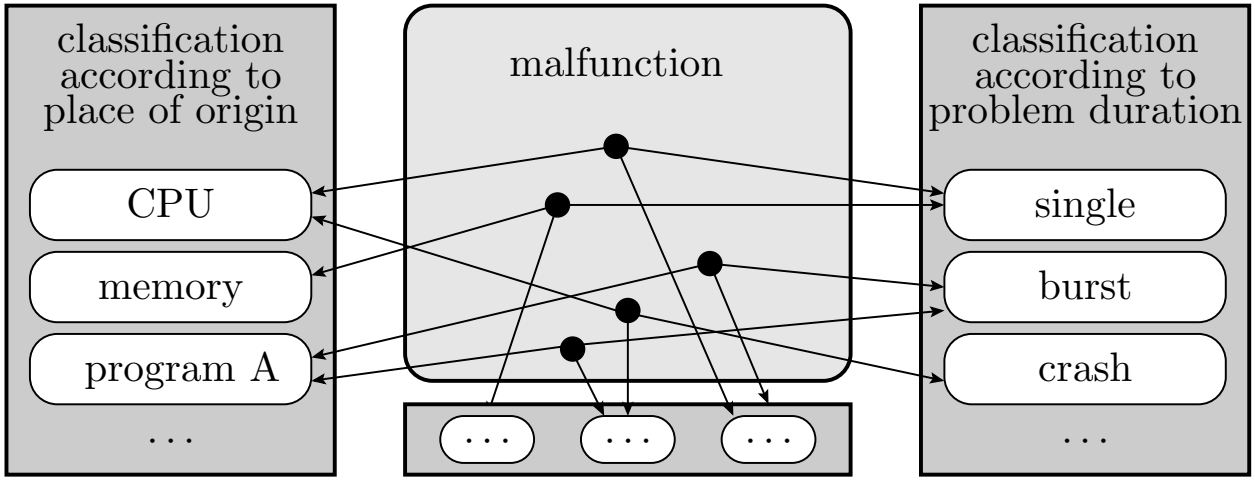
$MTTM$	**m**ean **t**ime **t**o **m**alfunction.
MTS	**m**ean **t**ime to **s**ervice.
ζ	malfunction rate during operation.
R	**r**eliability.
$\left[\frac{\mathrm{DS}}{\mathrm{MF}}\right]$	value in **d**elivered **s**ervices per **m**al**f**unction.
$\left[\frac{\mathrm{MF}}{\mathrm{DS}}\right]$	value in **m**al**f**unction per **d**elivered **s**ervices.

Partial reliabilities



The malfunctions (MF) of a system can be classified in different ways:

- according to location, cause, damage,
- MF of a particular subsystem only,
- MFs caused by HW, SW, ...
- only MFs affecting operational, data or access security.



With a unique assignment of each malfunction to exactly one class i, the total number of malfunctions $\#MF$ is the sum of the number of malfunctions $\#MF_i$ of all classes i:

$$\#MF = \sum_{i=1}^{\#MFC} \#MF_i$$

The malfunction rate as the relative frequency of MF per SR:

$$\zeta = \left.\frac{\#MF}{\#DS}\right|_{\text{ACR}} \tag{1.4}$$

$\#MF$	number of **m**al**f**unctions.
$\#MFC$	number of **m**al**f**unction **c**lasses.
$\#MF_i$	Number of MF of MF class i.

The malfunction rate is the sum of the malfunction rates of all malfunction classes

$$\zeta = \sum_{i=1}^{\#MFC} \zeta_i \tag{9}$$

and the reciprocal of the overall reliability is equal to the sum of the reciprocals of the partial reliabilities of all malfunction classes:

$$\frac{1}{R} = \sum_{i=1}^{\#MFC} \frac{1}{R_i} \tag{10}$$

ζ	malfunction rate during operation.
$\#MFC$	number of **m**al**f**unction **c**lasses.
ζ_i	MF rate of MF class i.
R	total **r**eliability.
R_i	partitial **r**eliability due to MF class i.

Example 1.2: Partial and total reliability

The malfunctions are either caused by the memory, the processor, the software or the rest. The subsystems have the following $MTTM$s:

subsystem i	memory	processor	software	all others
$MTTM_i$	500 h	3000 h	1000 h	2000 h

Average service duration (mean time to service) $MTS = 1\,\mathrm{min}$.

a) What are the four MF rates ζ_i and partial reliabilities R_i that can be derived from the $MTTM$ values?
b) What is the MF rate ζ and the reliability R of the overall system?

$MTTM$	**m**ean **t**ime **t**o **m**alfunction.
$MTTM_i$	**m**ean **t**ime **t**o **m**alfunction caused by MF of class i.
MTS	**m**ean **t**ime to **s**ervice.
R_i	partitial **r**eliability due to MF class i.
R	total **r**eliability.

subsystem i	memory	processor	software	all others
$MTTM_i$	500 h	3000 h	1000 h	2000 h

Average service duration (mean time to service) $MTS = 1\,\mathrm{min}$.

a) What are the four MF rates ζ_i and partial reliabilities R_i that can be derived from the $MTTM$ values?

subsystem i	memory	processor	software	all others
$MTTM_i$ in min	$3 \cdot 10^{-4}$	$18 \cdot 10^{-4}$	$6 \cdot 10^{-4}$	$12 \cdot 10^{-4}$
$R_i = \frac{MTTM_i}{MTS}$ in $\frac{\mathrm{DS}}{\mathrm{MF}}$	$3 \cdot 10^{-4}$	$18 \cdot 10^{-4}$	$6 \cdot 10^{-4}$	$12 \cdot 10^{-4}$
$\zeta_i = \frac{1}{R_i}$ in $\frac{\mathrm{MF}}{\mathrm{DS}}$	$3.33 \cdot 10^{-5}$	$5.56 \cdot 10^{-6}$	$1.67 \cdot 10^{-5}$	$8.33 \cdot 10^{-6}$

$MTTM_i$ **m**ean **t**ime **t**o **m**alfunction caused by MF of class i.
R_i partitial **r**eliability due to MF class i.
ζ_i MF rate of MF class i.
$[\frac{\mathrm{MF}}{\mathrm{DS}}]$ value in **m**al**f**unction per **d**elivered **s**ervices.

subsystem i	memory	processor	software	all others
$MTTM_i$	500 h	3000 h	1000 h	2000 h

Average service duration (mean time to service) $MTS = 1\,\mathrm{min}$.

b) What is the MF rate ζ and the reliability R of the overall system?

$$\begin{aligned}\zeta &= \left(3.33 \cdot 10^{-5} + 5.56 \cdot 10^{-6} + 1.67 \cdot 1 + 8.33 \cdot 10^{-6}\right) \left[\frac{\mathrm{MF}}{\mathrm{DS}}\right] \\ &= 6.39 \cdot 10^{-5} \left[\frac{\mathrm{MF}}{\mathrm{DS}}\right] \\ R &= \frac{1}{\zeta} = 1.57 \cdot 10^{4} \left[\frac{\mathrm{DS}}{\mathrm{MF}}\right]\end{aligned}$$

ζ	total malfunction rate.
R	total **r**eliability.
$\left[\frac{\mathrm{MF}}{\mathrm{DS}}\right]$	value in **m**al**f**unction per **d**elivered **s**ervices.
$\left[\frac{\mathrm{DS}}{\mathrm{MF}}\right]$	value in **d**elivered **s**ervices per **m**al**f**unction.

Safety

Damage due to malfunction

The potential damage caused by malfunctions ranges from insignificant to very large. The following safety levels are defined for industrial equipment according to IEC 61508 (SIL – **S**afety **I**ntegrity **L**evel) :

- SIL1: Minor damage to installations and property.
- SIL2: Major damage to installations, persons injured.
- SIL3: Injury to persons, some deaths.
- SIL4: Disasters, many deaths and serious environmental pollution.

The safety level specifies further parameters, e.g. upper limits for

- PFH (**p**robability of **f**ailure per **h**our) and
- PFD (**p**robability of **f**ailure on **d**emand):

SIL	1	2	3	4
$PFH_{\max}$	10^{-5}	10^{-6}	10^{-7}	10^{-8}
$PFD_{\max}$	10^{-1}	10^{-2}	10^{-3}	10^{-4}

We define safety alternatively as a partial reliability.

Safety-endangering malfunctions

Safety (Security) always refer to assumed hazards:

Security / safety	Safe from which hazards?
Operational safety	Personal and environmental damage
Data security	Theft of data
Security Data preservation	Loss of data
...	...

The rate of MF endangering the safety under consideration

$$\zeta_{\mathrm{S}} = \left.\frac{\#HM}{\#SR}\right|_{\mathrm{ACR}} \tag{11}$$

is by a factor η_{SE} smaller than the overall rate of all MF:

$$\zeta_{\mathrm{S}} = \zeta \cdot \eta_{\mathrm{SE}} \tag{12}$$

$$= \frac{MTS}{MTTH} \tag{13}$$

$\#HM$ number of **h**azzardous **m**alfunctions.
$MTTH$ **m**ean **t**ime **t**o **h**azzard.

Safety as partial reliability

A security against a certain MF class is the reciprocal of the MF rate through these MFs:

$$S = \left.\frac{\#SR}{\#HM}\right|_{\mathrm{ACR}} \tag{14}$$

$$S = 1/\zeta_{\mathrm{s}} \tag{15}$$

and thus the reliability divided by the proportion of hazardous FF:

$$S = \frac{R}{\eta_{\mathrm{SE}}} \tag{16}$$

More security: reliability$\Uparrow$ and/or $\eta_{\mathrm{SE}} \Downarrow$.

$\#HM$	number of **h**azzardous **m**alfunctions.
$\#DS$	number of **d**elivered **s**ervices.
ACR	**a**ppropriate **c**ounting **r**anges.
ζ_{S}	rate of **s**afety endangering MFs.
R	**r**eliability.
η_{SE}	percentage of **s**afety **e**ndagering MF.

Example 1.3: Safety through additional control device

A vehicle has a mean time between MF of $1000\,\mathrm{h}$. The proportion of MF endangering operational safety is 1% and the mean service time (mean journey time) is $1\,\mathrm{h}$. An additional electronic control device reduces the proportion of endangering MF to $\eta_{\mathrm{SE1}} = 10^{-3}$ HM per MF and has a reliability R_{CD}.

$MTTM = 1000\,\mathrm{h},\ \eta_{\mathrm{SE}} = 1\%,\ MTS = 1\,\mathrm{h},\ \eta_{\mathrm{SE1}} = 10^{-3}\ \left[\frac{\mathrm{HM}}{\mathrm{MF}}\right]$

a) What are the reliability R and the safety S of the system without the additional control device?

b) What must be the minimum reliability of the control device R_{CD} so that the additional control device increases the safety of the overall system at least fivefold ($S_{\mathrm{MT}} \geq 5 \cdot S$)?

$MTTM$	**m**ean **t**ime **t**o **m**alfunction.
η_{SE}	percentage of **s**afety **e**ndagering MF.
MTS	**m**ean **t**ime to **s**ervice.
R	**r**eliability without mulfunction treatment.
S	**s**afety without malfunction treatment.
R_{CD}	reliablility of the additional **c**ontrol **d**evice.
S_{MT}	**s**afety with **m**alfunction **t**reatment.

$MTTM$ =1000 h, $\eta_{\mathrm{SE}} = 1\%$, $MTS = 1\,\mathrm{h}$, $\eta_{\mathrm{SE1}} = 10^{-3}$ $\left[\frac{\mathrm{HM}}{\mathrm{MF}}\right]$

a) What are the reliability R and the safety S of the system without the additional control device?

Reliability:

$$R = \frac{MTTM}{MTS} \tag{1.7}$$

$$R = \frac{10^3\,\mathrm{h}}{1\,\mathrm{h}} = 10^3 \left[\frac{\mathrm{DS}}{\mathrm{MF}}\right]$$

Operational safety without additional control device:

$$S = \frac{R}{\eta_{\mathrm{SE}}} \tag{1.16}$$

$$S = 10^3 \left[\frac{\mathrm{DS}}{\mathrm{MF}}\right] / 1\% \left[\frac{\mathrm{HM}}{\mathrm{MF}}\right] = 10^5 \left[\frac{\mathrm{DS}}{\mathrm{HM}}\right]$$

R	**r**eliability without mulfunction treatment.
$MTTM$	**m**ean **t**ime **t**o **m**alfunction.
MTS	**m**ean **t**ime to **s**ervice.
η_{SE}	percentage of **s**afety **e**ndagering MF.
$\left[\frac{\mathrm{DS}}{\mathrm{MF}}\right]$	value in **d**elivered **s**ervices per **m**al**f**unction.

$MTTM$ =1000 h, $\eta_{\mathrm{SE}} = 1\%$, $MTS = 1$ h, $\eta_{\mathrm{SE1}} = 10^{-3}$ $\left[\frac{\mathrm{HM}}{\mathrm{MF}}\right]$

b) What must be the minimum reliability of the control device R_{CD} so that the additional control device increases the safety of the overall system at least fivefold ($S_{\mathrm{MT}} \geq 5 \cdot S$)?

$$S_{\mathrm{MT}} \geq 5 \cdot S = 5 \cdot \frac{R}{\eta_{\mathrm{SE}}}$$

$$S_{\mathrm{MT}} = \frac{1}{\eta_{\mathrm{SE1}}} \cdot \frac{1}{\frac{1}{R} + \frac{1}{R_{\mathrm{CD}}}} \geq 5 \cdot \frac{R}{\eta_{\mathrm{SE}}}$$

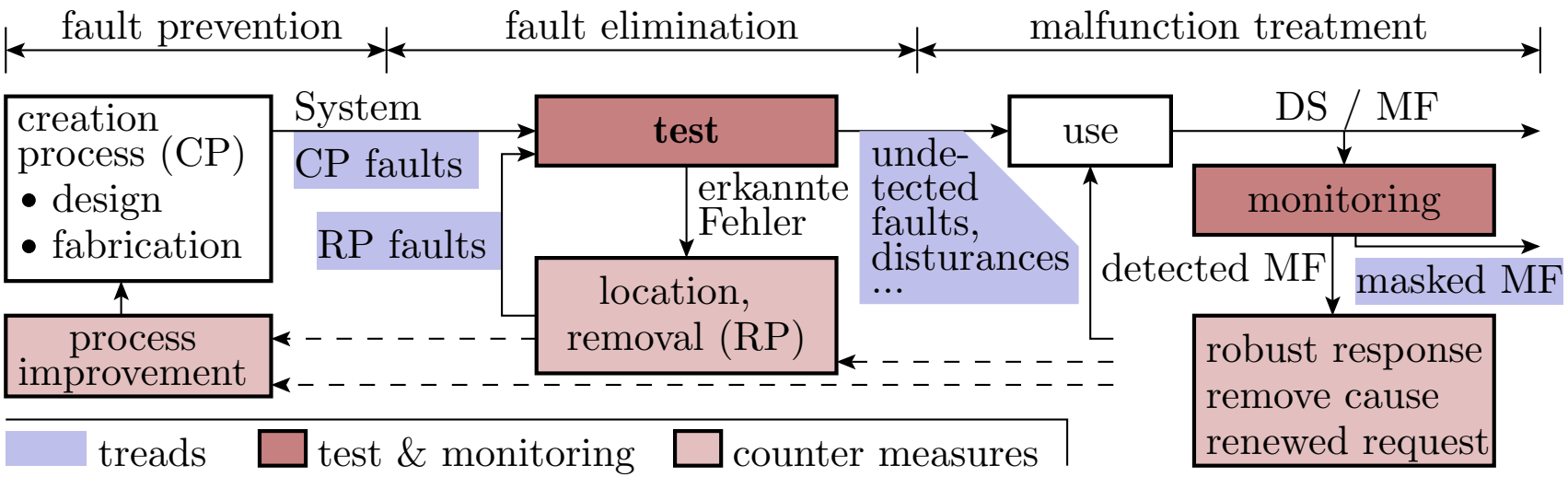
$$R_{\mathrm{CD}} \geq \frac{1}{\frac{\eta_{\mathrm{SE}}}{5 \cdot \eta_{\mathrm{SE1}} \cdot R} - \frac{1}{R}} = \frac{1}{\frac{2}{R} - \frac{1}{R}} = R$$

The additional control device must be at least as reliable as the vehicle itself.

As an alternative to todays ethical discussions on whether autonomous vehicles should run over children, pensioners, ... , a sufficient increase in safety compared to driver-controlled vehicles + third-party liability insurance should be demanded.

Summary

Ensuring dependability



Dependability is ensured on three levels:

- Damage prevention through monitoring and appropriate handling of detected MF caused by faults, disturbances and failures.
- Elimination of detected faults.
- Elimination or reduction of the causes of fault emergence.

Service model, availability

IT systems, HW, SW, creation processes, ... are regarded as service providers that generate outputs from inputs on demand:

- Service results (SR) and malfunction functions (MF) can thus be counted and times of availability, for repair ..., can be measured.
- Successive MF (MF bursts) usually have a common cause and are counted as one MF.

Availability is the relative frequency that the system is available when service is requested, describable as

- average proportional time of availability:

$$A = \frac{MTTM}{MTTM+MTTR} \tag{1.1}$$

- counter probability of unavailability:

$$A = 1 - PFD \tag{1.2}$$

- and as a function of the rate of service provided:

$$A = 1 - (1 - \eta_{\mathrm{DS}}) \cdot \frac{MTTR}{MTS} \tag{1.3}$$

Malfunction rate and reliability

Malfunction rate:

- relative frequency of malfunction per delivered service:

$$\zeta = \left.\frac{\#MF}{\#DS}\right|_{\text{ACR}} \tag{1.4}$$

- Ratio of mean service duration MTS to $MTTM$:

$$\zeta = \frac{MTS}{MTTM} \tag{1.5}$$

- sum of the MF rates of all MF classes:

$$\zeta = \sum_{i=1}^{\#MFC} \zeta_i \tag{1.9}$$

Reliability:

- expected number of delivered services per malfunction:

$$R = \left.\frac{\#DS}{\#MF}\right|_{\text{ACR}} \tag{1.6}$$

- reciprocal of the malfunction rate:

$$R = 1/\zeta \tag{1.8}$$

- reciprocal of the reciprocal sum of partial reliabilities:

$$\frac{1}{R} = \sum_{i=1}^{\#MFC} \frac{1}{R_i} \tag{1.10}$$

Hazardous malfunctions and safety

Rate of safety-endangering malfunctions:

- relative frequency of safety-endangering MF per DS:

$$\zeta_{\mathrm{S}} = \left.\frac{\#HM}{\#DS}\right|_{\mathrm{ACR}} \tag{1.11}$$

- proportion of the malfunction rate:

$$\zeta_{\mathrm{S}} = \zeta \cdot \eta_{\mathrm{SE}} \tag{1.12}$$

- Ratio of mean service duration MTS to $MTTH$:

$$\zeta_{\mathrm{S}} = \frac{MTS}{MTTH} \tag{1.13}$$

Safety:

- expected number of DSs per hazardous MF:

$$S = \left.\frac{\#DS}{\#HM}\right|_{\mathrm{ACR}} \tag{1.14}$$

- reciprocal of the rate of safety-endangering malfunction:

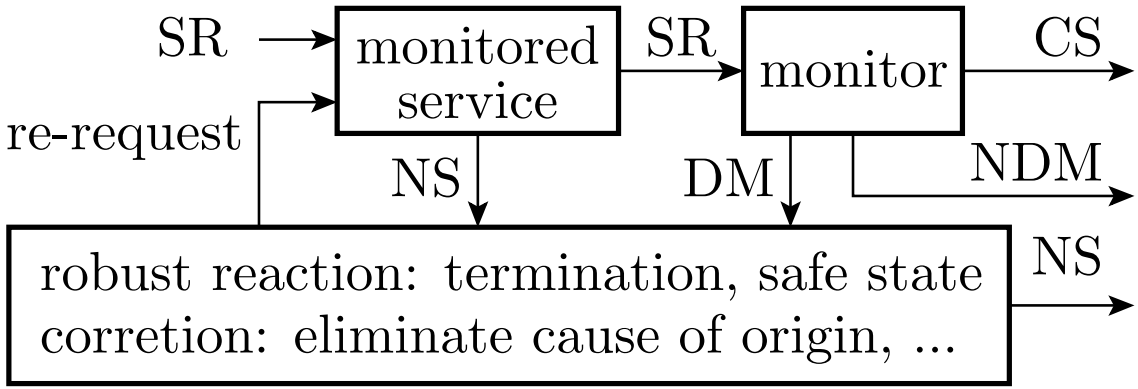
$$S = \frac{1}{\zeta_{\mathrm{S}}} \tag{1.15}$$

- reliability divided by the proportion of safety-endangering MF:

$$S = \frac{R}{\eta_{\mathrm{SE}}} \tag{1.16}$$

MF Treatment

MF treatment during operation

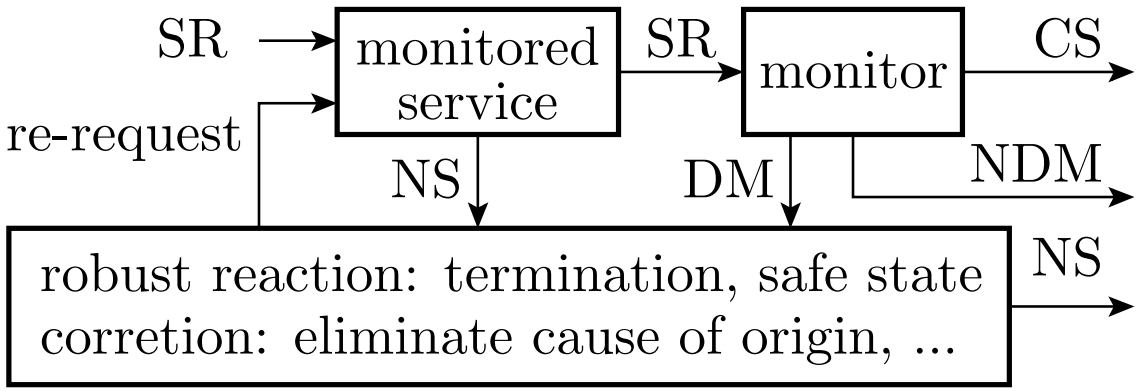


Monitoring the delivered service results (DS)

- detects only a part of the malfunction (MF) and
- possibly classifies correct services as MF.

SR	**s**ervice **r**equest.
DS	delivered service.
NS	no service.
DM	**d**etected **m**alfunction.
NDM	**n**ot **d**etected **m**alfunction.
CS	correct service.

3. MF Treatment



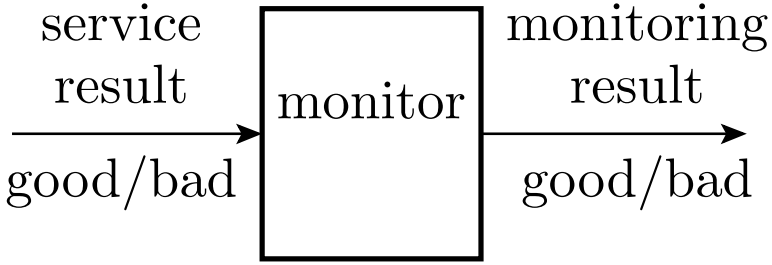
Reaction to detected MF:

- Robust response to detected malfunctions: Controlled behaviour to avoid damage and danger (do not use DS, if necessary, establish a safe state, ...).
- Logging of the observed misbehaviour for the manufacturer for troubleshooting.
- Restore functionality: repair / reconfiguration, reinitialisation.
- Correction of MF by repeated service request..
- If not correctable, termination without service (NS)..

SR	**s**ervice **r**equest.
DS	delivered service.
CS	correct service.

Monitoring

Parameters of monitoring



service result	monit. result	malfunction monitoring
bad	bad	—
good	good	—
bad	good	masking
gut	bad	phantom MF

1. Malfunctiion coverage, percentage of detectable MF:

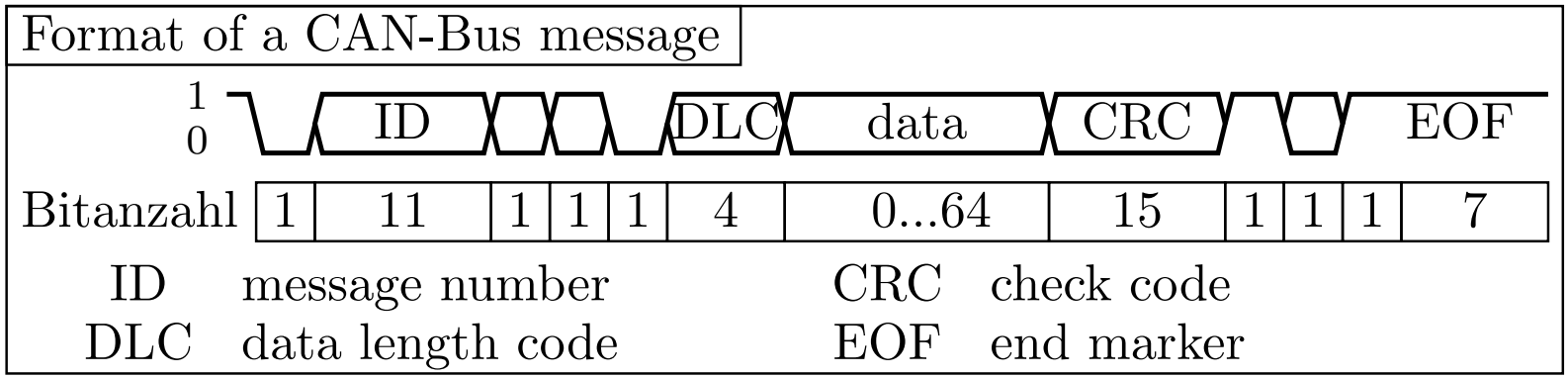
$$MC = \left.\frac{\#DM}{\#MF}\right|_{\mathrm{ACR}} \tag{17}$$

2. Phantom MF rate, percentage of correct SR classified as MF:

$$\zeta_{\mathrm{Phan}} = \left.\frac{\#PM}{\#DS}\right|_{\mathrm{ACR}} \tag{18}$$

$\#DM$	number of **d**etected **M**Fs.
$\#MF$	number of **m**al**f**unctions.
$\#PM$	number of **p**hantom **m**alfunction, i.e. DS classified as MF.
ACR	**a**ppropriate **c**ounting **r**anges.

Format- und Wertekontrollen



Services comprise:

- Format: value-independent characteristics: time limits, ranges, ...
- Data: Values of the data objects.

Classification of monitoring procedures for digital services:

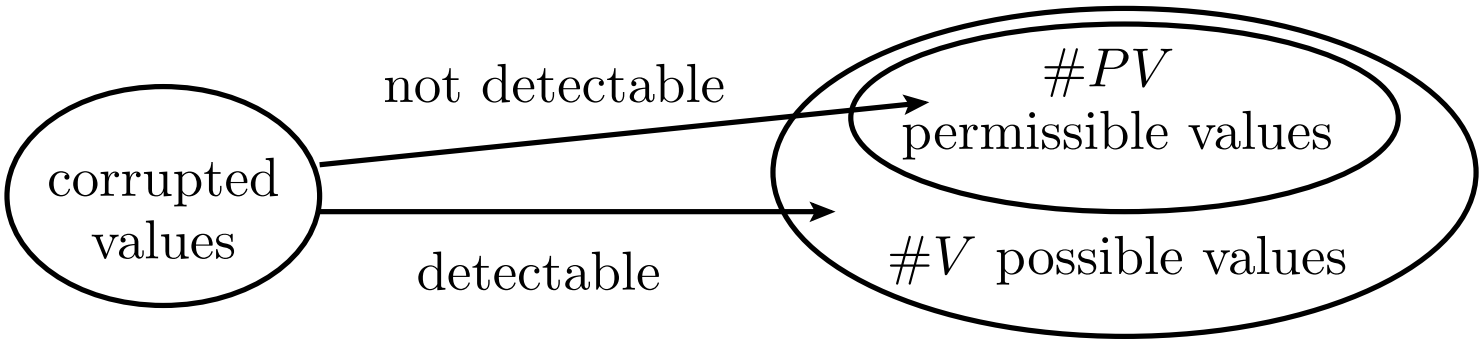
1. format checks: check of only value-independent characteristics. Services with format errors are always wrong and services with correct format may be wrong, i.e. only check for admissibility.
2. value checks: (additional) check of data values.

Format checks are easier to perform and often achieve higher MC and smaller phantom MF rates for digital services.

Format checks

Exploitation of data redundancy

Format checks (error-detecting codes, check sums, range monitoring, ...) mostly exploit information redundancy.



Malfunction coverage is the proportion of incorrect values mapped to prohibited values. If corruptions are mapped equally often to permissible and impermissible values and all impermissible values are detected

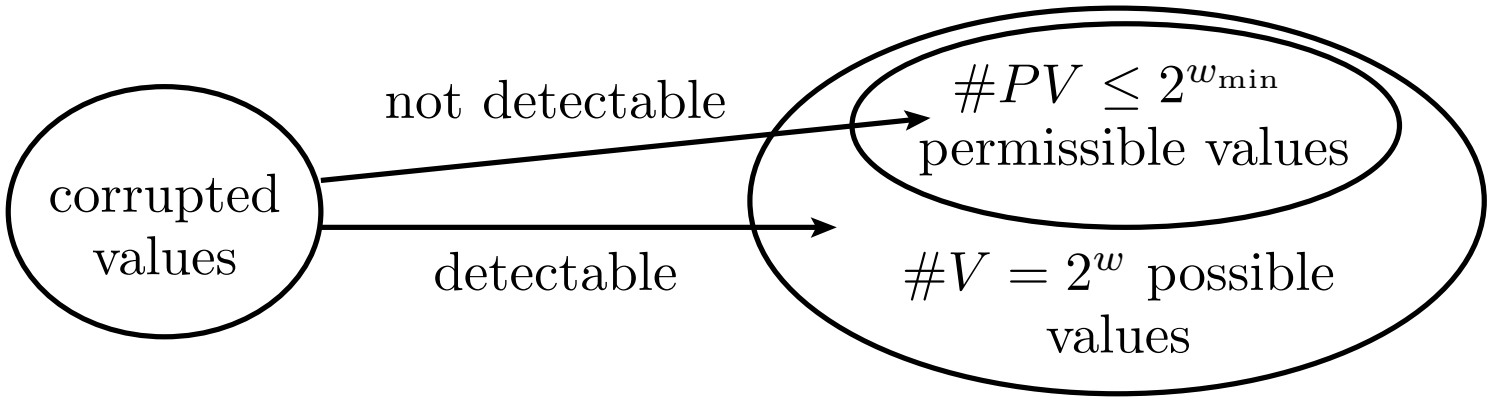
$$MC = 1 - \frac{\#PV}{\#V}$$

MC	**m**alfunction **c**overage, percentage of detected malfunctions.
$\#PV$	number of **p**ermitted **v**alues.
$\#V$	number of possible **v**alues.

Redundant bits

Assume that $w_{\min}$ bits are sufficient to distinguish all permissible values.
When represented with r additional (redundant) bits:

$$w = r + w_{\min}$$



$$MC = 1 - \frac{\#PV}{\#V} \geq 1 - \frac{2^{w_{\min}}}{2^{r+w_{\min}}} = 1 - 2^{-r}$$

MC	**m**alfunction **c**overage, percentage of detected malfunctions.
$\#PV$	number of **p**ermitted **v**alues.
$\#V$	number of possible **v**alues.
$w_{\min}$	**min**. number of data bits.
r	number of **r**edundant bits.

Check of a format with r redundant bits and

- uniform mapping of the distortions to possible values
- and proof of all impermissible values:

$$MC \geq 1 - 2^{-r} \tag{19}$$

$$\zeta_{\mathrm{Phan}} = 0$$

r	10	20	30
MC	$\approx 99.9\%$	$\approx 1 - 10^{-6}$	$\approx 1 - 10^{-9}$

Assuming $w_{\min} = 10^3$, no significant additional expenditure.

Ideal behaviour: fault-detecting codes and check sums.

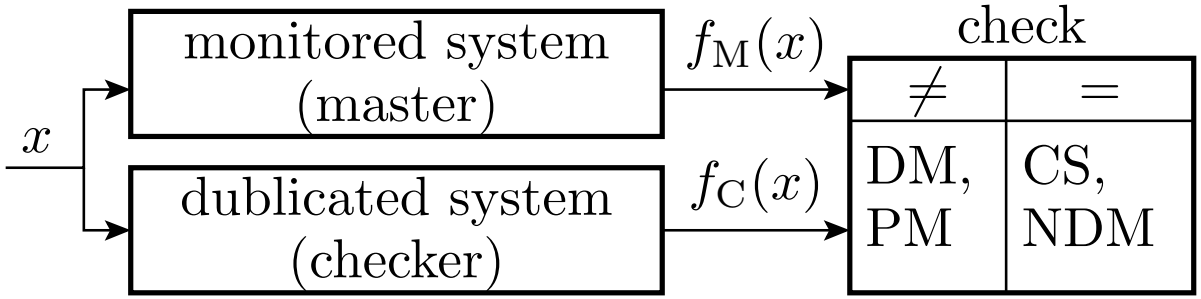
Format checks without uniform mapping of falsifications to permissible and impermissible values with nevertheless good effort-benefit ratio: checks of value ranges, data types, syntax, ... (see sec. 4.2.2 *Information redundancy*).

MC	**m**alfunction **c**overage, percentage of detected malfunctions.
r	number of **r**edundant bits.
ζ_{Phan}	**phan**tom MF rate.

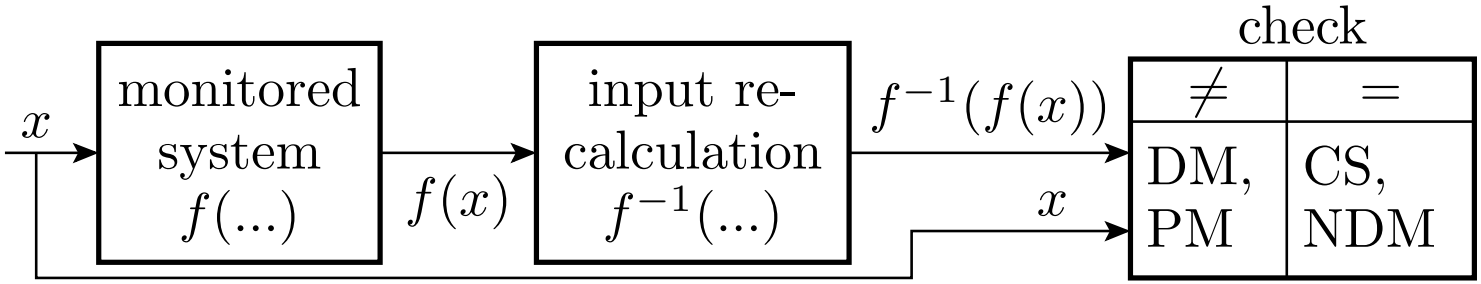
Value checks

Techniques for value checks

- Master checker principle, duplication and comparison:

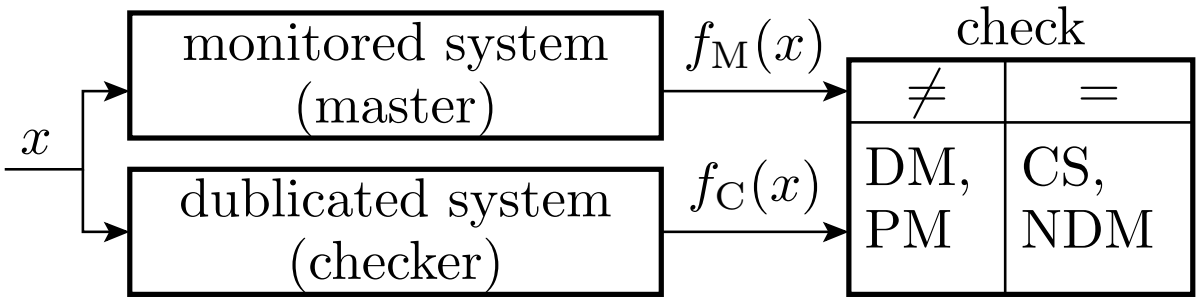


- Loop test, input recalculation and comparison, e.g. Monitoring sending by receiving and comparing the received data with the transmitted data:



- Data correctness test, e.g. for search path from A to B through a graph, check »path found leads from A to B«.

Properties of Master Checker systems



In the case of a DS with many bits and a low MF rate, practically all of the Master and Checker MFs without common cause are detectable:

$$MC = \eta_{\mathrm{Div}} \tag{20}$$

Checker-MFs with a deviating cause are phantom MFs:

$$\zeta_{\mathrm{Phan}} = \zeta_{\mathrm{Chk}} \cdot (1 - \eta_{\mathrm{Div}}) \tag{21}$$

CS	correct service.
DM	**d**etected **m**alfunction.
NDM	**n**ot **d**etected **m**alfunction.
PM	**p**hantom **m**alfunction, correct delivered service classified as malfunction.
η_{Div}	**div**ersity rate, percentage MFs without common cause.

Diversity

In technical terms, diversity is understood to mean different realisations of the same tasks to avoid the same MF in the case of multiple calculations or repetition. If

- MF occur very rarely and
- there are many possibilities for different FM

it is practically impossible that both calculations are disturbed simultaneously and falsified in the same way. In contrast, if the same fault comes into effect, it is almost certain that two identical calculations will be falsified in the same way. If the calculations are diverse, only a part of the MF due to faults will be equal:

$$\eta_{\mathrm{Div}} = 1 - \eta_{\mathrm{F}} \cdot \eta_{\mathrm{CF}} \tag{22}$$

η_{Div}	**div**ersity rate, percentage MFs without common cause.
η_{F}	percentage of malfunctions caused by **f**aults.
η_{CF}	percentage of malfunctions with a **c**ommon **f**ault as cause.

Avoidance of equal MF due to faults

The diversity and thus the MC of master checker systems be can be increased by constructive and organisational measures compared to that for two identical calculations with the same faults:

Extented diversity	Constructive or organisational action	additional detectable MF
HW diversity	execution on different HW	manufacturing defects, failures
HW design diversity	independently designed HW	HW design faults
Syntactic diversity	SW versions differently translated	SW translation faults
software diversity	independently designed SW	SW design faults
diverse use	Repetition* with changed SR	input error

* In case of deviating target values, unsuitable for duplication and comparison.

Diversity of software versions

Software faults as the main source of MF require diversity in the creation processes of the two versions and their faults:

- Complete development at least twice
- by separate teams, no communication,
- from a non-diverse specification, ...

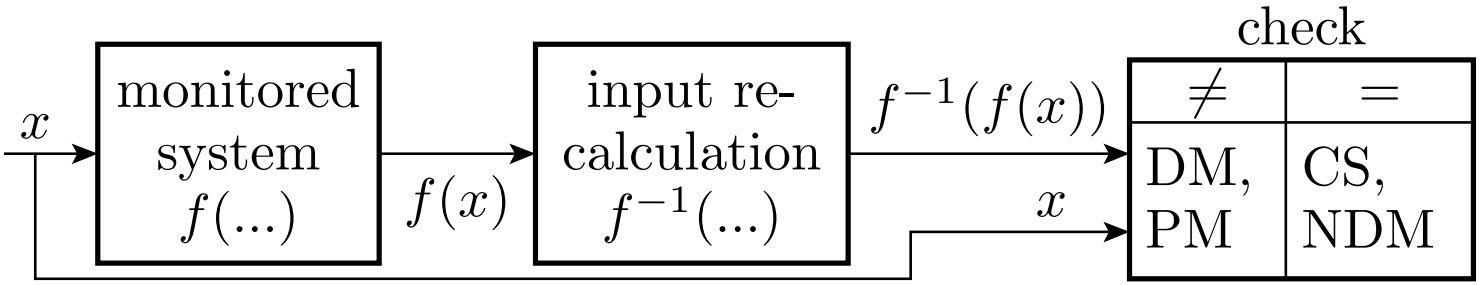
Original euphoric opinion that diversity towards all bugs except those in the specification could be achieved in this way, not confirmed. Direct or indirect communication between the development teams about the interpretation of the specification, during testing etc. carries commonalities into the designs. Tendency of people to repeat certain mistakes, ... Achievable diversity according to[3], ... $\eta_{\mathrm{CF}} \geq 10\%$, MF coverage doubling and comp. according to eq.1.22 und 1.24:

$$MC_{\mathrm{MC}} = 1 - \eta_{\mathrm{F}} \cdot \eta_{\mathrm{CF}} \leq 90\%$$

A check with $r = 10$ bit information redundancy achieves according to eq. 1.21 $MC \leq 99.9\%$ almost without additional effort and without PM.

[3] U. Voges, Software-Diversität und ihre Modellierung - Software-Fehlertoleranz und ihre Bewertung durch Fehler- und Kostenmodelle, Springer (1989)

Proporties loop test



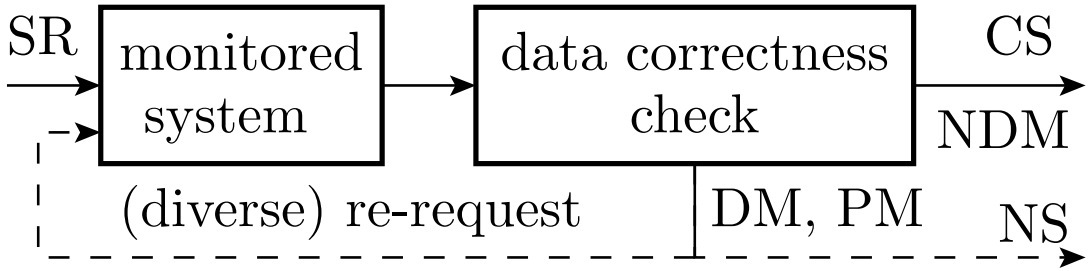
Since $f(\ldots)$ and $f^{-1}(\ldots)$ differ in algorithm and fault effect, a higher diversity than with doubling and comparison is to be expected and a MF coverage much higher the diversity rate:

$$MC \gg \eta_{\mathrm{Div}}$$

Only applicable if, $f(\ldots)$ is a reversibly unique mapping. Especially suitable if $f^{-1}(\ldots)$ is much simpler than $f(\ldots)$, e.g. root $\Leftrightarrow$ square.

$f^{-1}(\ldots)$	inverse function.
MC	**m**alfunction **c**overage, percentage of detected malfunctions.
η_{Div}	**div**ersity rate, percentage MFs without common cause.

Data correctness check

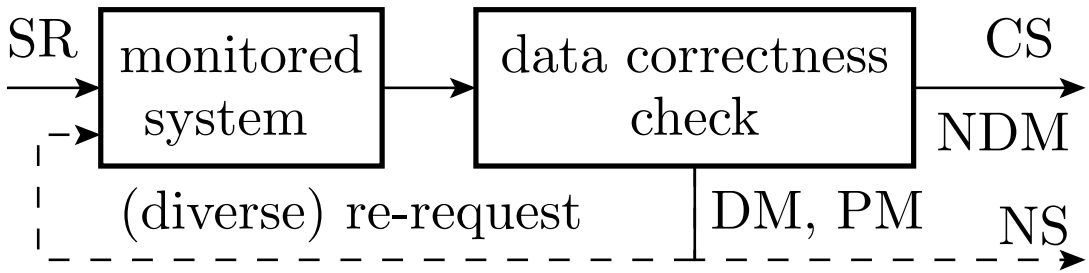


Example:

- Search path through a graph⇒ admissible path.
- Search test for error proof ⇒ fault simulation.
- Sort a list ⇒ list sorted and contains all elements.

Malfunction coverage MC is that of the monitoring algorithm, often very high, but for most target functions there is no such monitoring option.

SR	**s**ervice **r**equest.
DS	delivered service.
DM	**d**etected **m**alfunction.
CS	correct service.
NDM	**n**ot **d**etected **m**alfunction.



Notice, for the typical form of search algorithms

```
Retry until control passed
    Guess the result
```

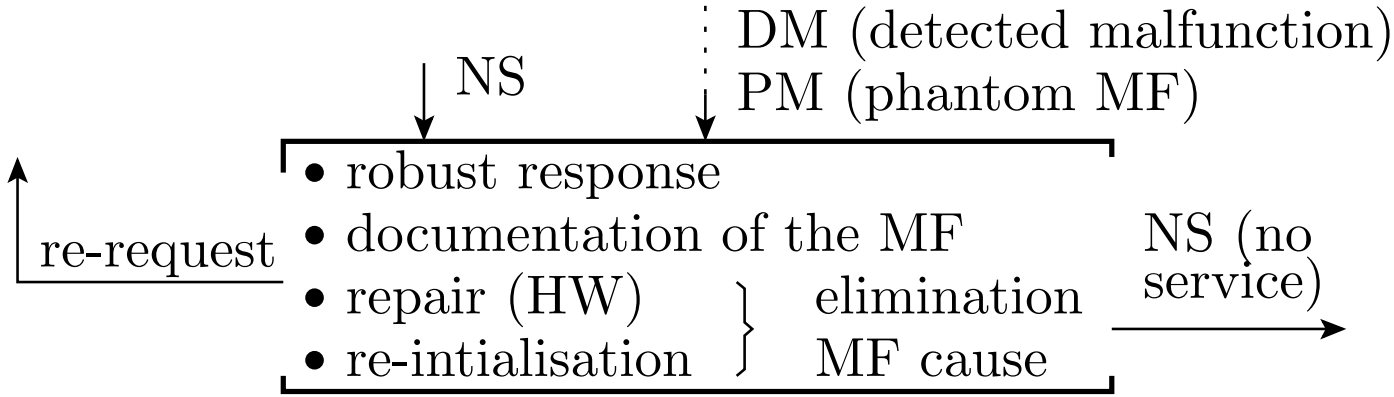
the MF rate ζ with no malfunction treatment strives towards 1 and with sorting out of detected MFs:

$$\zeta_{\mathrm{MT}} = \lim_{\zeta \to 1} (\zeta \cdot (1 - MC)) = 1 - MC$$

CS	correct service.
NDM	**n**ot **d**etected **m**alfunction.
PM	**p**hantom **m**alfunction, correct delivered service classified as malfunction.
ζ_{MT}	MF rate after **m**ulfunction **t**reatment.
ζ	malfunction rate without mulfunction treatment.
MC	**m**alfunction **c**overage, percentage of detected malfunctions.

Handling of detected MF

Handling of detected MF



Robust response to detected MF incl. damage prevention:

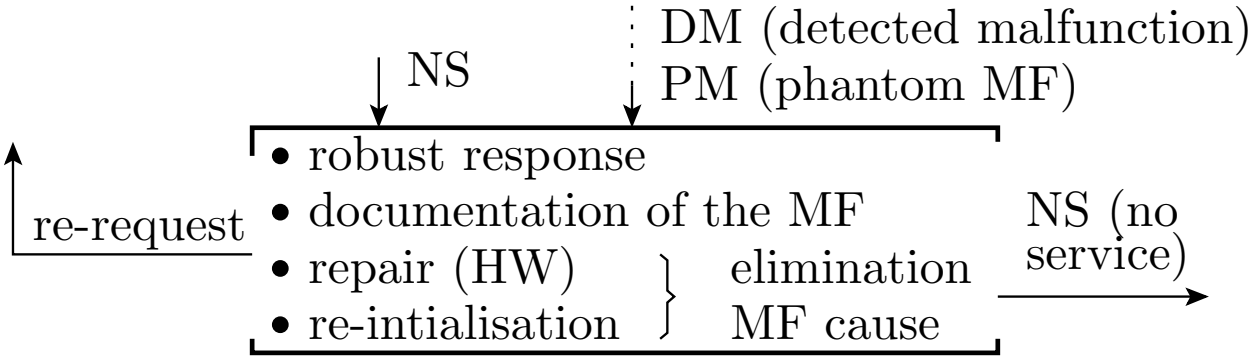
- cancel processing, save data, ...
- transition to a safe state, e.g. emergency stop.

Dokumentation of MF:

- error message, core dump, cap file (Windows), ... (see sec. 1.4.6 *Maturing process*).

Elimination of the cause of the MF:

- Repair of failed HW or
- Reconfiguration with/without reduced performance, ...
- Restoration of a / the last permissible system state.



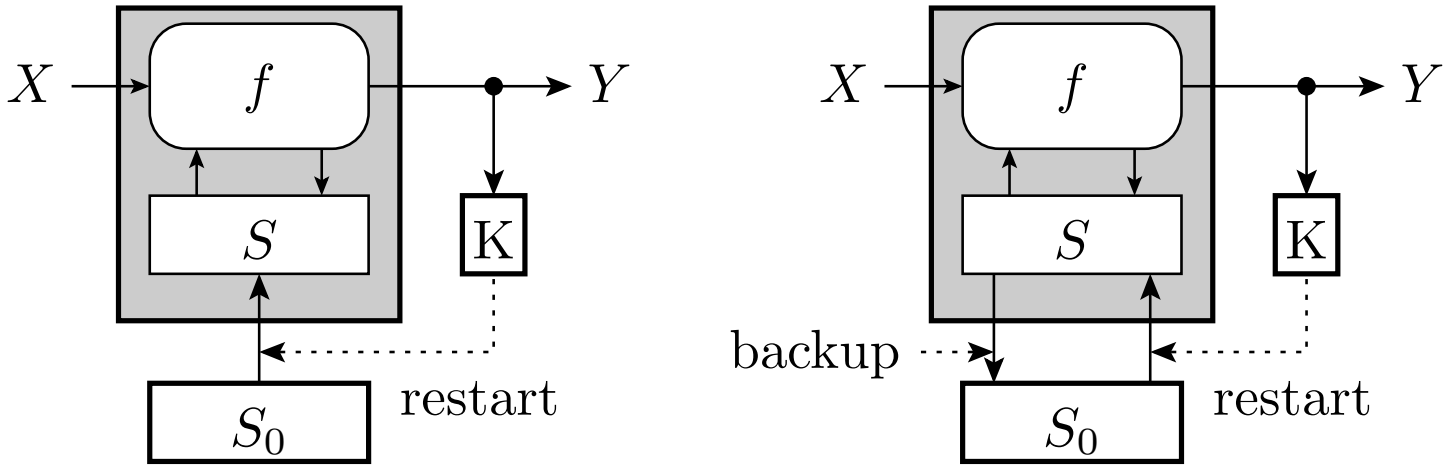
The same MF after a new request indicates the same cause of occurrence, usually a fault or failure. Options for further action:

- only robust reaction without correction,
- else (assumed fault):
  - Change request to the manufacturer (see sec. 1.4.6 *Maturing process*)
  - Looking for ways to bypass faults.

Correction without (diverse) reserve units requires complex manual interaction. Unsuitable for automatic MF treatment.

SR	**s**ervice **r**equest.
DM	**d**etected **m**alfunction.
PM	**p**hantom **m**alfunction, correct delivered service classified as malfunction.
NS	no service.

Reinitialisation



With MF, internal data are often falsified. To return to a functional state, the internal data must be reinitialised with permissible values:

- Static reinitialisation (reset): fixed initial state,
- Dynamic reinitialisation: Regular backup during operation. Loading the last backup after a crash.

Redo calculation from last backup to failure. Often only data are backed up that cannot be recalculated easily, e.g. in the case of editors, logistics systems, databases, ... the inputs since the last complete backup.

Avoidance of problematic MF

Exclusion of MF for which a robust response is difficult, by

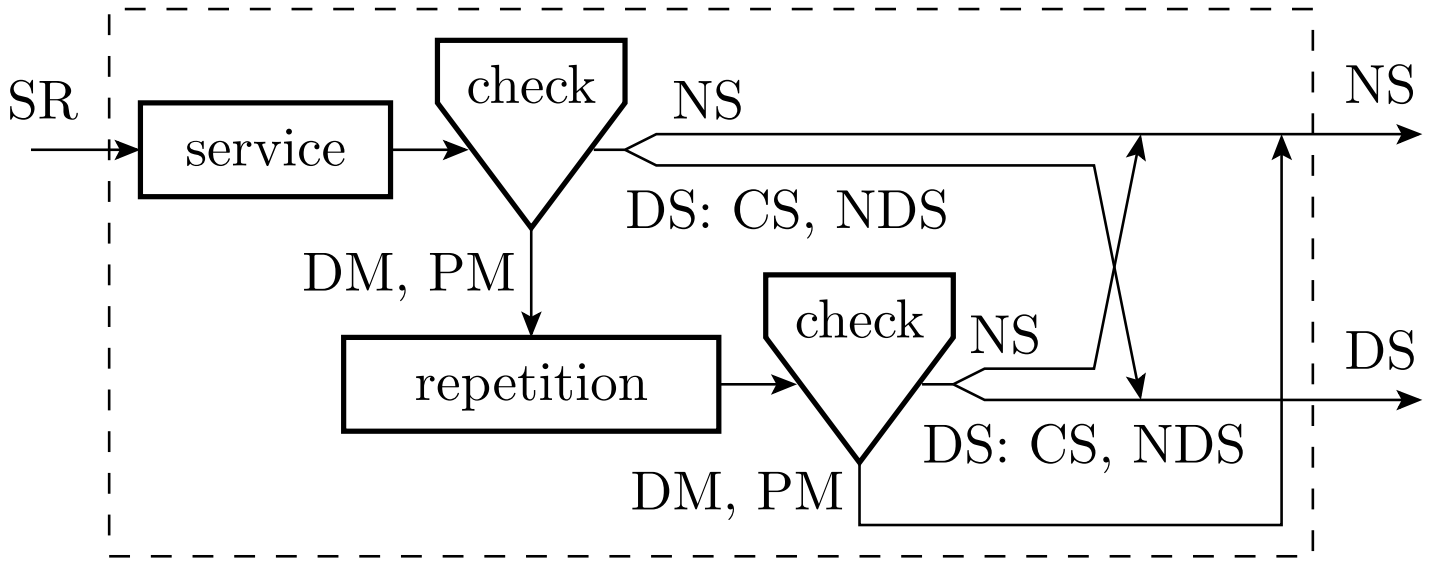
- system design, application guidelines,
- health and safety regulations,
- certification processes, ...

Design principles for safety critical systems:

- Closed-circuit principle: design principle in which the system automatically switches to a safe state in the event of failure:
  - Railway signalling: fault indication in case of missing quiescent current.
  - Fire alarm system: alarm in case of wire breakage.
  - Vehicle brake: Braking when brake hose bursts, ...
- MF isolation: exclusion of MF propagation between functionally independent components, e.g. separate processes executed on the same computer.
- Fire walls: Exclusion of MF propagation via subsystem interfaces that require special protection, also against cyber attacks.

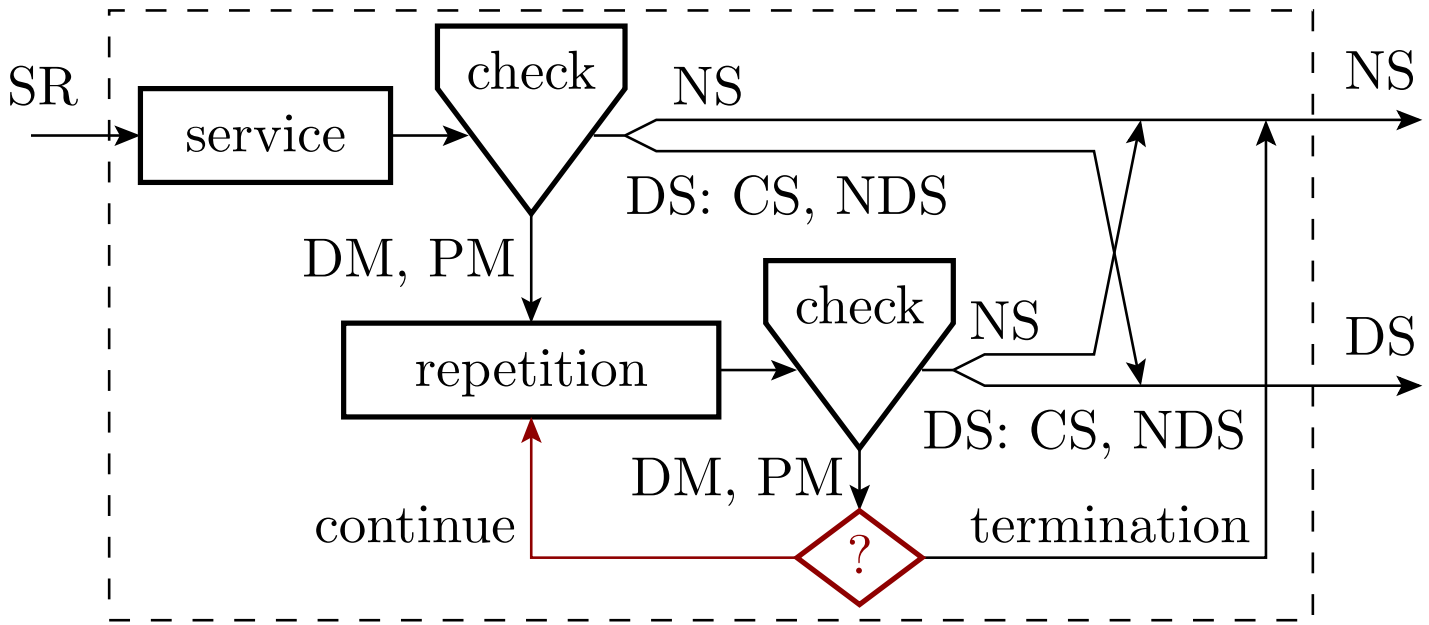
Correction

Correction by one-time repetition



- If a malfunction (DM) or phantom malfunction (PM) is detected, reinitialisation, repetition and a new check.
- Eliminate all DM and PM that do not reoccur during repetition.
- Repeated DM and PM become non-performed services (NS).

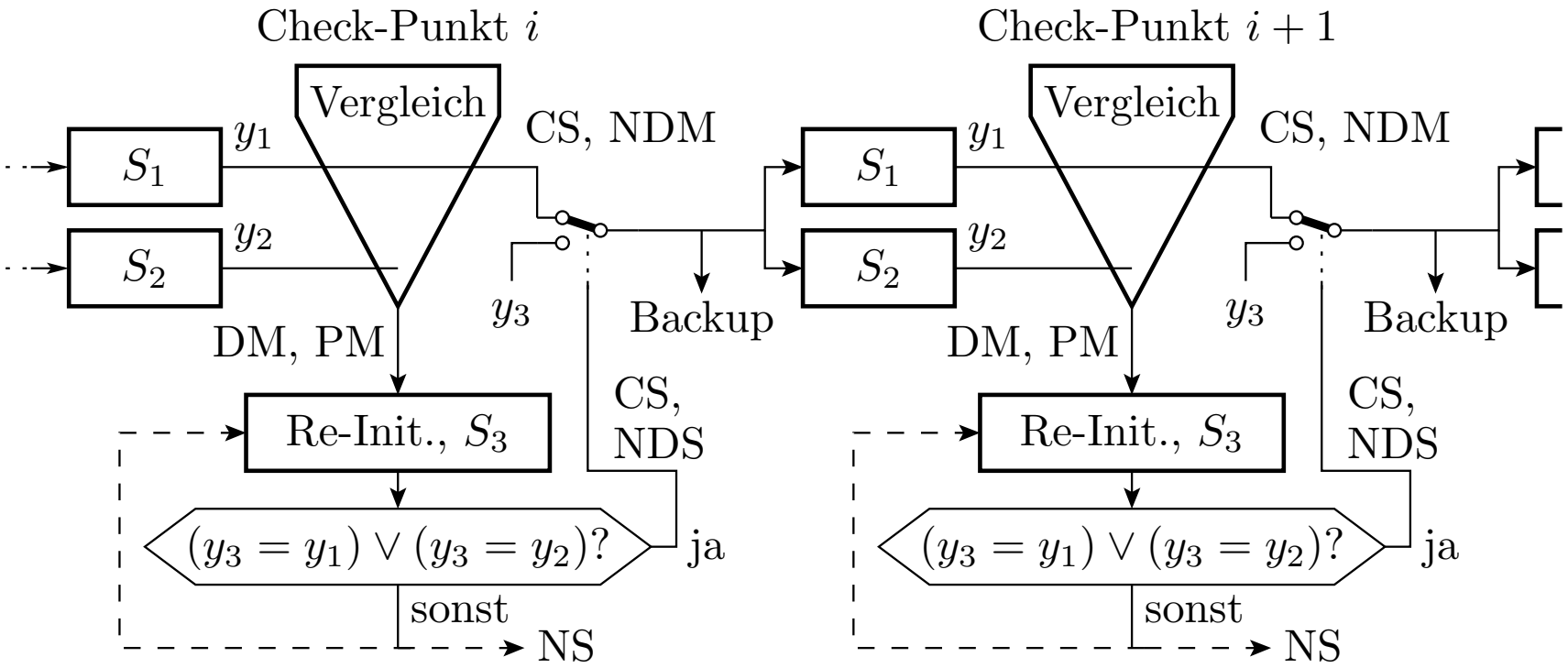
Repeat until successful



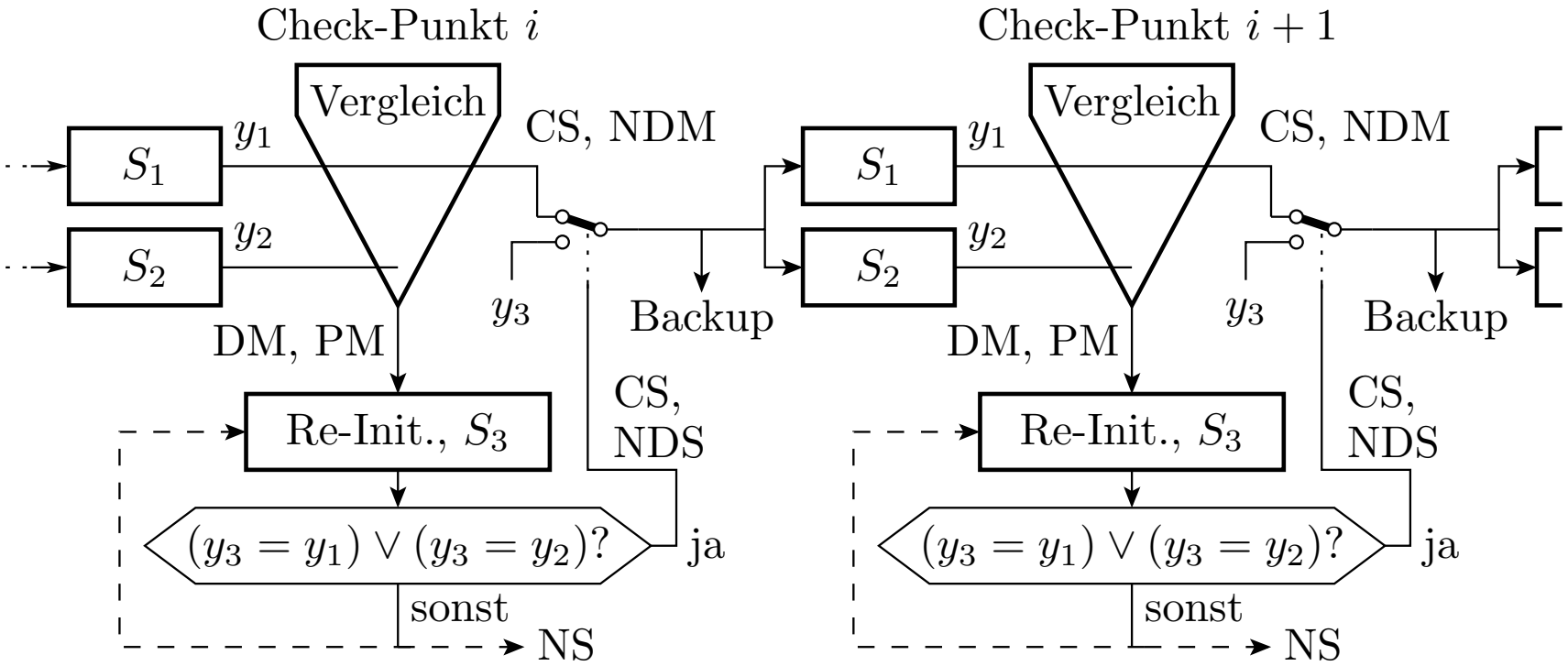
Repetition up to a termination condition:

- no recognisable falsification
- maximum number of repetitions reached
- matching falsifications due to suspected faults or failures as cause.

Master checker with repetition after MF



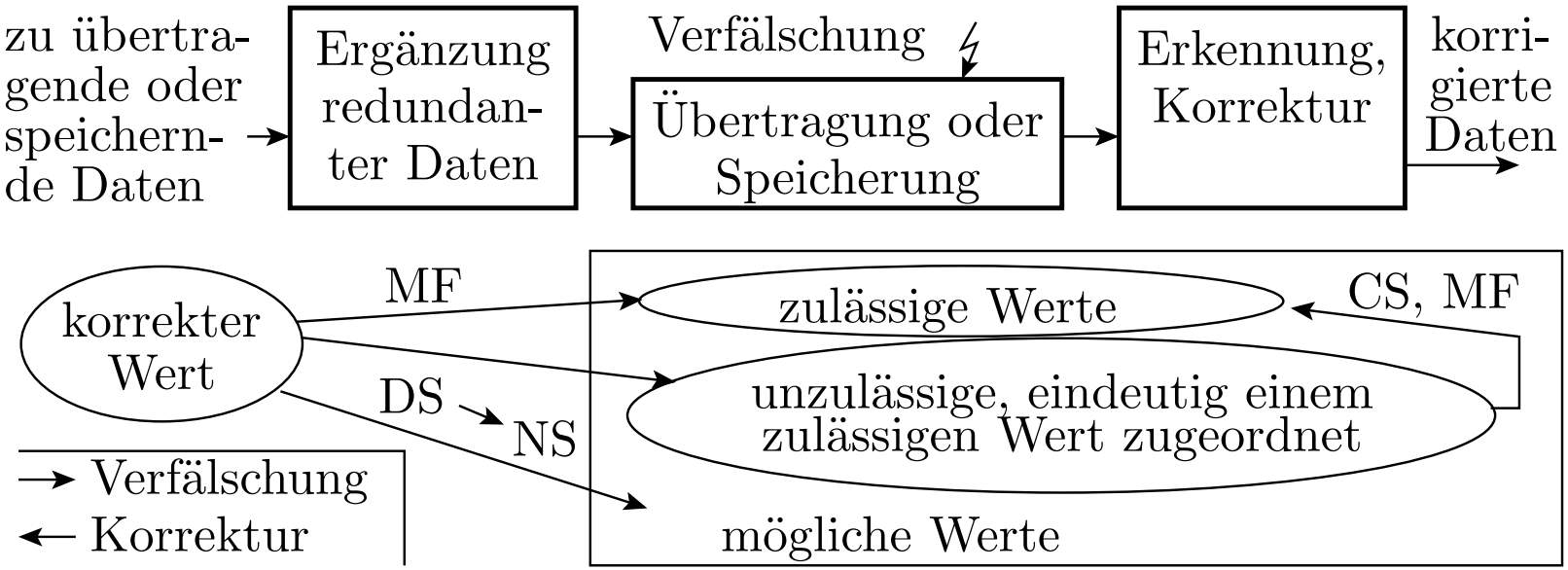
- two calculations S_1 and S_2 performed in parallel.
- At programmed checkpoints in the programme, the (intermediate) results are compared and, if they match, they are also saved in a backup memory.
- In case of a detected malfunction (DM) or phantom malfunction (PM), reinitialisation with the last saved state, ...



In case of a detected malfunction (DM) or phantom malfunction (PM)

- Re-initialisation with the last saved state,
- Perform a third calculation with the same inputs,
- If the results are the same as those of one of the two initial calculations, the result of the third calculation is used onward and saved in the backup memory.
- If no match, no service (NS) or further retries.

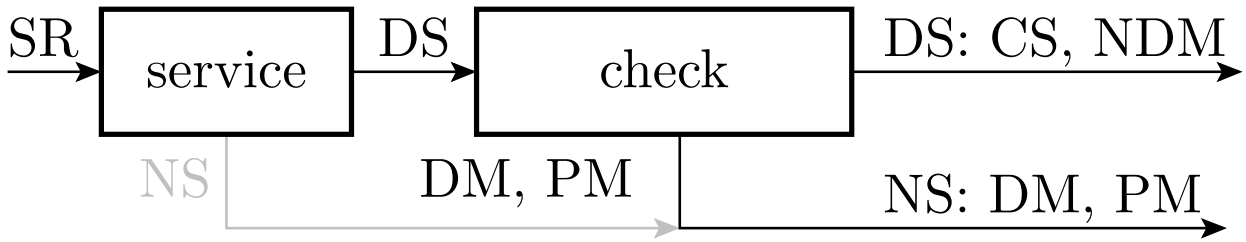
Fehlerkorrigierende Codes



- Correction of single-bit and burst corruption after data transmission and storage, also in RAIDs (see sec. 4.3.2).
- More data redundancy than fault detecting codes.
- The distortions that are taken into account are 100% corrected, others are only partially recognised and partially incorrectly corrected.

Dependability improvement

Skipping of detected malfunctions



The rate of delivered services (DS) decreases from 1 by the rate of detected and the rate of phantom malfunctions:

$$\eta_{\mathrm{DS}} = 1 - \zeta \cdot MC - \zeta_{\mathrm{Phan}} \tag{23}$$

For the MF rate, the expected number of MFs is reduced to the proportion of MFs that are not detected and the expected number of DS reduces to η_{DS}:

$$\zeta_{\mathrm{MT}} = \frac{(1 - MC) \cdot \zeta}{\eta_{\mathrm{DS}}} \tag{24}$$

η_{DS}	rate of **d**elivered **s**ervice results.
ζ	malfunction rate without mulfunction treatment.
MC	**m**alfunction **c**overage, percentage of detected malfunctions.
ζ_{Phan}	**phan**tom MF rate.

Availability and reliability

Availability according to

$$A = 1 - (1 - \eta_{\mathrm{DS}}) \cdot \frac{MTTR}{MTS} \tag{1.3}$$

with η_{DS} after sorting out detected MF on the slide before:

$$A_{\mathrm{MT}} = 1 - (\zeta \cdot MC - \zeta_{\mathrm{Phan}}) \cdot \frac{MTTR}{MTS} \tag{25}$$

The reliability as the reciprocal of the MF rate on the slide before:

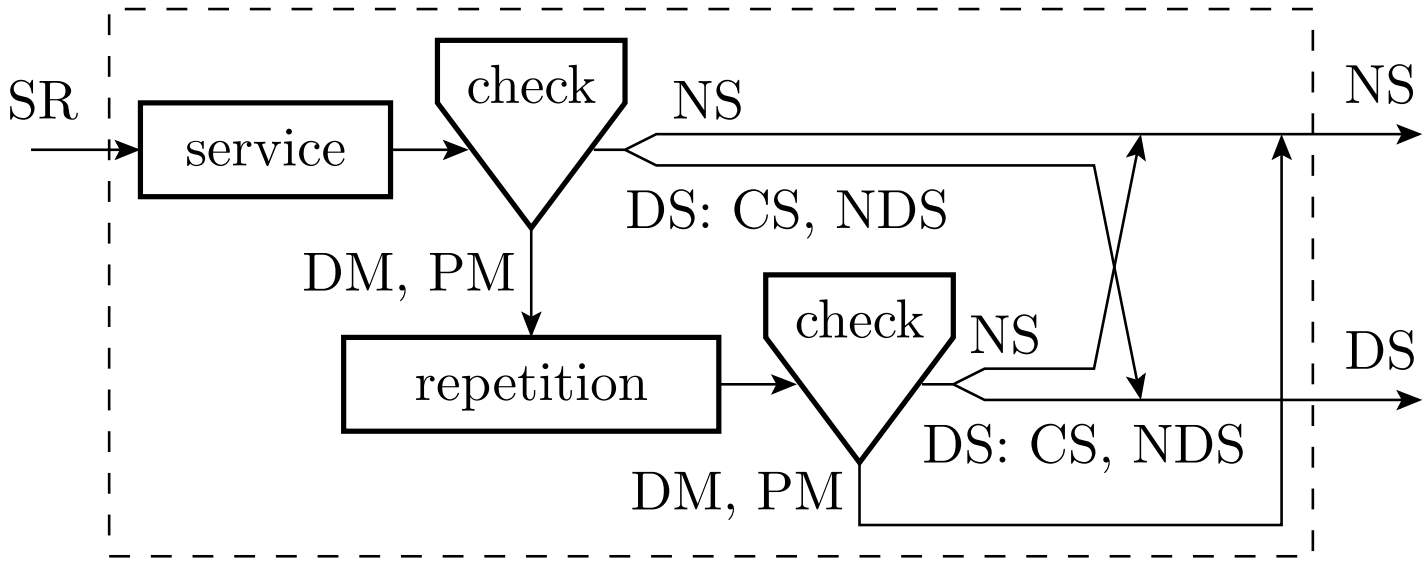
$$R_{\mathrm{MT}} = \frac{1-\zeta \cdot MC - \zeta_{\mathrm{Phan}}}{(1-MC)\cdot\zeta} \tag{26}$$

In practice, almost always applies $\zeta \ll 1$ und $\zeta_{\mathrm{Phan}} \ll 1$:

$$R_{\mathrm{MT}} = \frac{1}{(1-MC)\cdot\zeta} = \frac{R}{1-MC}$$

A_{MT}	**a**vailability with malfunction treatment.
ζ	malfunction rate without mulfunction treatment.
MC	**m**alfunction **c**overage, percentage of detected malfunctions.
ζ_{Phan}	**phan**tom MF rate.
R_{MT}	**r**eliability with **m**alfunction **t**reatment.
R	**r**eliability without mulfunction treatment.

Correction by max. one repetition



Repeat results and their relative frequencies of occurrence:

DM→NS	$1-\eta_{\mathrm{Div}}$	same wrong result
DM→NS	$\eta_{\mathrm{Div}} \cdot \zeta$	different wrong result
DM→CS	$\eta_{\mathrm{Div}} \cdot (1-\zeta)$	correction
PM→NS	$1-\eta_{\mathrm{Div}}$	same phantom malfunction
PM→CS	$\eta_{\mathrm{Div}} \cdot (1-\zeta)$	elimination of phantom MF
PM→MF	$\eta_{\mathrm{Div}} \cdot \zeta$	conversion to malfunction

MF rate after correction

DM→NS	$1 - \eta_{\mathrm{Div}}$	same wrong result
DM→NS	$\eta_{\mathrm{Div}} \cdot \zeta$	different wrong result
DM→CS	$\eta_{\mathrm{Div}} \cdot (1 - \zeta)$	correction
PM→NS	$1 - \eta_{\mathrm{Div}}$	same phantom malfunction
PM→CS	$\eta_{\mathrm{Div}} \cdot (1 - \zeta)$	elimination of phantom MF
PM→MF	$\eta_{\mathrm{Div}} \cdot \zeta$	conversion to malfunction

- Rate of delivered services:

$$\eta_{\mathrm{DS}} = 1 - \underbrace{\zeta \cdot MC \cdot (1-\eta_{\mathrm{Div}})}_{\text{same MF after repetition}} - \underbrace{\cancel{\zeta \cdot MC \cdot \eta_{\mathrm{Div}} \cdot \zeta}}_{\text{different MF after rep.}^*} - \underbrace{\zeta_{\mathrm{Phan}} \cdot (1-\eta_{\mathrm{Div}})}_{\text{same PM after rep.}} \quad (27)$$

- MF rate after correction:

$$\zeta_{\mathrm{MT}} = \frac{1}{\eta_{\mathrm{DS}}} \left(\underbrace{(1 - MC) \cdot \zeta}_{\text{not detected MF}} + \underbrace{\eta_{\mathrm{Div}} \cdot \zeta_{\mathrm{Phan}} \cdot \zeta \cdot (1 - MC)}_{\text{mot detected MF emerged from PM}} \right) \quad (28)$$

* Neglect, because very unlikely due to ζ^2.

η_{Div} **div**ersity rate, percentage MFs without common cause.

Availability and reliability

The availability

$$A = 1 - (1 - \eta_{\mathrm{DS}}) \cdot \frac{MTTR}{MTS} \tag{1.3}$$

with η_{DS} from the slide before:

$$A_{\mathrm{MT}} = 1 - (\zeta \cdot MC + \zeta_{\mathrm{Phan}}) \cdot (1 - \eta_{\mathrm{Div}}) \cdot \frac{MTTR}{MTS}$$

Compared to MF treatment without repetition, the difference to 100% is reduced to the proportion of common cause faults $1 - \eta_{\mathrm{Div}}$.
Reliability as the reciprocal of ζ_{MT}:

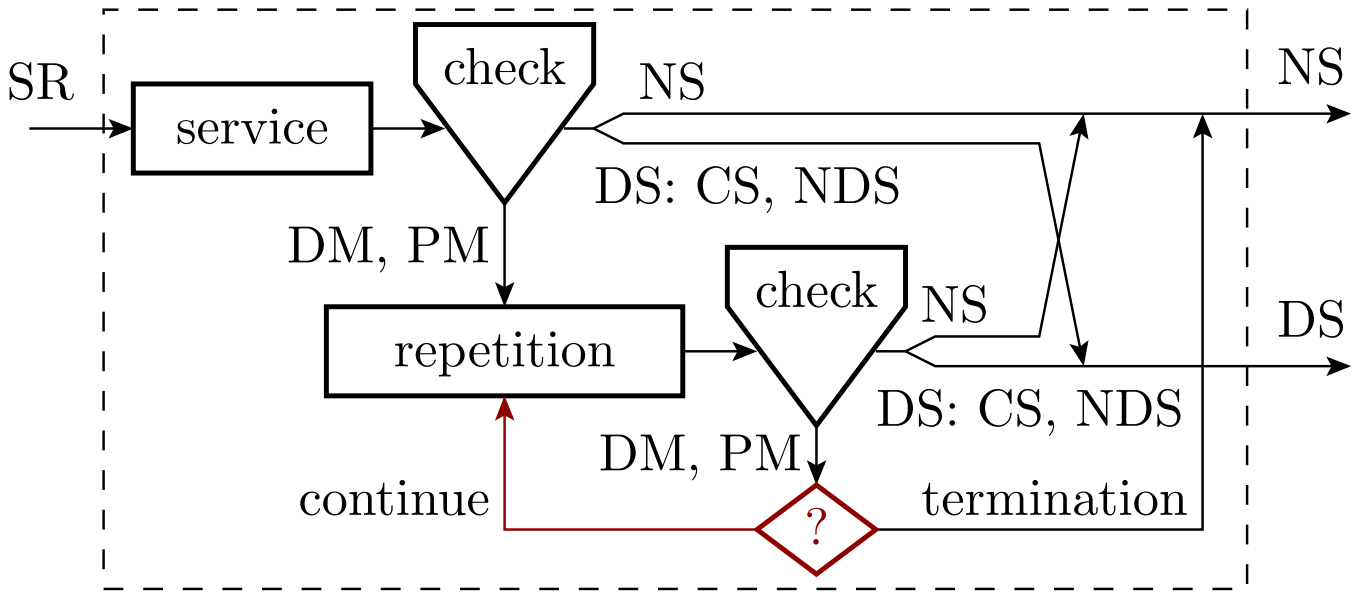
$$R_{\mathrm{MT}} = \frac{\eta_{\mathrm{DS}}}{(1 - MC) \cdot \zeta \cdot (1 + \eta_{\mathrm{Div}} \cdot \zeta_{\mathrm{Phan}})} \tag{29}$$

For $\zeta \ll 1$ and $\zeta_{\mathrm{Phan}} \ll 1$ the same reliability results as without repetition after detected malfunctions:

$$R_{\mathrm{MT}} = \frac{R}{1-MC}$$

A_{MT}	**a**vailability with malfunction treatment.
ζ	malfunction rate without mulfunction treatment.
MC	**m**alfunction **c**overage, percentage of detected malfunctions.
η_{Div}	**div**ersity rate, percentage MFs without common cause.

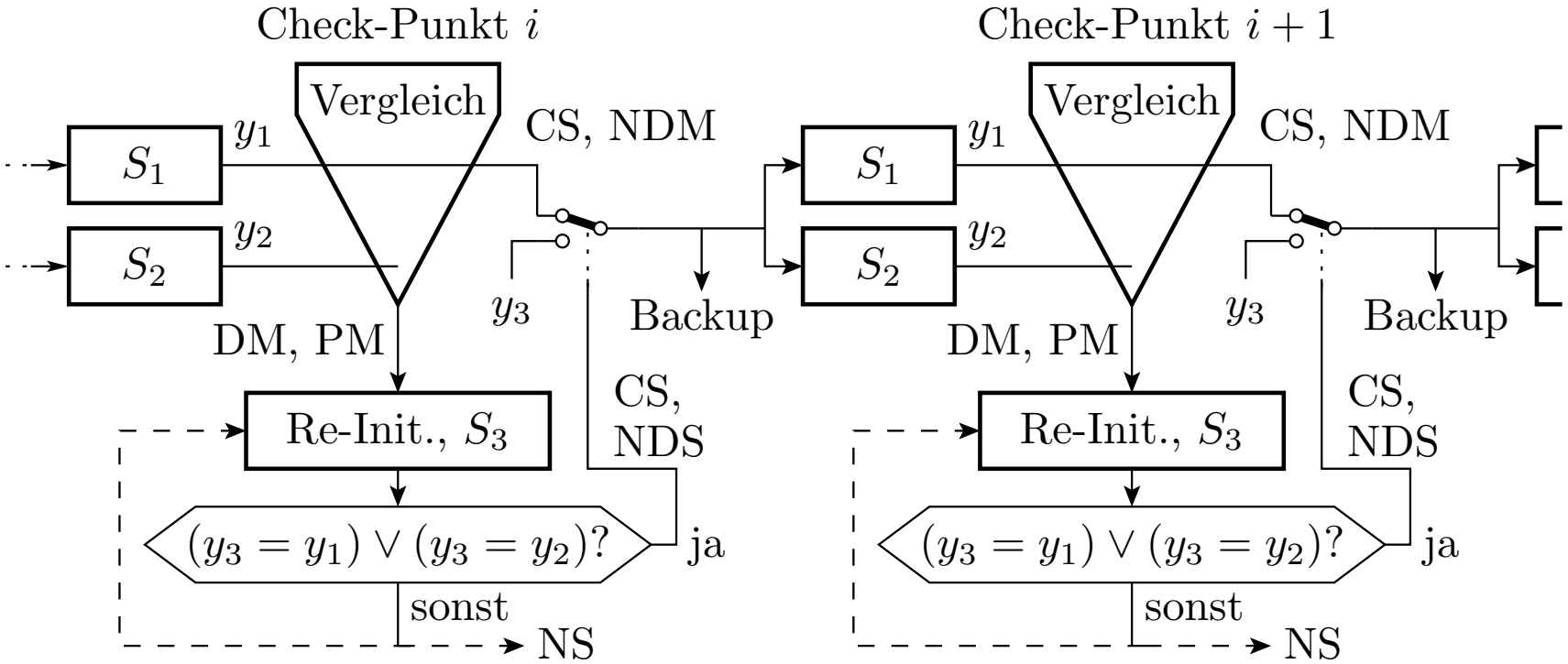
Repeat until successful



Change from only »*one repetition*«:

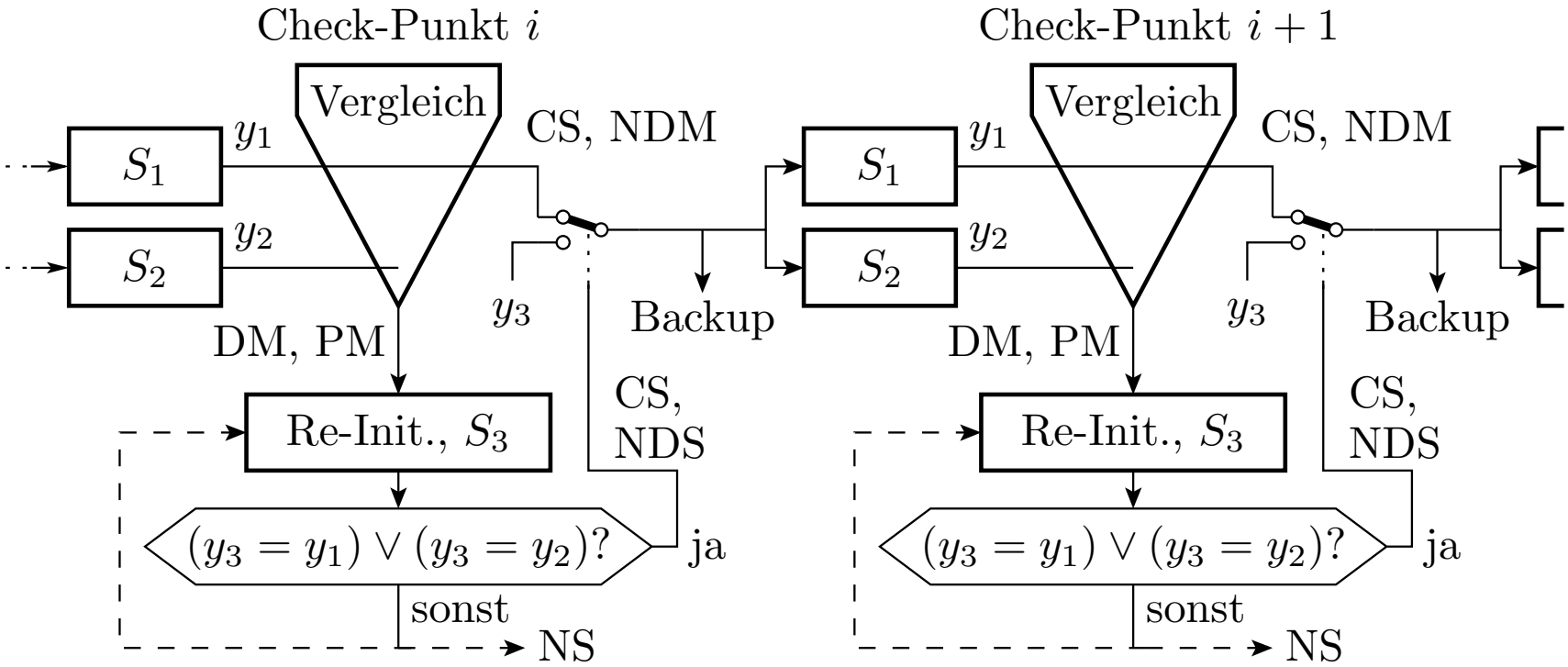
- Additional correction of the DM and PM changed after the first repetition.
- For $\zeta \ll 1$ hardly any influence on availability and reliability. Therefore only for large ζ.
- Estimation later as an exercise.

Master checker with repetition after MF



Assumptions:

- provided services (DS) have the same MS rate ζ.
- In case of a common fault as cause, all DS will be equal.
- Without a common cause, max. 1 of the 2 or 3 DS are falsified. (The rate of n MF randomly equal decreases with ζ^n.)



DM	$\zeta \cdot \eta_{\mathrm{Div}}$	S_1 wrong, S_2 correct
PM	$\zeta \cdot \eta_{\mathrm{Div}}$	S_1 correct, S_2 wrong
NDS	$2 \cdot \zeta \cdot (1 - \eta_{\mathrm{Div}})$	S_1 and S_2 equally wrong
CS	$1 - 2 \cdot \zeta$	neither S_1 nor S_2 wrong
PM→CS	1^*	elimination of phantom MF
DM→CS	1^*	conversion to malfunction

* Strictly speaking, the probability of elimination per repetition is $1 - \zeta$ and for repetition up to the termination condition it will be 1.

DM	$\zeta \cdot \eta_{\mathrm{Div}}$	S_1 wrong, S_2 correct
PM	$\zeta \cdot \eta_{\mathrm{Div}}$	S_1 correct, S_2 wrong
NDS	$\zeta \cdot (1 - \eta_{\mathrm{Div}})$	S_1 and S_2 equally wrong
CS	$1 - \zeta - \zeta \cdot \eta_{\mathrm{Div}}$	neither S_1 nor S_2 wrong
PM→CS	1	elimination of phantom MF
DM→CS	1	conversion to malfunction

- Services not provided do not exist in the model:

$$\eta_{\mathrm{DS}} = 1$$

- MF rate after correction is the common cause MF rate:

$$\zeta_{\mathrm{MT}} = \underbrace{\zeta \cdot (1 - \eta_{\mathrm{Div}})}_{S_1 \text{ and } S_2 \text{ equally wrong}}$$

- Reliability as the reciprocal of the MF rate:

$$R_{\mathrm{MT}} = \frac{R}{1-\eta_{\mathrm{Div}}} \tag{30}$$

ζ_{MT}	MF rate after **m**ulfunction **t**reatment.
ζ	Matching MF rate of the single calculations.
η_{Div}	**div**ersity rate, percentage MFs without common cause.
R	reliability without mulfunction treatment.

Summary

Monitoring parameters

- Malfunction coverage and phantom MF rate in general:

$$MC = \left.\frac{\#DM}{\#MF}\right|_{\text{ACR}} \tag{1.17}$$

$$\zeta_{\text{Phan}} = \left.\frac{\#PM}{\#DS}\right|_{\text{ACR}} \tag{1.18}$$

- Classification of monitoring into format and value checks.

Format checks:

- Uniform mapping of MF to permissible and impermissible values and r redundant bits (error-detecting codes and checksums):

$$MC \geq 1 - 2^{-r} \tag{1.21}$$

- No phantom malfunctions.
- The calculation of error-detecting code words and checksums as well as further format checks (e.g. syntax, value range, ...) will be described later in sec. 4.2.

Value control techniques:

- Master-checker as a universal method:

$$MC = \eta_{\mathrm{Div}} \tag{1.22}$$

$$\zeta_{\mathrm{Phan}} = \zeta_{\mathrm{Chk}} \cdot (1 - \eta_{\mathrm{Div}}) \tag{1.23}$$

$$\eta_{\mathrm{Div}} = 1 - \eta_{\mathrm{F}} \cdot \eta_{\mathrm{CF}} \tag{1.24}$$

  Detection of all MF by disturbances and with diversitary designs up to 90% of MF due to faults.

- Loop test: higher expected MC and less PM than Master Checker, but can only be used for reversible unique mappings.
- Correctness Checks: also often good MC, but also not usable for most applications.

Usually only format properties are monitored during operation.

Response to detected MF

- Robust response to avoid damage from MF.
- Documentation of the MF to support troubleshooting and fault elimination.
- Restoring functionality:
  - Reinitialisation (static, dynamic),
  - after failure: repair or reconfiguration,
  - If a fault is suspected: Change request, search for a workaround.
- Correction: repetition, error-detecting codes.

Avoidance of difficult-to-handle and dangerous MF e.g. by:

- Closed-circuit current principle, fault isolation,
- fire walls,...

System with MF treatment

- Termination of processing after a detected MF or PF:

$$A_{\mathrm{MT}} = 1 - (\zeta \cdot MC + \zeta_{\mathrm{Phan}}) \cdot \frac{MTTR}{MTS} \tag{1.25}$$

$$R_{\mathrm{MT}} = \frac{1-(\zeta \cdot MC + \zeta_{\mathrm{Phan}})}{(1-MC)\cdot\zeta} \tag{1.26}$$

- Master-Checker pair with abort after comparison error:

$$A_{\mathrm{MCS}} = 1 - \zeta \cdot \frac{MTTR}{MTS} \tag{1.27}$$

$$R_{\mathrm{MCS}} = \frac{R-1}{(1-\eta_{\mathrm{Div}})} \tag{1.28}$$

- Max. one repetition after MF or PM:

$$A_{\mathrm{MT}} = 1 - (\zeta \cdot MC + \zeta_{\mathrm{Phan}}) \cdot (1-\eta_{\mathrm{Div}}) \cdot \frac{MTTR}{MTS} \tag{1.29}$$

$$R_{\mathrm{MT}} = \frac{1-(\zeta \cdot MC + \zeta_{\mathrm{Phan}})\cdot(1-\eta_{\mathrm{Div}})}{(1-MC)\cdot\zeta} \tag{1.30}$$

- 3-version majority vote:

$$A_{\mathrm{MV3}} = 1 - \zeta \cdot (1-\eta_{\mathrm{Div}}) \cdot \frac{MTTR}{MTS} \tag{1.31}$$

$$R_{\mathrm{MV3}} = \frac{R}{(1-\eta_{\mathrm{Div}})} - 1 \tag{1.32}$$

For the correction methods discussed and also others, the following almost always applies in practice:

- even without MF treatment already very low MF rate $\zeta \ll 1$,
- low phantom malfunction rate $\zeta_{\mathrm{Phan}} \ll 1$ and
- conversion of PM to NDM very unlikely

and thus

- Availability reduction due to non-performed services negligible and reliability increase due to MF treatment:

$$R_{\mathrm{MT}} = \frac{R}{1-MC} \tag{31}$$

MC	**m**alfunction **c**overage, percentage of detected malfunctions.
PM	**p**hantom **m**alfunction, correct delivered service classified as malfunction.
NDM	**n**ot **d**etected **m**alfunction.

Fault elimination

Wiederholung: Gefährdungen für IT-System

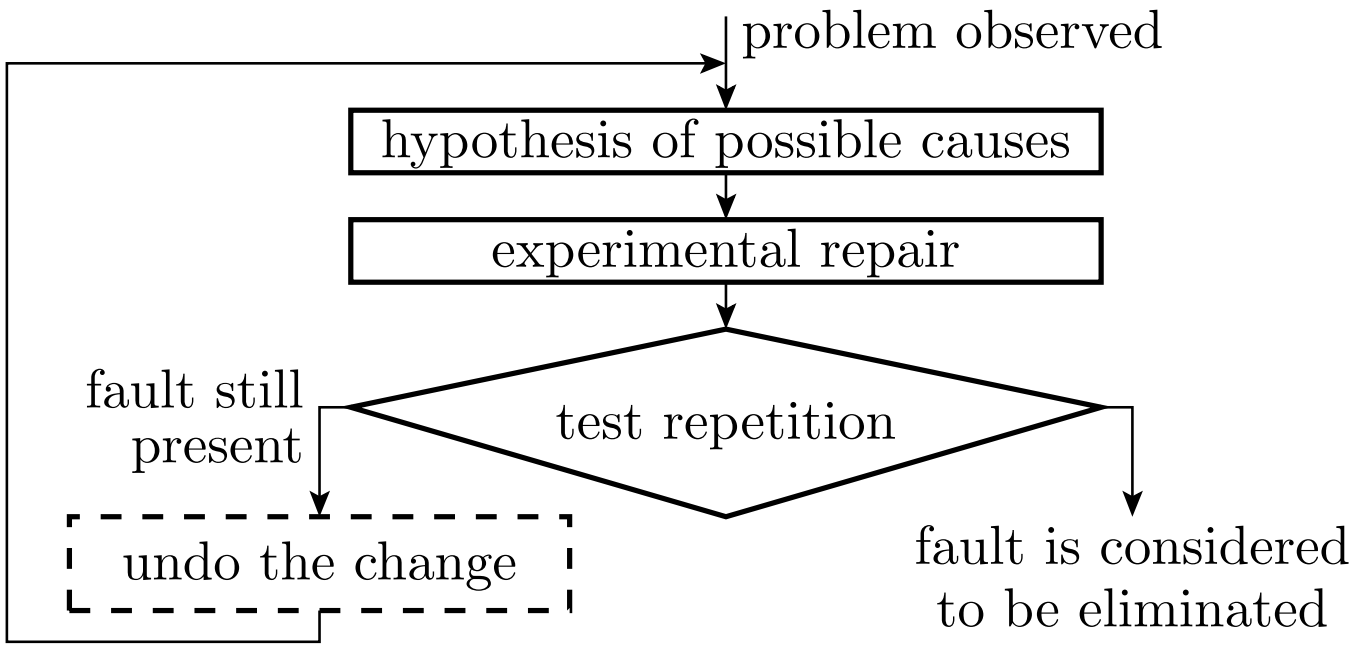
- Disturbances:
  - Random, non-reproducible cause-effect relations,
  - Hazard prevention: Monitoring and correction, usually by identical repetition.

- Faults:
  - Arising with the system or during troubleshooting,
  - Hazard prevention: Fault elimination, fault avoidance.

- Failures:
  - Faults occurring during operation,
  - Hazard prevention by MF treatment, maintenance test and redundancies (see sec. 4.3).

Elimination iteration

Troubleshooting by experimental repair

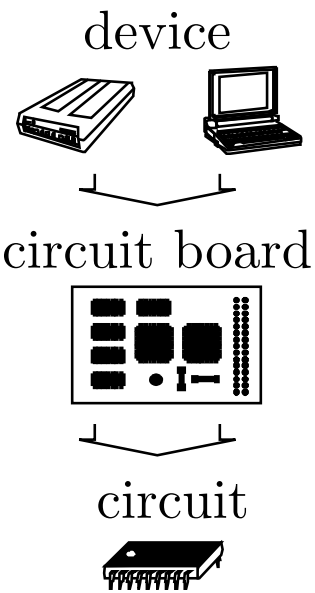


- Iteration of removal attempts for hypothetical faults and success control by test repetition.
- Removes all faults detectable by the test.
- To avoid emergence of new faults during repair undo changes after unsuccessful repair attempts.

Repair with few replaceable components

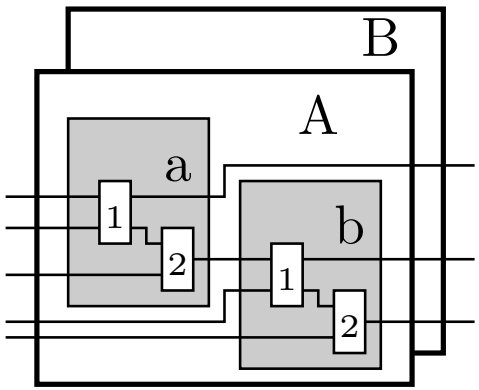
A repair-oriented system has a hierarchical structure of exchangeable components, e.g.

1. Level: Exchangeable devices.
2. Level: Exchangeable circuit boards.
3. Level: Exchangeable circuits.

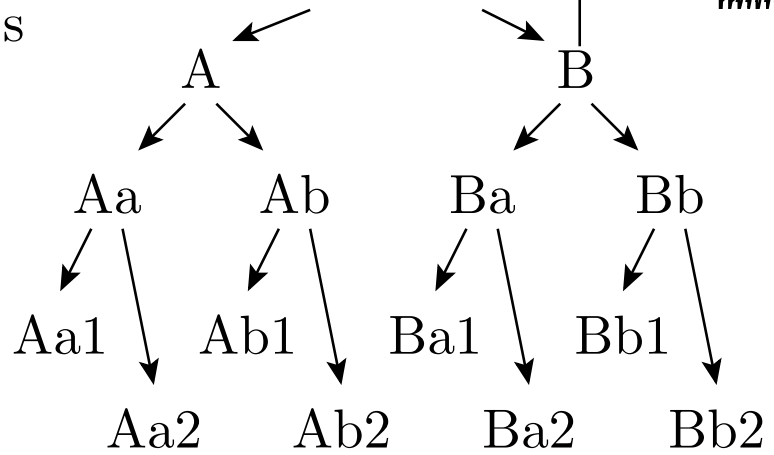


Fault localisation by systematic swapping:

hierarchical system with exchangeable components



swap tree



General practice of repair mechanics

- Rough estimation which computer part could be defective from the error symptoms.
- Check the connectors for contact problems by unplugging, cleaning, plug in again and testing.
- Replacement of possibly defective parts with spare parts and repeating the tests.

Requirements:

- Repeatable tests that prove the fault.
- Sufficient spare parts.
- General technical knowledge*.

Is an undo necessary after unsuccessful spare part installation? If so, why?

It is favourable to exchange half, of the faulty half also half, ... Why?

* Understanding the overall functionallity of the system is not mandatory.

Fault diagnosis & isolation

Fault diagnosis

Estimation of location, cause and elimination options of faults from test results to mitigate

- the number of repair attempts,
- the need for spare parts,
- the number of faults that are created during repair attempts
- including those that are not eliminated by undoing changes.

General diagnostic techniques:

- success-oriented swapping and
- Tracing of falsifications against the data or calculation flow.

The alternative to fault diagnosis is blind troubleshooting, i.e. randomly guessing the causes of MF and how to eliminate them:

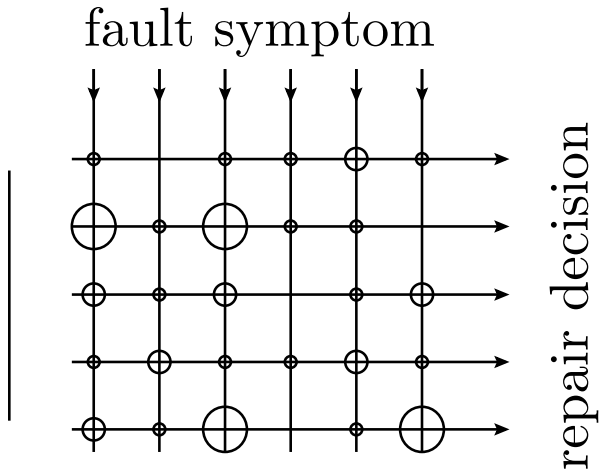
- For systems without the possibility of systematic swapping, e.g. SW and HW designs, chances of success are low, but
- in absence of documentation or system understanding the only option.

Success-oriented swapping

Products have weak points. Most problems go back to a small proportion of the possible causes, Pareto principle:

- Counting the successes of different repair alternatives.
- In case of repair, start with the most promising option.

◎ Number of times so far that the repair decision for the system has been successful

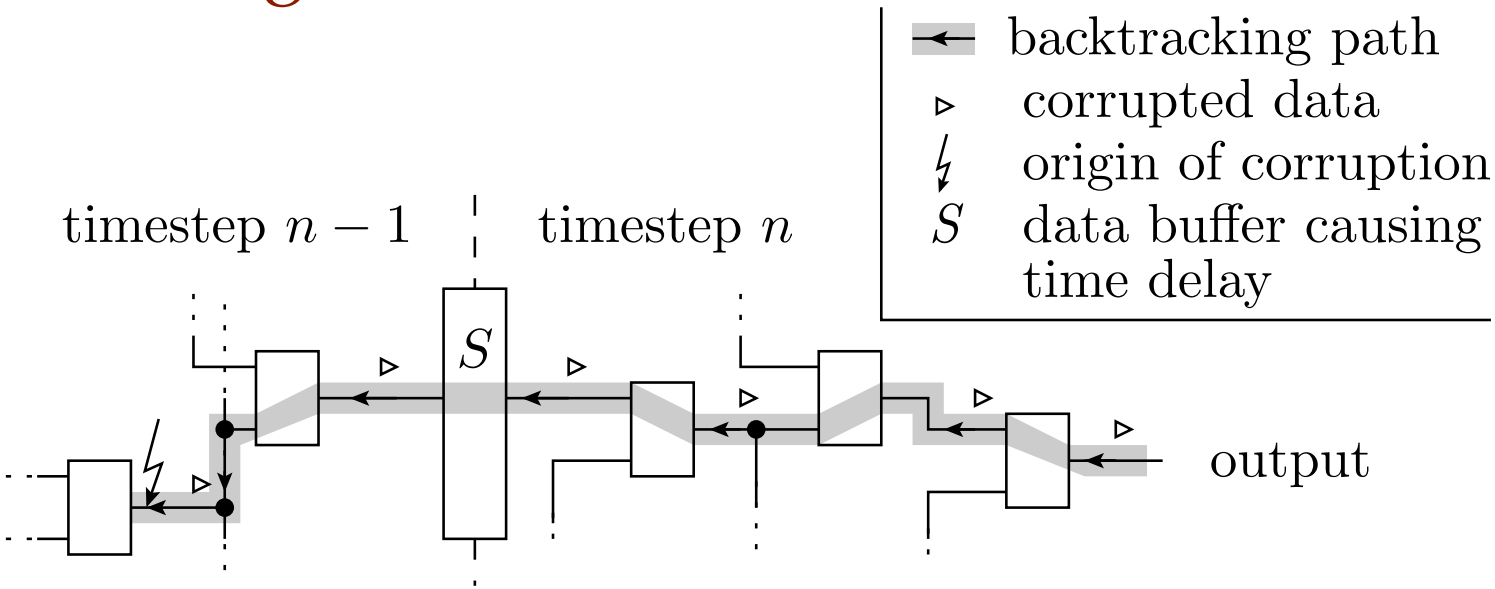


After unsuccessful repair attempts, undo change to reduce the fault generation rate during repair.

* In 1906, the Italian economist Vilfredo Pareto studied the distribution of land ownership in Italy and found that about 20% of the population owns about 80% of the land. This has entered the vernacular as the Pareto 20%-80% rule.

################ hier weiter ##########

Backtracing of data falsifications



Starting from a detected false output, backtracking against calculation or signal flow to the component that maps correct inputs to falsified outputs, if necessary over timesteps and/or hierarchically descending.

In addition to the located component, the source of the corruption can also be, for example, a short circuit in the case of HW or a misdirected write access in the case of SW.

Design for repair and testing

Collections of

- Rules of good practice, to enable / simplify testing, fault localisation and repair, and
- Antipatterns that significantly complicate the work.

Some rules of good practice:

- Modular system of exchangeable / separately testable functional blocks.
- Deterministic behaviour with directed IPO calculation flow.
- MF isolation to prevent propagation of MFs across module boundaries.
- Observability and controllability of (important) internal values.

Standard example for an antipattern:

- Big ball of mud: large, unstructured, poorly documented system that no one really understands anymore.

The lecture assumes a design style that supports test and repair.

Fault isolation

Preventing malfunctions from spreading to other subsystems:

- Physical and spatial separation of subsystems to reduce the risk of identical causes for MF (common faults, simultaneous failures, ...)
- Restrict MF propagation to information and processing flow.
- No access to data and resources of other functional units except via defined interfaces.
- Preventing incorrectly operating subsystems from damaging correctly operating subsystems.

Test

Testing

Procedure for the detection of faults. Basic classification:

- Static tests: direct control of features.
- Dynamic tests: testing the system function with a sample of example inputs.

Features controllable with static tests:

- Documentations: Comprehensibility, completeness, ...
- Software: syntax, design rules, type compatibility and API user rules, ...
- Printed circuit boards: Absence of shorts and breaks, ...

Static tests are already possible after the first design steps and during production, dynamic tests only for the finished product.

Before use, systems are usually subjected to a variety of different static and dynamic tests.

Parameters of tests

As with any control with the possible outcomes good or bad, two types of misclassification are possible:

- Non-detectable faults, modelled by the parameter fault coverage:

$$FC = \left.\frac{\#DF}{\#F}\right|_{\mathrm{ARC}} \tag{32}$$

- Phantom faults. Tests that classify correct test results as false. Modelled by the phantom MF rate during the test:

$$\zeta_{\mathrm{PhanT}} = \left.\frac{\#PM}{n}\right|_{\mathrm{ARC}} \tag{33}$$

$\#DF$	number of **d**etectable **f**aults.
$\#F$	number of **f**aults.
ζ_{PhanT}	**phan**tom MF rate during **t**est.
$\#PM$	number of **p**hantom **m**alfunction, i.e. DS classified as MF.
n	number of tests.
ACR	**a**ppropriate **c**ounting **r**anges.

Tests also need to be tested

A phantom fault (e.g. a MF during test evaluation)

- starts a superfluous elimination iteration
- in which new undetectable faults may arise.

Test results are usually checked by comparison with target values:

- Masking of faults by comparison MF and
- phantom faults due to wrong target values, ...

For newly developed tests, check that

- correct test results are classified as correct and
- incorrect test results are classified as incorrect.

If a test detects a fault, it should first be ruled out that it is not a phantom fault.

If phantom faults are handled sensibly, their influence on the total number of emergences is insignificant. We will neglect phantom faults in later developed models.

Test selection for dynamic tests

Dynamic tests practically always check the function only for a tiny sample of all possible inputs. The FC depends on the amount and selection of the test samples.

Test selection strategies:

- fault oriented,
- random with regard to the expected faults or
- a mix of fault oriented and random selection.

At the time of test selection the faults to be found and after the test the faults not found are unknown.

Without knowledge of the faults to be found:

- the fault-oriented selection and evaluation is based on fault assumptions (modeled faults or mutations), and
- the detection of the actual faults is a matter of chance.

Stuck-at faults

Modelled faults and fault model

A *fault model* is an algorithm for the calculation of a set of possible distortions that can occur in a design description.

A modelled fault is a single one of these distortions.

Determine the FC (fault simulation):

- Repeat for each test
  - Determine the target outputs
  - Repeat for all modelled faults of the fault set
    - check if the fault is distorting the output
    - if yes, mark and delete from fault list.

Fault-oriented test generation:

- Repeat for all modelled faults of the fault set
  - Search inputs for which the fault falsifies outputs

Both tasks require a lot of computational effort. For digital circuits state of the art for decades (see sec. 5.2), for SW similar developments recognisable (see sec. 6.3).

The stuck-at fault modell

The stuck-at fault model has been the most common fault model for digital circuits for several decades. In the lecture below, it will be our standard fault model.

The stuck-at fault model generates for a circuit of logic gates for all connections of all gates two model faults:

- Value constantly zero (sa0, stuck-at-0) und
- Value constantly one (sa1, stuck-at-1)

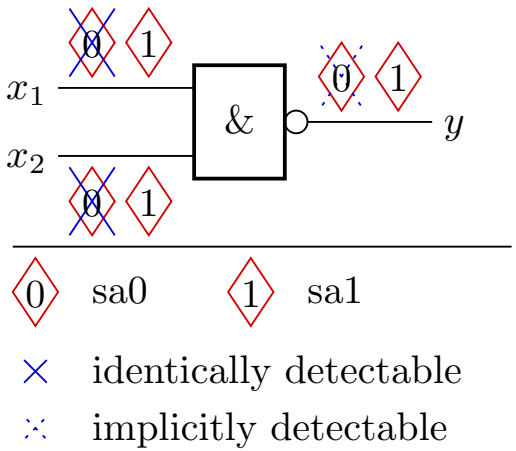
The initial fault set is reduced by identically detectable, implicitly detectable and redundant (non-detectable) fault assumptions.

In the currently developing test selection techniques for software parallels to the stuck-at-fault model can be identified (see sec. 6.3 *Test selection*).

Stuck-at faults for a logic gate

For each gate connector are assumed:

- sa0 (stuck-at-0) fault,
- sa1 (stuck-at-1) fault.



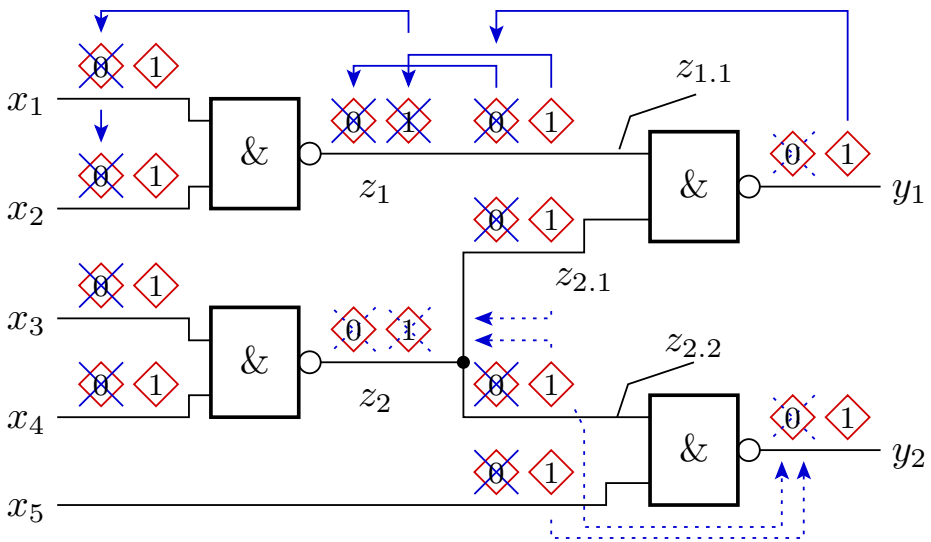
x_2	x_1	$\overline{x_2 \wedge x_1}$	sa0(x_1)	sa1(x_1)	sa0(x_2)	sa1(x_2)	sa0(y)	sa1(y)
0	0	1	1	1	1	1	0	1
0	1	1	1	1	1	0	0	1
1	0	1	1	0	1	1	0	1
1	1	0	1	0	1	0	0	1

identically detectable (same detection set)

detection implication associated input detects the fault

- Identical detectable faults are taken as one. Optional deletion of redundant and implicitly detectable model faults.
- The generated fault set contains similarly detectable model faults for all potential faults of the real circuit (see sec. 5.1.4 *Detection relations*).

Identical and implicitly detectable faults in the circuit network



identically detectable

implicity detectable

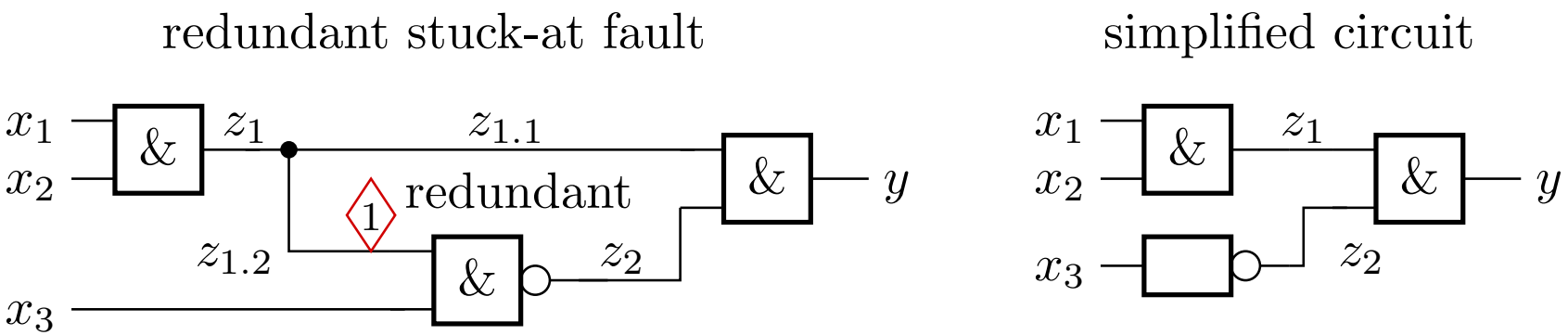
24 initial faults
14 not identicaly detectable faults
10 without implicitly detectable faults

	set of identical detectable faults	implicit detectable
1	sa0(x_1), sa0(x_2), sa1(z_1), sa1($z_{1.1}$)	
2	sa1(x_1)	
3	sa1(x_2)	
4	sa0(x_3), sa0(x_4), sa1(z_2)	9, 12
5	sa1(x_3)	
6	sa1(x_4)	
7	sa0(z_2)	5, 6, 8, 11
8	sa0(z_1), sa0($z_{1.1}$), sa0($z_{2.1}$), sa1(y_1)	2, 3
9	sa1($z_{2.1}$)	
10	sa0(y_1)	1, 9
11	sa0($z_{2.2}$), sa0(x_5), sa1(y_2)	
12	sa1($z_{2.2}$)	
13	sa1(x_5)	
14	sa0(y_2)	12, 13

Redundant faults

Definition redundant (modelled) fault

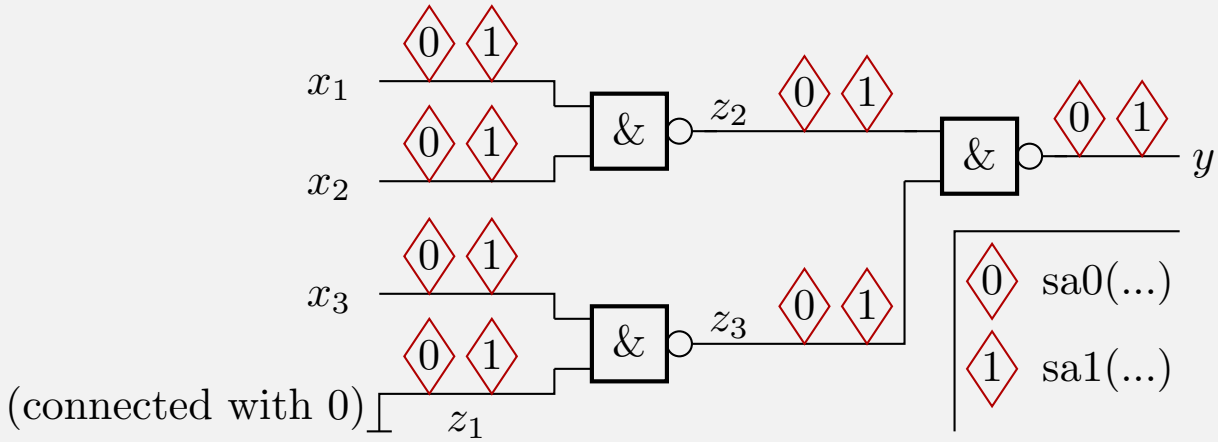
Falsification of the system description, which does not affect the function and is therefore not detectable with dynamic tests.



- Error excitation requires $z_1 = 0$ and observability of z_2 at y requires $z_2 = 1$. No input $x_3x_2x_1$ can prove the fault.
- The elimination of redundant faults also serves to simplify the system description..

Example 1.4: stuck-at fault set

Circuit with 12 stuck-at faults:

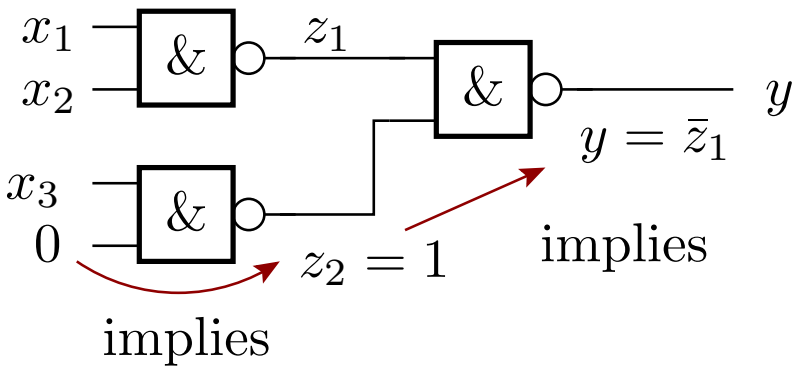


a) Determine gate circuit and initial stuck-at fault set after redundancy elimination.
b) Remove from remaining fault set identically and implicitly detectable stuck-at faults.

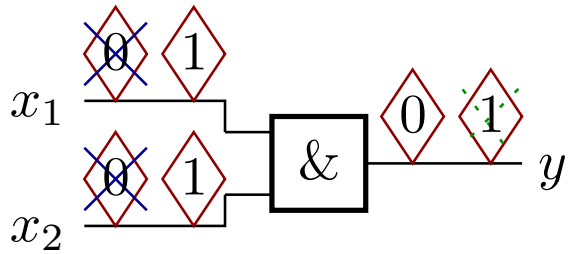
a) Determine gate circuit and initial stuck-at fault set after redundancy elimination.
b) Remove from remaining fault set identically and implicitly detectable stuck-at faults.

a) The function does not depend on x_3 and is: $y = x_1 \wedge x_2$

simplification option

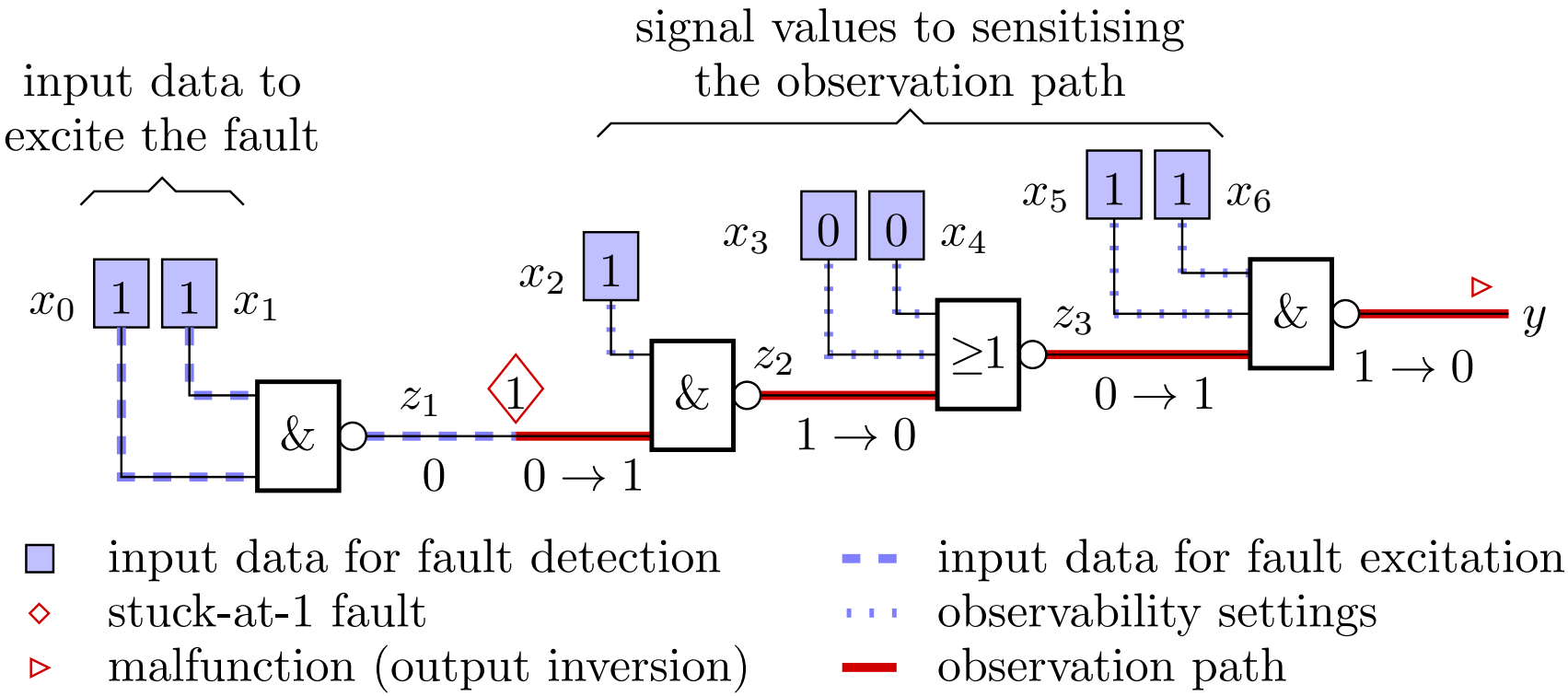


reduction of the amount of faults for the simplified circuit



b) At the remaining AND gate, sa0(x_i) are identically detectable with sa0(y) and the detection of sa1(x_1) and sa1(x_2) implies that of sa1(y).

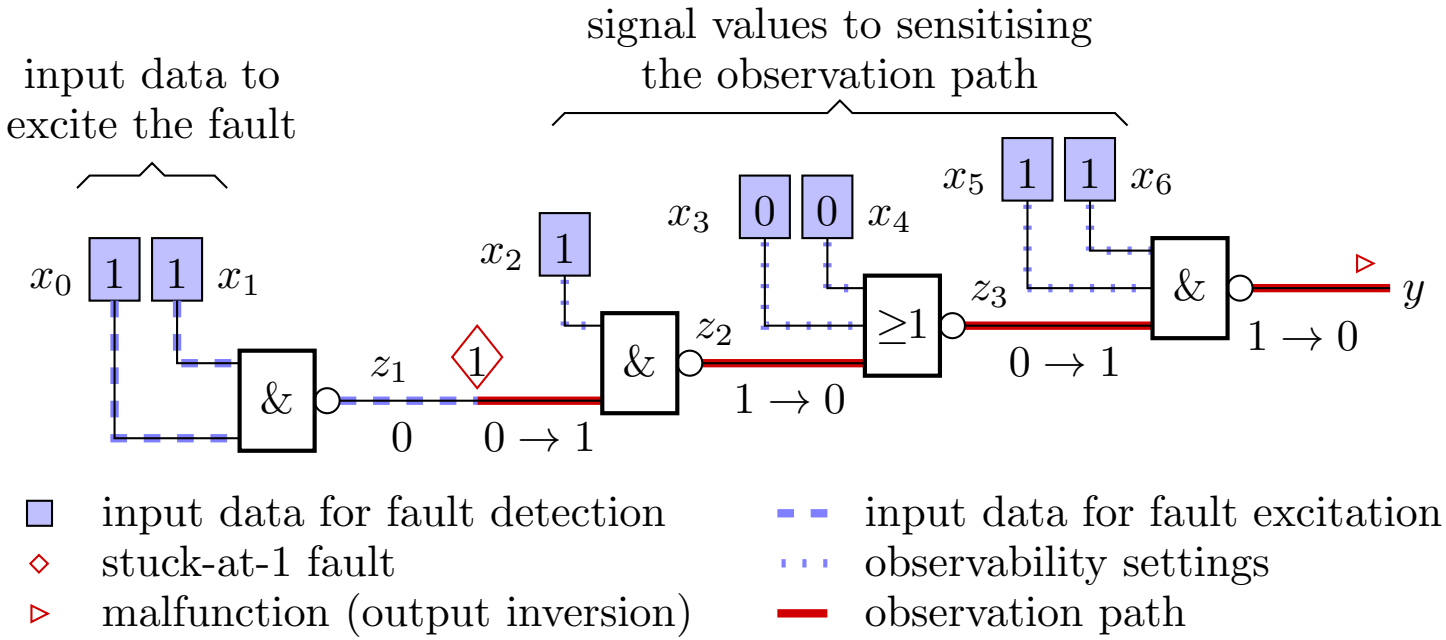
Test search and detection probability



Search by path sensitisation(see sec. 5.2.2 *D algorithm*):

- Search of inputs for setting »0« at the error location and
- Sensitising an observation path to an output.

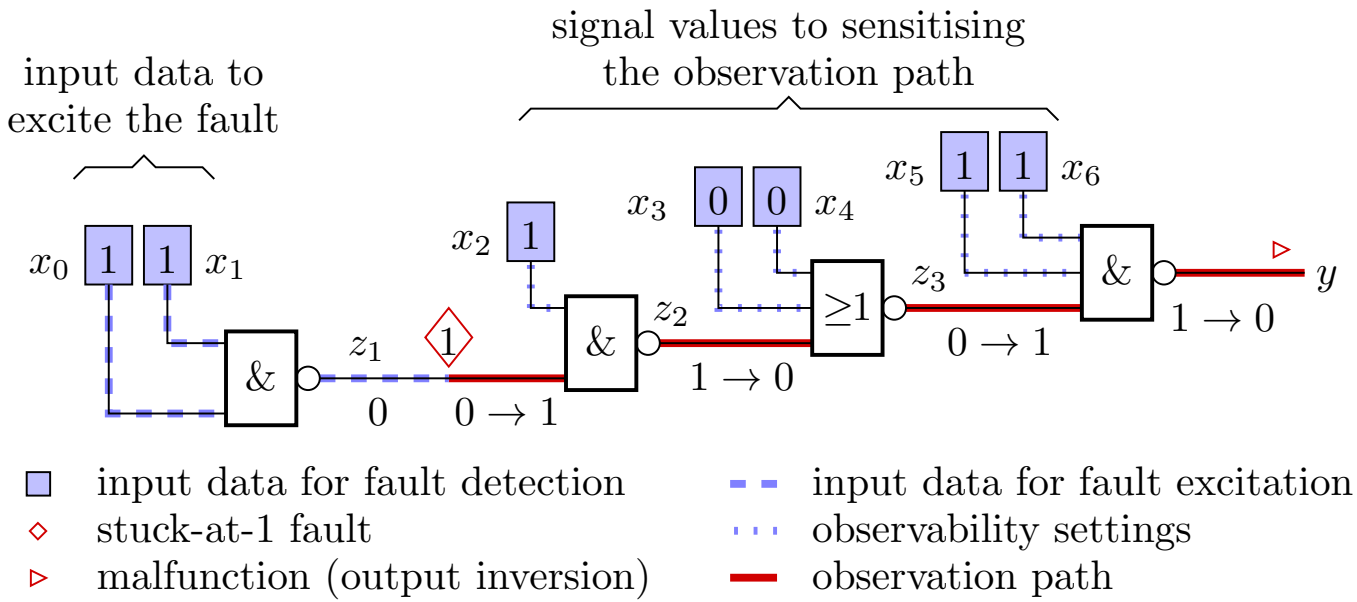
Fault detection sets



Fault detection set (input sets for the fault detection):

Input set fault stimulation: $M_{\mathrm{FS}} = \{\text{-----11}\}$
Input set observability: $M_{\mathrm{FO}} = \{\text{11001--}\}$
Input set fault detection: $M_{\mathrm{FS}} \cap M_{\mathrm{FO}} = \{1100111\}$

Fault detection probability



Random test (assumption all 128 inputs are equally likely):

- fault exciting with $2^5 = 32$ of 128 inputs: $p_{\mathrm{FS}} = 2^{-2}$
- observable with $2^2 = 4$ of 128 inputs: $p_{\mathrm{FO}} = 2^{-5}$
- detectable with one of 128 inputs: $p_{\mathrm{FD}} = p_{\mathrm{FS}} \cdot p_{\mathrm{FO}} = 2^{-7}$

p_{FS} **p**robability of **f**ault **s**timulation.
p_{FO} **p**robability of **f**ault **o**bservation.
p_{FD} **p**robability of **f**ault **d**etection.

Reliability after fault elim.

Reliability after elimination of all detected faults

Example 1.5: not eliminated faults

Programme size 10.000 NLOC. 30 ... 100 faults per 1000 NLOC. Fault coverage of the test $FC = 70\%$. Expected number of faults after elimination of all detectable faults:

$$10.000\,\mathrm{NLOC} \cdot \frac{30\ [\mathrm{F}]\ \ldots\ 100\,[\mathrm{F}]}{1000\,\mathrm{NLOC}} \cdot (1 - 70\%) = 100\,[\mathrm{F}]\ \ldots\ 300\ [\mathrm{F}]$$

How reliable is a system with 100 to 300 faults?

Preview: In the case of a random test and elimination of all detected faults, the fault-related partial reliability R_{F} increases proportionally to the number of dynamic tests n and inversely proportionally to the expected number of faults not eliminated μ_{FNE}:

$$R_{\mathrm{F}} \sim \frac{n}{\mu_{\mathrm{FNE}}}$$

$[F]$ value in **f**aults.

Malfunction rate caused by faults

Each not eliminated fault i causes malfunctions at the MF rate ζ_i (in MF per DS). The sum of the MF rates of all faults

$$\zeta_{\sum} = \sum_{i=1}^{\#F_{\mathrm{NE}}} \zeta_i$$

is an upper limit $\zeta \leq \zeta_{\sum}$ and for $\zeta_{\sum} \ll 1$ and, if almost all MF only have one fault as cause, practically equal to the MF rate by all faults:

$$\zeta_{\mathrm{F}} = \sum_{i=1}^{\#F_{\mathrm{NE}}} \zeta_i \quad \text{für} \quad \zeta \ll 1$$

$\#F_{\mathrm{NE}}$	number of **f**aults **n**ot **e**liminated.
ζ_i	MF rate caused by fault i.
ζ_{F}	malfunction rate caused by **f**aults.

Simple estimation

Under the assumptions:

- elimination of all detectable faults,
- mean MF rate $\bar{\zeta} \leq 1/n$ per fault not eliminated

the MF rate for all not eliminated faults together is max.:

$$\zeta_{\mathrm{F}} \leq \frac{\mu_{\mathrm{FNE}}}{n}$$

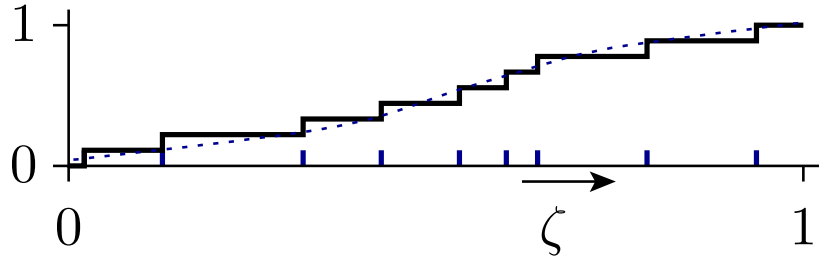
Die fehlerbezogene Teilzuverlässigkeit beträgt mindestens:

$$R_{\mathrm{F}} \geq \frac{n}{\mu_{\mathrm{FNE}}}$$

$\bar{\zeta}$	mean malfunction rate per fault.
n	number of tests.
ζ_{F}	malfunction rate caused by **f**aults.
μ_{FNE}	expected number of **n**ot **e**liminated **f**aults.
R_{F}	**f**aults-related partial **r**eliability.

A more precise estimation

$F(\zeta) = \mathbf{P}(Z \leq \zeta)$



step function for a finite number of faults

approximation by a continuous function

The MF rate distribution describes for each value $\zeta \in (0, 1)$ the probability that Z is not greater than ζ. When $F_Z(\zeta)$ is approximated by a continuous distribution function, the density of the MF rate is (see later slide set 3):

$$h(\zeta) = \frac{\mathrm{d}F(\zeta)}{\mathrm{d}\zeta} \text{ mit } \int_0^1 h(\zeta) \cdot d\zeta = 1$$

Z	malfunction rate, random variable.
ζ	value of the malfunction rate.
$F(\zeta)$	Distribution function of the malfunction rate.
$h(\zeta)$	Density function of the malfunction rate.

The mean malfunction rate per fault is the expected value of the malfunction rate Z (see later eq. 3.6):

$$\bar{\zeta} = \mathbb{E}[Z] = \int_0^1 h(\zeta) \cdot \zeta \cdot d\zeta$$

and the total MF rate due to faults is the expected number of faults not eliminated times the mean MF rate per fault:

$$\zeta_{\mathrm{F}} = \sum_{i=1}^{\#F_{\mathrm{NE}}} \zeta_i = \mu_{\mathrm{FNE}} \cdot \bar{\zeta}$$

$\bar{\zeta}$	mean malfunction rate per fault.
Z	malfunction rate, random variable.
ζ	value of the malfunction rate.
$h(\zeta)$	Density function of the malfunction rate.
$\#F_{\mathrm{NE}}$	number of **f**aults **n**ot **e**liminated.
ζ_i	MF rate caused by fault i.
μ_{FNE}	expected number of **n**ot **e**liminated **f**aults.

Typical fault coverage of random tests

For a random test, a reduction in the proportion of undetectable faults $1 - FC(n)$ by one decade requires an increase in the number of tests n by more than one decade. This is the property of a power function:

$$1 - \mathbb{E}\left[FC\left(n\right)\right] = \left(\frac{n}{n_{\min}}\right)^{-k} \quad \text{mit } n \geq n_{\min} \text{ und } 0 < k < 1 \tag{34}$$

k	1	0,5	0,33	0,25
$\frac{n}{n_{\min}}$ for $1 - \mathbb{E}\left[FC\left(n\right)\right] = 0{,}1$	10	100	10^3	10^4

FC	**f**ault **c**overage, percentage of detectable faults.
$n_{\min}$	reference test number for $FC = 0$.
n	number of tests including $n_{\min}$.
k	form factor of the distribution of the malfunction rate $(0 < k < 1)$.

Distribution function of the FM rate

With the simplification* that a random test of length n detects all faults with a MF rate $\zeta \geq \frac{1}{n}$, the distribution function of the MF rate before eliminating faults detectable with further tests is:

$$F(\zeta) = 1 - \mathbb{E}\left[FC\left(\zeta = \frac{1}{n}\right)\right] = \begin{cases} (n_{\min} \cdot \zeta)^k & 0 \leq \zeta < \frac{1}{n_{\min}} \\ 1 & \zeta \geq \frac{1}{n_{\min}} \end{cases}$$

Then, when all faults detectable with a test number $n \geq n_{\min}$ are eliminated, n takes the role of $n_{\min}$ in the equation before:

$$F(\zeta) = \begin{cases} (n \cdot \zeta)^k & 0 \leq \zeta < \frac{1}{n} \\ 1 & \zeta \geq \frac{1}{n} \end{cases} \tag{35}$$

$F(\zeta)$	Distribution function of the malfunction rate.
FC	**f**ault **c**overage, percentage of detectable faults.
$n_{\min}$	reference test number for $FC = 0$.
n	number of tests for which all detected faults are eliminated including $n_{\min}$.
k	form factor of the distribution of the malfunction rate $(0 < k < 1)$.
*	in fact n is the mean detection length for faults with $\zeta_i = \frac{1}{n}$, not the exact one.

MF rate density, mean MF rate

Distribution function when all faults with $\zeta \geq 1/n$ are eliminated:

$$F(\zeta) = \begin{cases} (n \cdot \zeta)^k & 0 \leq \zeta < \frac{1}{n} \\ 1 & \zeta \geq \frac{1}{n} \end{cases} \tag{1.37}$$

Density of the MF rate and mean MF rate per not eliminated fault:

$$h(\zeta) = \frac{\mathrm{d}F(\zeta)}{\mathrm{d}\zeta} = \begin{cases} k \cdot n^k \cdot \zeta^{k-1} & 0 \leq \zeta < \frac{1}{n} \\ 0 & \text{sonst} \end{cases} \tag{36}$$

$$\bar{\zeta} = \int_0^{\frac{1}{n}} h(\zeta) \cdot \zeta \cdot \mathrm{d}\zeta = \frac{k}{(k+1) \cdot n} \tag{37}$$

$F(\zeta)$	Distribution function of the malfunction rate.
$h(\zeta)$	Density function of the malfunction rate.
k	form factor of the distribution of the malfunction rate $(0 < k < 1)$.
n	number of tests for which all detected faults are eliminated including $n_{\min}$.
ζ	value of the malfunction rate.
$\bar{\zeta}$	mean malfunction rate per fault.

Expected number of Faults and MF rate

With the postulated decrease in the proportion of undetectable faults

$$1 - \mathbb{E}\left[FC\left(n\right)\right] = \left(\frac{n}{n_{\min}}\right)^{-k} \tag{1.36}$$

the expected number of faults that are not eliminated is

$$\mu_{\mathrm{FNE}}\left(n\right) = \mu_{\mathrm{FNE}}\left(n_{\min}\right) \cdot \left(\frac{n}{n_{\min}}\right)^{-k}$$

and the MF rate due to faults is:

$$\zeta_{\mathrm{F}} = \mu_{\mathrm{FNE}}\left(n\right) \cdot \bar{\zeta} = \frac{\mu_{\mathrm{FNE}}\left(n\right) \cdot k}{\left(k+1\right) \cdot n}$$

μ_{FNE}	expected number of **n**ot **e**liminated **f**aults.
$n_{\min}$	reference test number for $FC = 0$.
n	number of tests for which all detected faults are eliminated including $n_{\min}$.
k	form factor of the distribution of the malfunction rate $(0 < k < 1)$.
ζ_{F}	malfunction rate caused by not eliminated **f**aults.

Elimination probabilities for real tests

Instead of eliminating all faults with$\zeta \geq \frac{1}{n}$ a non-elimination probability $p_{\mathrm{FNE}}(\zeta)$ depending on ζ is assumed:

$$\zeta_{\mathrm{F}} = \mu_{\mathrm{FNE}}(n_{\min}) \cdot \int_0^{n_{\min}} p_{\mathrm{FNE}}(\zeta) \cdot \zeta \cdot h_Z(\zeta) \cdot \mathrm{d}\zeta$$

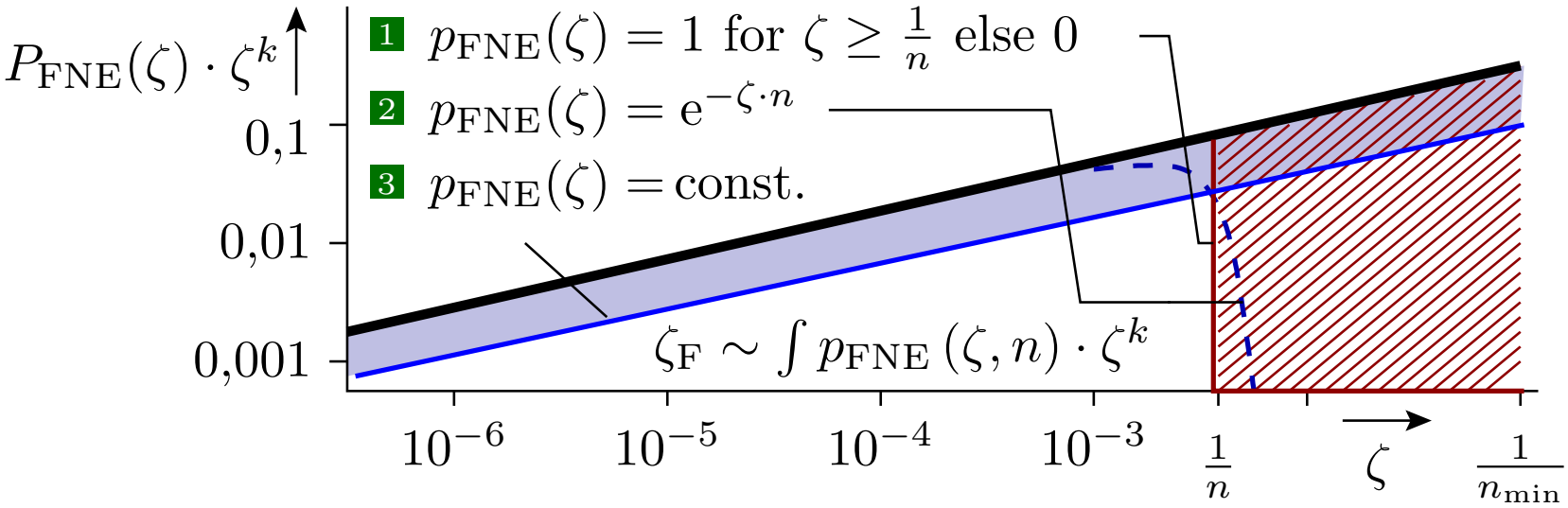
For the assumed density of the MF rate before removal iteration and minimum MF rate $\zeta \leq \frac{1}{n_{\min}}$:

$$h_Z(\zeta, n_{\min}) = k \cdot n_{\min}^k \cdot \zeta^{k-1} \quad \text{für } 0 \leq \zeta \leq \frac{1}{n_{\min}} \text{ sonst } 0$$

the malfunction rate is proportional to the area under the curve:

$$p_{\mathrm{FNE}}(\zeta) \cdot \zeta^k$$

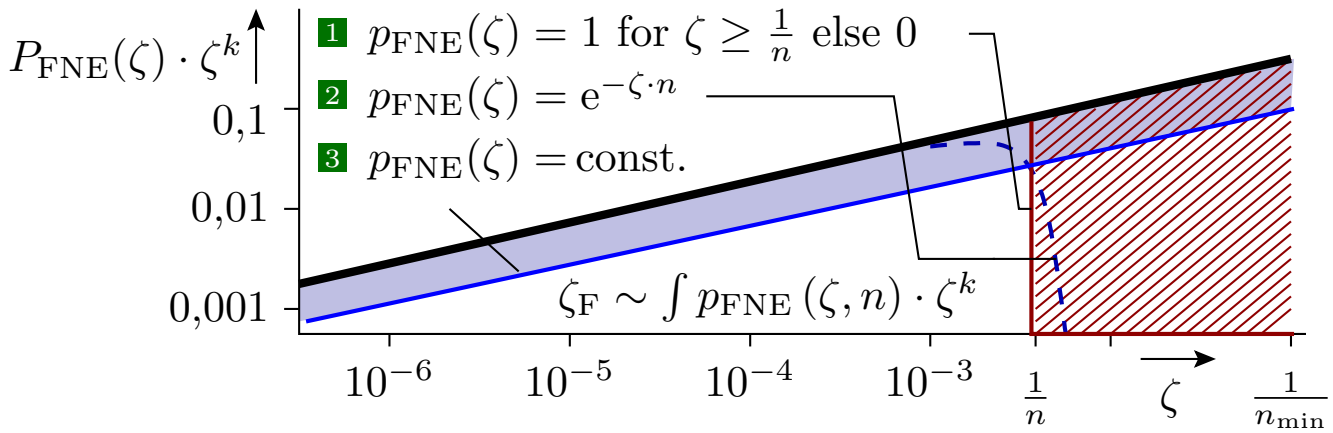
p_{FNE}	**p**robability of **f**ault **n**ot **e**liminated.
ζ_{F}	malfunction rate caused by not eliminated **f**aults.
μ_{FNE}	expected number of **n**ot **e**liminated **f**aults.
ζ	value of the malfunction rate.
$h(\zeta)$	Density function of the malfunction rate.



1. Proof exactly with $n = \frac{1}{\zeta}$ tests (our simplification):

$$\begin{aligned}
\zeta_{\mathrm{F}}(n) &= \mu_{\mathrm{FNE}}(n_{\min}) \cdot \int_0^{\frac{1}{n}} h(\zeta, n_{\min}) \cdot \zeta \cdot \mathrm{d}\zeta \\
&= \mu_{\mathrm{FNE}}(n_{\min}) \cdot \int_0^{\frac{1}{n}} k \cdot n_{\min}^k \cdot \zeta^{k-1} \cdot \zeta \cdot \mathrm{d}\zeta \\
&= \mu_{\mathrm{FNE}}(n_{\min}) \cdot \left(\frac{n}{n_{\min}}\right)^k \cdot \frac{k}{(k+1) \cdot n} = \frac{\mu_{\mathrm{FNE}}(n) \cdot k}{(k+1) \cdot n}
\end{aligned}$$

2. Detection with mean $\frac{1}{\zeta}$ tests.
3. detection probability p_{FNE} independent of ζ.



1. Proof exactly with $n = \frac{1}{\zeta}$ tests (our simplification):

$$\zeta_{\mathrm{F}}(n) = \frac{\mu_{\mathrm{FNE}}(n) \cdot k}{(k+1) \cdot n}$$

2. Detection with mean $\frac{1}{\zeta}$ tests (tatsächliches Verhalten*):

$$\zeta_{\mathrm{F}} = \mu_{\mathrm{FNE}}(n_{\min}) \cdot \int_0^1 \mathrm{e}^{-\zeta \cdot n} \cdot h(\zeta, n_{\min}) \cdot \zeta \cdot \mathrm{d}\zeta$$

$$= \mu_{\mathrm{FNE}}(n_{\min}) \cdot \int_0^1 \mathrm{e}^{-\zeta \cdot n} \cdot k \cdot n_{\min}^k \cdot \zeta^{k-1} \cdot \zeta \cdot \mathrm{d}\zeta = \frac{\mu_{\mathrm{FNE}}(n) \cdot k}{\cancel{(k+1)} \cdot n}$$

3. $p_{\mathrm{FNE}} \neq f(\zeta) \rightarrow \zeta_{\mathrm{F}} \sim p_{\mathrm{FNE}}$.

* The derivation follows later (see sec. 3.5 *Gamma distribution*). We nevertheless omit the term $k+1$ in the denominator in the following as well.

Splitting into pre-test and random test

Before the thorough random test, pre-tests are carried out:

- static tests: reviews, syntax, ...
- a few rough tests to see whether anything works at all and
- some specifically sought tests for limits and special cases..

For static and fault-oriented searched tests, p_{FNE} depends less on ζ than for the random test. Blanket assumption that pre-tests detect a proportion of FC_{PT} faults, of which are all eliminated, and $n_{\min} \geq 1$ dynamic tests are included:

$$\mu_{\mathrm{FNE}}\left(n_{\min}\right) = \mu_{\mathrm{FCR}} \cdot \left(1 - FC_{\mathrm{PT}}\right) \tag{38}$$

$$\zeta_{\mathrm{F}} = \frac{k \cdot \mu_{\mathrm{FNE}}\left(n_{\min}\right)}{n_{\min}} \tag{39}$$

μ_{FNE}	expected number of **n**ot **e**liminated **f**aults.
$n_{\min}$	reference test number random test for $FC = 0$.
μ_{FCR}	expected number of **f**aults from **c**reation and repair processes.
FC_{PT}	**f**ault **c**overage of the **p**re **t**ests.
ζ_{F}	malfunction rate caused by **f**aults.

After a total of n random tests

Expected number of faults and MF rate after also elimination of all further detected faults:

$$\mu_{\mathrm{FNE}}(n) = \mu_{\mathrm{FNE}}(n_0) \cdot \left(\frac{n}{n_0}\right)^{-k} \tag{40}$$

$$\zeta_{\mathrm{F}}(n) = \frac{k \cdot \mu_{\mathrm{FNE}}(n)}{n} \tag{41}$$

$$\zeta_{\mathrm{F}}(n) = \zeta(n_0) \cdot \left(\frac{n}{n_0}\right)^{-(k+1)} \tag{42}$$

Form factor:

$$k = \log\left(\frac{\zeta_{\mathrm{F}}(n_0)}{\zeta_{\mathrm{F}}(n_1)}\right) / \log\left(\frac{n_1}{n_0}\right) - 1 \tag{43}$$

μ_{FNE}	expected number of **n**ot **e**liminated **f**aults.
k	form factor of the distribution of the malfunction rate $(0 < k < 1)$.
ζ_{F}	malfunction rate caused by **f**aults.
n_0, n_1	number of tests with known malfunction rate or expected number of faults, respectively.
n	number of tests including n_0 or n_1, respectively.

Fault-related partial reliability

Fault-related partial reliability as reciprocal of MF rate by fault according to eq. 1.43 bzw.1.44:

$$R_{\mathrm{F}}(n) = \frac{n}{k \cdot \mu_{\mathrm{FNE}}(n)} \tag{44}$$

$$R_{\mathrm{F}}(n) = R_{\mathrm{F}}(n_0) \cdot \left(\frac{n}{n_{\min}}\right)^{k+1} \tag{45}$$

R_{F}	**f**aults-related partial **r**eliability.
n_0	number of tests with known MF rate or number of faults, respectively.
n	number of tests including n_0.
μ_{FNE}	expected number of **n**ot **e**liminated **f**aults.
k	form factor of the distribution of the malfunction rate $(0 < k < 1)$.

Example 1.6: Reliability triple test effort

a) By what factor are the MF rate and number of faults reduced when the number of dynamic tests is tripled? Form factors MF rate distribution $k \in \{0.3, 0.5\}$.

b) What gain in reliability can be expected if the staff of the testing department is tripled?

a) Estimated values for MF rate and fault number reduction and the increase in reliability:

$$\frac{\zeta_{\mathrm{F}}\left(3 \cdot n_0\right)}{\zeta_{\mathrm{F}}\left(n_0\right)} = 3^{-(k+1)}; \quad \frac{\mu_{\mathrm{FNE}}\left(3 \cdot n_0\right)}{\mu_{\mathrm{FNE}}\left(n_0\right)} = 3^{-k}; \quad \frac{R_{\mathrm{F}}\left(3 \cdot n_0\right)}{R_{\mathrm{F}}\left(n_0\right)} = 3^{k+1}$$

	$\frac{\mu_{\mathrm{FNE}}(3 \cdot n_0)}{\mu_{\mathrm{FNE}}(n_0)}$	$\frac{\zeta_{\mathrm{F}}(3 \cdot n_0)}{\zeta_{\mathrm{F}}(n_0)}$	$\frac{R_{\mathrm{F}}(3 \cdot n_0)}{R_{\mathrm{F}}(n_0)}$
$k = 0.3$	0.72	0.24	4.17
$k = 0.5$	0.56	0.19	5.19

b) The 3-fold staffing for testing and debugging increases the reliability at product release by 4 to 5-fold.

Reliability and safety in operation

To the estimated MF rate due to faults, add the MF rate due to disturbances:

$$R = \frac{1}{\zeta_{\mathrm{F}} + \zeta_{\mathrm{D}}} \tag{46}$$

According to eq. 1.33, the MF treatment increases the reliability to:

$$R_{\mathrm{MT}} = \frac{1}{(\zeta_{\mathrm{F}} + \zeta_{\mathrm{D}}) \cdot (1 - MC)} \tag{47}$$

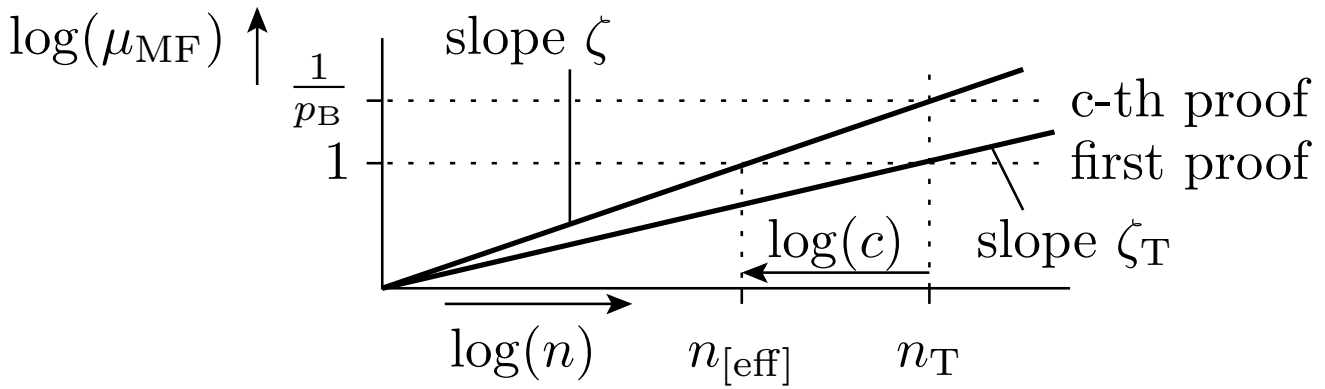
The safety in eq. 1.16 is only compromised by the proportion η_{SE} of unrecognised MF:

$$S = \frac{1}{(\zeta_{\mathrm{F}} + \zeta_{\mathrm{D}}) \cdot (1 - MC) \cdot \eta_{\mathrm{SE}}} \tag{48}$$

ζ_{F}	malfunction rate caused by **f**aults.
ζ_{D}	malfunction rate due to **d**isturbance.
R_{MT}	**r**eliability with **m**alfunction **t**reatment.
MC	**m**alfunction **c**overage, percentage of detected malfunctions.
η_{SE}	percentage of **s**afety **e**ndagering MF.

Effective number of tests

The effective number of tests $n_{[\mathrm{eff}]}$ is the equivalent test number for which all detectable defects are eliminated.



Differences between the rate of MF during the test for which the causative faults are removed and the MF rate in the field can be compensated for by test length scaling:

$$n_{[\mathrm{eff}]} = c \cdot n_{\mathrm{T}} \quad \text{mit } c = \frac{\zeta}{\zeta_{\mathrm{T}}} \tag{49}$$

μ_{MF}	expected number of malfunctions.
$n_{[\mathrm{eff}]}$	**eff**ective number of dynamic tests for which all detected faults are eliminated.
c	test number enlargement.
ζ	malfunction rate during operation.
ζ_{T}	rate of MF during **t**est whose causing faults are eliminated.

- If faults during the test have half the MF rate as in operation, then the same fault coverage is achieved in operation with half the number of tests:

$$c = \frac{\zeta}{\zeta_{\mathrm{T}}} = 2; \quad n_{[\mathrm{eff}]} = 2 \cdot n_{\mathrm{T}}$$

- Elimination of detected faults only with probability $p_{\mathrm{FEs}} < 1$, e.g. due to an error culture without elimination success control:

$$c = p_{\mathrm{FE}}; \quad n_{[\mathrm{eff}]} = p_{\mathrm{FE}} \cdot n_{\mathrm{T}}$$

- Deviating MF rate of modeled faults ζ_{MF} from ζ of actual faults:

$$c = \frac{\zeta}{\zeta_{\mathrm{M}}}; \quad n_{[\mathrm{eff}]} = c \cdot n_{\mathrm{T}}$$

The number of tests n will further be the effective number of tests.

$n_{[\mathrm{eff}]}$	**eff**ective number of dynamic tests for which all detected faults are eliminated.
c	test number enlargement.
ζ	malfunction rate during operation.
ζ_{T}	rate of MF during **t**est whose causing faults are eliminated.
ζ_{M}	Malfunction rate due to **m**odelled faults during test.
p_{FE}	**p**robability of **f**ault **e**limination.

Maturing process

The problem of ever larger IT systems

The expected number of faults grows proportionally with the size of the system or the effort required to create it, see later eq.

$$\mu_{\mathrm{FCP}} = \xi \cdot C \tag{1.73}$$

and the reliability in use decreases inversely proportional to the expected number of faults from the development or production processes:

$$R_{\mathrm{F}} \sim \frac{n^{k+1}}{\mu_{\mathrm{FCP}}}$$

μ_{FCP}	expected number of **f**aults from **c**reation **p**rocess.
ξ	fault generation rate **c**reation process.
C	metric for **c**reation effort or scale.
R_{F}	**f**aults-related partial **r**eliability.
n	number of tests.
k	form factor of the distribution of the malfunction rate $(0 < k < 1)$.

$$R_{\mathrm{F}} \sim \frac{n^{k+1}}{\xi \cdot C}$$

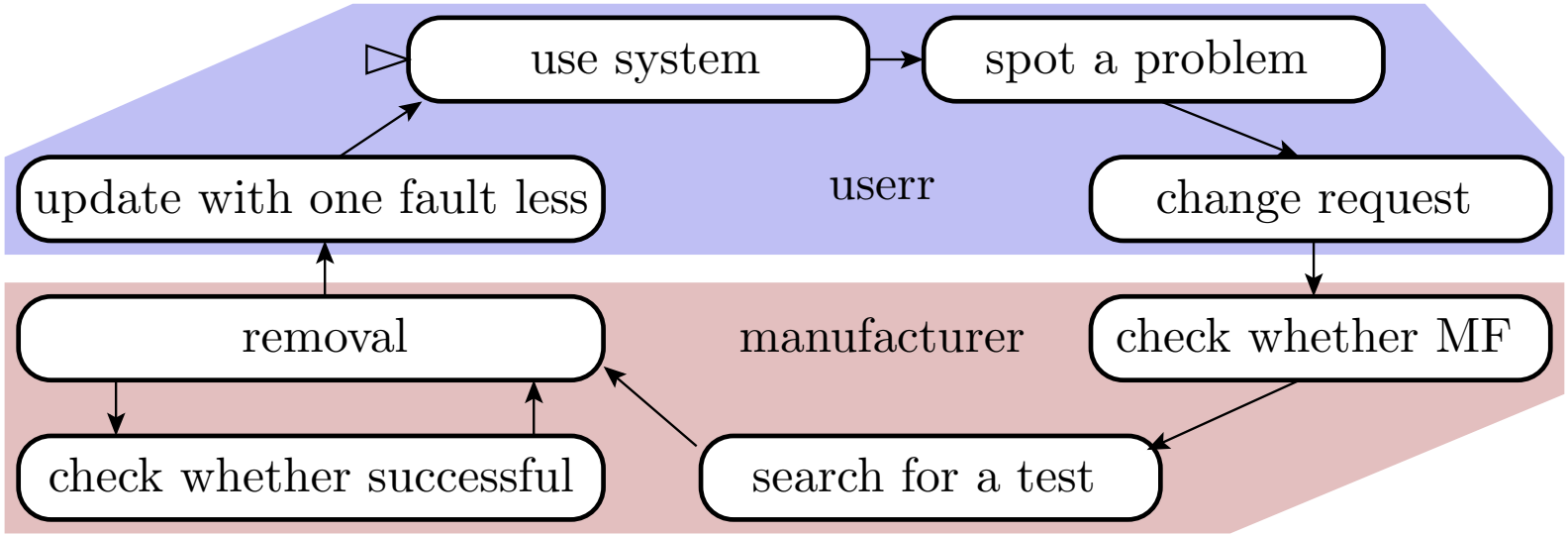
Compensating for the loss of reliability due to the increasing system size requires an increase in the number of tests:

$$n_1 = n_0 \cdot \left(\frac{C_1}{C_0}\right)^{\frac{1}{k+1}} \tag{50}$$

R_{F}	**f**aults-related partial **r**eliability.
n	number of tests.
k	form factor of the distribution of the malfunction rate $(0 < k < 1)$.
ξ	fault generation rate **c**reation process.
n_1	required number of tests for inreased creation effort C.
n_0	sufficient number of tests for creation effort or size C_0.
C_1	metric for the increased **c**reation effort or scale.
C_0	**c**reation effort or scale, for which n_0 tests are sufficient .

Maturing process

The alternative to ever longer testing times before deployment is to install a maturing process with users as testers.



- Capture the MFs during system use.
- Collecting the data to reconstruct the MF.
- Submission to the manufacturer.
- Search for tests for reproducible fault detection.
- Fault elimination by experimental repair.
- Issuing and installing updates without the eliminated faults.

Fault elimination probability

1. In the event of a suspected malfunction, the user submits a change request. Alternatively, the system sends a MF report. MF reports are collected in drawers of suspected same cause.
2. The manufacturer favours drawers in the removal process that suggest faults with frequent serious MF.
3. Search for tests that stimulate the MFs in a reproducible way. The tests are used for fault localisation and to check the elimination success.
4. Experimental repair. Installation of updates.

On average, a fault i is not removed until it has caused many MF:

$$p_{\mathrm{FE}.i} \ll \zeta_i$$

and has thousands or millions of users over the years.

$p_{\mathrm{FE}.i}$	**f**ault **e**limination **p**robability of fault i.
ζ_i	MF rate of MF class i.

Effective number of tests

$$n_{\mathrm{M}} = p_{\mathrm{FE}} \cdot \mu_{\mathrm{NU}} \cdot \eta_{\mathrm{ST}} \cdot t_{\mathrm{M}} + n_{\mathrm{MR}} \gg n_{\mathrm{MR}} \tag{51}$$

Strictly speaking, the expected number of faults does not decrease continuously with the maturation period, but discretely with the version releases. Increase of the effective test set length with the number of versions at release intervals of the same length:

$$n_{\mathrm{M}} = \underbrace{p_{\mathrm{FE}} \cdot \mu_{\mathrm{NU}} \cdot \eta_{\mathrm{SU}} \cdot t_{\mathrm{VR}}}_{n_{\mathrm{MV}}} \cdot u + n_{\mathrm{MR}} \tag{52}$$

n_{M}	effective number of services, for which all detected faults are eliminated.
p_{FE}	**p**robability of **f**ault **e**limination.
μ_{NU}	expected **n**umber of **u**ser.
η_{SU}	mean number of **s**ervices per **u**ser and use time.
t_{M}	**m**aturing **t**ime.
n_{MR}	Effective number of tests before the first and each subsequent version release.
t_{VR}	**V**ersion **r**elease interval.
n_{MV}	additional effective number of tests per **v**ersion **r**elease interval.
u	version number of the maturing object.

Decrease in the number of faults and the MF rate

In the remainder of this section, only the case will be considered:

- No emergence of new faults or complete elimination of newly emerged faults before version release.
- No malfunctions due to disturbances:

$$\mu_{\mathrm{FNE}}\left(n_{\mathrm{M}}\right) = \mu_{\mathrm{FNE}}\left(n_{\mathrm{M0}}\right) \cdot \left(\frac{n_{\mathrm{M}}}{n_{\mathrm{M0}}}\right)^{-k} \tag{53}$$

$$\zeta_{\mathrm{F}}\left(n_{\mathrm{M}}\right) = \frac{k \cdot \mu_{\mathrm{FNE}}(n_{\mathrm{M}})}{n_{\mathrm{M}}} \tag{54}$$

$$\zeta_{\mathrm{F}}\left(n_{\mathrm{M}}\right) = \zeta_{\mathrm{F}}\left(n_{\mathrm{M0}}\right) \cdot \left(\frac{n_{\mathrm{M}}}{n_{\mathrm{M0}}}\right)^{-(k+1)} \tag{55}$$

μ_{FNE}	expected number of **n**ot **e**liminated **f**aults.
n_{M}	effective number of services, for which all detected faults are eliminated.
n_{MR}	Effective number of tests before the first and each subsequent version release.
ζ_{F}	malfunction rate caused by **f**aults.
k	form factor of the distribution of the malfunction rate $(0 < k < 1)$.

Increase in reliability and safety

Reliability as reciprocal of MF rate with according to slide before for $\zeta_{\mathrm{D}} = 0$:

$$R\left(n_{\mathrm{M}}\right) = \frac{n_{\mathrm{M}}}{k \cdot \mu_{\mathrm{FNE}}(n_{\mathrm{M}})} \tag{56}$$

$$R\left(n_{\mathrm{M}}\right) = R\left(n_{\mathrm{M0}}\right) \cdot \left(\frac{n_{\mathrm{M}}}{n_{\mathrm{M0}}}\right)^{k+1} \tag{57}$$

Reliability with MF treatment according to eq. 1.33 reciprocal of the rate of undetectable MF:

$$R_{\mathrm{MT}}\left(n_{\mathrm{M}}\right) = \frac{R(n_{\mathrm{M}})}{1-MC} = \frac{R(n_{\mathrm{M0}})}{1-MC} \cdot \left(\frac{n_{\mathrm{M}}}{n_{\mathrm{M0}}}\right)^{k+1} \tag{58}$$

According to eq. EqSREtaSE, the safety is still greater by the reciprocal value of the proportion of safety-endangering MF:

$$S\left(n_{\mathrm{M}}\right) = \frac{R(n_{\mathrm{M}})}{(1-MC) \cdot \eta_{\mathrm{SE}}} \tag{59}$$

μ_{FNE}	expected number of **n**ot **e**liminated **f**aults.
n_{M}	effective number of services, for which all detected faults are eliminated.
n_{MR}	Effective number of tests before the first and each subsequent version release.

$$R_{\mathrm{MT}}\left(n_{\mathrm{M}}\right) = \frac{R(n_{\mathrm{M0}})}{1-MC} \cdot \left(\frac{n_{\mathrm{M}}}{n_{\mathrm{M0}}}\right)^{k+1} \tag{1.62}$$

High reliability required:

- high reliability $R\left(n_{\mathrm{M0}}\right)$ at product release,
- high MC of the fault function handling and a
- high effective number of tests

$$n_{\mathrm{M}} = p_{\mathrm{FE}} \cdot \mu_{\mathrm{NU}} \cdot \eta_{\mathrm{ST}} \cdot t_{\mathrm{M}} + n_{\mathrm{MR}} \tag{1.55}$$

  - high probability p_{FE} that when a MF is observed, the causative fault is removed,
  - a large expected number of users μ_{NU},
  - many used services per user and time η_{ST} and
  - a long maturing time t_{M}.

Systems that have matured for many years have high reliabilities that are unattainable by other means. Difficult to replace with new systems (see Y2K problem).

New / alternative systems are often much less reliable in the first years of use than the systems they replace. If this affects acceptance, they do not mature either ...

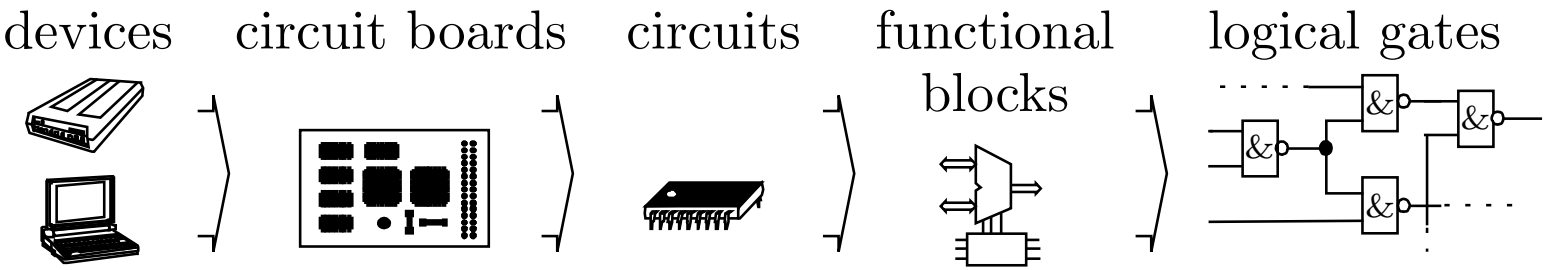
Malfunction avoidance – user learning processes

When familiarising oneself with a new IT system, it is typical that MFs occur frequently at the beginning and less and less frequently with increasing use, because the user learns to work around the faults and weak points in the system. Here, too, an increase in reliability can be observed with the time of use.

When knowledge about workarounds is shared, e.g. via forums, FAQ pages, the entire user community learns. Summation of the usage periods t_{M} of many users.

Modular test

IT systems are modular



- Computer systems consist of computers, IO devices, printers, network components, ...
- Computers and accessories consist of hardware and software.
- Software consists of programme modules, which are composed of instructions that are in turn reproduced by machine commands.
- Machine commands are services provided by the hardware. The hardware consists of function modules, these mostly of gates and these in turn of transistors.
- Higher-level systems inherit the functions and faults of their subsystems.

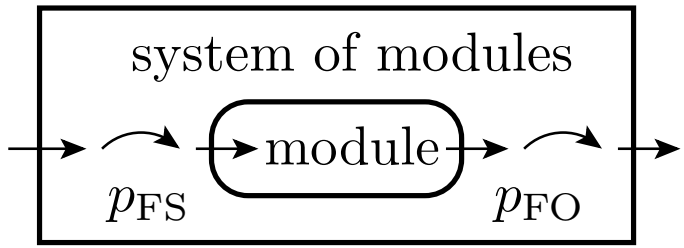
Modularity is important for ...

- Design process: splitting into subtasks, reuse of partial designs, ...
- Test: Thorough test of the components before insertion into the higher-level system.
- Repair: Replaceable components.
- Increase the effective test set length for component-internal faults.

Effective number of tests, test organisation

MF rate of the system during operation due to intra-module faults:

$$\zeta_{\mathrm{Sys}} = p_{\mathrm{FS}} \cdot p_{\mathrm{FO}} \cdot \zeta_{\mathrm{Mod}}$$



ncrease of the effective number of tests:

$$n_{[\mathrm{eff}]} = c \cdot n_{\mathrm{T}} \quad \text{mit } c = \frac{\zeta_{\mathrm{Mod}}}{\zeta_{\mathrm{Sys}}} = \frac{1}{p_{\mathrm{FE}} \cdot p_{\mathrm{FO}}} \gg 1$$

IIn a reasonable testing technology, modules are thoroughly tested before installation in higher-level systems and the tests of the higher-level system mainly check connections between modules.

$\zeta_{[\mathrm{Sys}]}$	paritial malfunction rate of the overall system due to faults within the module.
p_{FS}	**p**robability of **f**ault **s**timulation.
p_{FO}	**p**robability of **f**ault **o**bservation.
ζ_{Mod}	MF rate of the **mod**ule during isolated test.
c	test number enlargement.
$n_{[\mathrm{eff}]}$	**eff**ective number of dynamic tests for which all detected faults are eliminated.
n_{T}	**n**umber of **t**ests.

Yield, defect level

Defect level

For non-repairable systems and replaceable components, the number of faults is not of interest, but only whether a fault is contained.

Defect level:

$$DL = \left.\frac{\#DP}{\#P}\right|_{\mathrm{ACR}} \tag{60}$$

Units of measurement dpu (defects per unit), dpm (defects per million):

$$1\,\mathrm{dpu} = 10^6\,\mathrm{dpm}$$

For expected number of faults $\mu_\mathrm{F} \ll 1$ (almost never more than one fault per product):

$$DL = \mu_\mathrm{F}$$

DL	**d**efect **l**evel.
$\#P$	number of **p**roducts.
$\#DP$	number of **d**efective **p**roducts thereof.
ACR	**a**ppropriate **c**ounting **r**anges.
μ_F	expected number of **f**aults.

Defect coverage and Yield

The defect coverage is the percentage of detected defective products:

$$DC = \left.\frac{\#IDP}{\#DP}\right|_{\mathrm{ACR}} \tag{61}$$

The yield is the percentage of products found to be good

$$Y = 1 - \left.\frac{\#IDP}{\#P}\right|_{\mathrm{ACR}} \tag{62}$$

and depends on the defect coverage of the test used to sort out the defective parts:

$$Y = 1 - DL_{\mathrm{M}} \cdot DC \tag{63}$$

DC	**d**efect **c**overage, percentage of detectable defective devices.
$\#IDP$	number of **i**dentifiably defective **p**roducts.
$\#DP$	number of **d**efective **p**roducts thereof.
Y	**y**ield.
$\#P$	number of **p**roducts.
DL_{M}	**d**efect **l**evel after **m**anufacturing.

$$Y = 1 - DL_{\mathrm{M}} \cdot DC \tag{1.67}$$

Without tests is the defect coverage $DC = 0$ and the yield $Y = 1$.

Example 1.7: Yield and defect level

Yield $Y = 95\%$, estimated with a test that detects $DC = 50\%$ of defective products. Defect level?

Transformation of eq. 1.67 according to the defect level:

$$DL = \frac{1 - Y}{DC} = \frac{0{,}95\%}{50\%} = 10\%$$

Y	**y**ield.
DL	**d**efect **l**evel.
DC	**d**efect **c**overage, percentage of detectable defective devices.

Defect level after sorting out

When sorting out the detected defective objects, the number of defective objects in the numerator and denominator is both reduced by the number of detected defective objects $\#P \cdot DL_{\mathrm{M}} \cdot DC$:

$$DL_{\mathrm{R}} = \frac{\#P \cdot DL_{\mathrm{M}} - \#P \cdot DL_{\mathrm{M}} \cdot DC}{\#P - \#P \cdot DL_{\mathrm{M}} \cdot DC} = \frac{DL_{\mathrm{M}} \cdot (1 - DC)}{1 - DL_{\mathrm{M}} \cdot DC} \tag{64}$$

In case of replacement, the number of products $\#P$ is to be chosen larger by the number of sorted out products.

Yield instead of error share after production:

$$Y = 1 - DL_{\mathrm{M}} \cdot DC \tag{1.67}$$

$$DL_{\mathrm{R}} = \frac{(1-Y)\cdot(1-DC)}{DC \cdot Y} \tag{65}$$

DL_{R}	**d**efect **l**evel after **r**eplacement of detected defective parts.
$\#P$	number of **p**roducts.
DL_{M}	**d**efect **l**evel after **m**anufacturing.
DC	**d**efect **c**overage, percentage of detectable defective devices.
Y	**y**ield.

Example 1.8: Defect level of tested circuits

Circuit yield $Y = 80\%$, defect level after test and sorting out of detected defective ICs $DL_{\mathrm{R}} = 1000\,\mathrm{dpm}$. What is the DC?

$$DL_{\mathrm{R}} = \frac{(1-Y)\cdot(1-DC)}{DC\cdot Y} = 1000\,\mathrm{dpm}$$

$$DC = \frac{1-Y}{DL_{\mathrm{R}}\cdot Y+1-Y} = \frac{1-80\%}{10^{-3}\cdot 80\%+1-80\%} = 99.6\%$$

For tested ICs one finds in the literature $DL_{\mathrm{R}} = 200\,\mathrm{dpm}$ up to $1000\,\mathrm{dpm}$, for the stuck-at fault coverages of the test sets only $FC_{\mathrm{SA}} = 95\%$ to 99%. The proportion of undetectable stuck-at faults is obviously a power of ten greater than the proportion of undetectable defective ICs (see sec. 5.1.4 *Detection relations*):

- Is DC much higher than the stuck-at fault coverage or
- is the information for DL_{R} of the tested ICs is much too optimistic?

DC	**d**efect **c**overage, percentage of detectable defective devices.
DL_{R}	**d**efect **l**evel after **r**eplacement of detected defective parts.
Y	**y**ield.

Systems of pre-tested sybsystems

System consisting of many tested parts, each with a small defect level $DL_i \ll 1$. The system test only checks for connection faults, but detects almost no defective parts. Why?
Expected number of faults in the overall system:

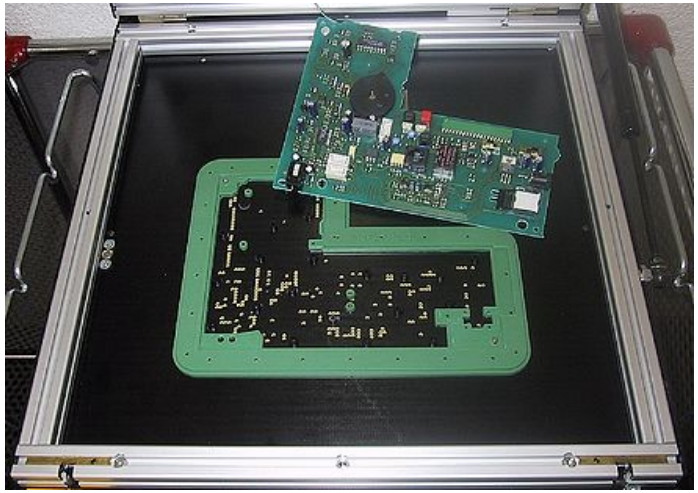
$$\mu_{\mathrm{FSys}} = \mu_{\mathrm{FCon}} \cdot (1 - FC_{\mathrm{Con}}) + \sum_{i=1}^{\#Prt} DL_i \tag{66}$$

For $\mu_{\mathrm{FSys}} \ll 1$:

$$DL_{\mathrm{Sys}} = \mu_{\mathrm{FSys}} \tag{67}$$

μ_{FSys}	expected number of **f**aults in the overall **sys**tem.
μ_{FCon}	expected number of **con**nection **f**aults.
FC_{Con}	fault coverange for **con**nection faults.
$\#Prt$	number of **p**ar**t**s.
DL_i	**d**efect **l**evel of component i.
DL_{Sys}	**d**efect **l**evel of the **sys**tem.

Circuit board



Electronic circuit boards consist of tested components and are usually clamped on a bed of nails for testing. Target defects: wire breaks, short circuits and assembly errors.

Fault coverage for shorts, breaks and misplaced components is practically $FC_{\mathrm{Con}} = 1$ and no coverage for defective components:

$$\mu_{\mathrm{FSys}} = \sum_{i=1}^{\#Prt} DL_i \tag{68}$$

For $\mu_{\mathrm{FSys}} \ll 1$ the overall system has the defect level:

$$DL_{\mathrm{Sys}} = \mu_{\mathrm{FSys}} \tag{1.71}$$

μ_{FSys}	expected number of **f**aults in the overall **sys**tem.
$\#Prt$	number of **p**a**rt**s.
DL_i	**d**efect **l**evel of component i.

Example 1 (Defect level of a circuit board)

Number and defect levels for all component types:

Typ	number	DL_i
circuit board	1	20 dpm
integrated circuits	20	200 dpm
discrete components	35	10 dpm
solder joints	560	1 dpm

$$DL_{\text{Sys}} = \mathbb{E}\left[\#F_{\text{Sys}}\right] = 10\,\text{dpm} + 20 \cdot 200\,\text{dpm} + 35 \cdot 10\,\text{dpm} + 560 \cdot 1\,\text{dpm}$$
$$= 5000\,\text{dpm} = 0{,}005\,\text{dpu}$$

About every 200th device contains an undetected defective part. Computer hardware can contain defective circuits, but only those that very rarely cause MF.

DL_{Sys}	Fehleranteil des Systems (**d**efect **l**evel of the **sys**tem).
DL_i	Fehleranteil von Bauteil i (**d**efect **l**evel of component i).
dpm	fehlerhafte Objekte pro eine millionen Stück (**d**efecs **p**er **m**illion).

Summary

4.1 Elimination iteration, 4.2 Fault diagnosis

Fault elimination: Iteration of removal attempts for hypothetical faults and success control by repeating the test:

- Elimination of all detectable faults.
- Undoing after unsuccessful repair attempts.
- In modular systems by systematic swapping.

Fault diagnosis: Estimation of location, cause and elimination options for faults from test results:

- Pareto principle: As a rule, only a small percentage of possible causes is responsible for the majority of faults.
- Traceability against the calculation or signal flow.

Design for repair:

- exchangeable modules, deterministic behaviour,
- directed computation flow, fault isolation, ...

With a reasonable repair technology, the percentage of defects removed is almost as large as the defect coverage of the tests.

4.3 Test

Classification of test methods:

- Static tests: direct control of characteristics.
- Dynamic tests: probing the system function.

Characteristics:

- Fault coverage, proportion of detectable faults:

$$FC = \left.\frac{\#DF}{\#F}\right|_{\mathrm{ACR}} \tag{1.34}$$

- Phantom MF rate during the test, proportion of test outputs that are detected as incorrect but are in fact correct:

$$\zeta_{\mathrm{PhanT}} = \left.\frac{\#PM}{n}\right|_{\mathrm{ACR}} \tag{1.35}$$

Before troubleshooting, "test the tests" for phantom faults!

Selection strategies for dynamic tests:

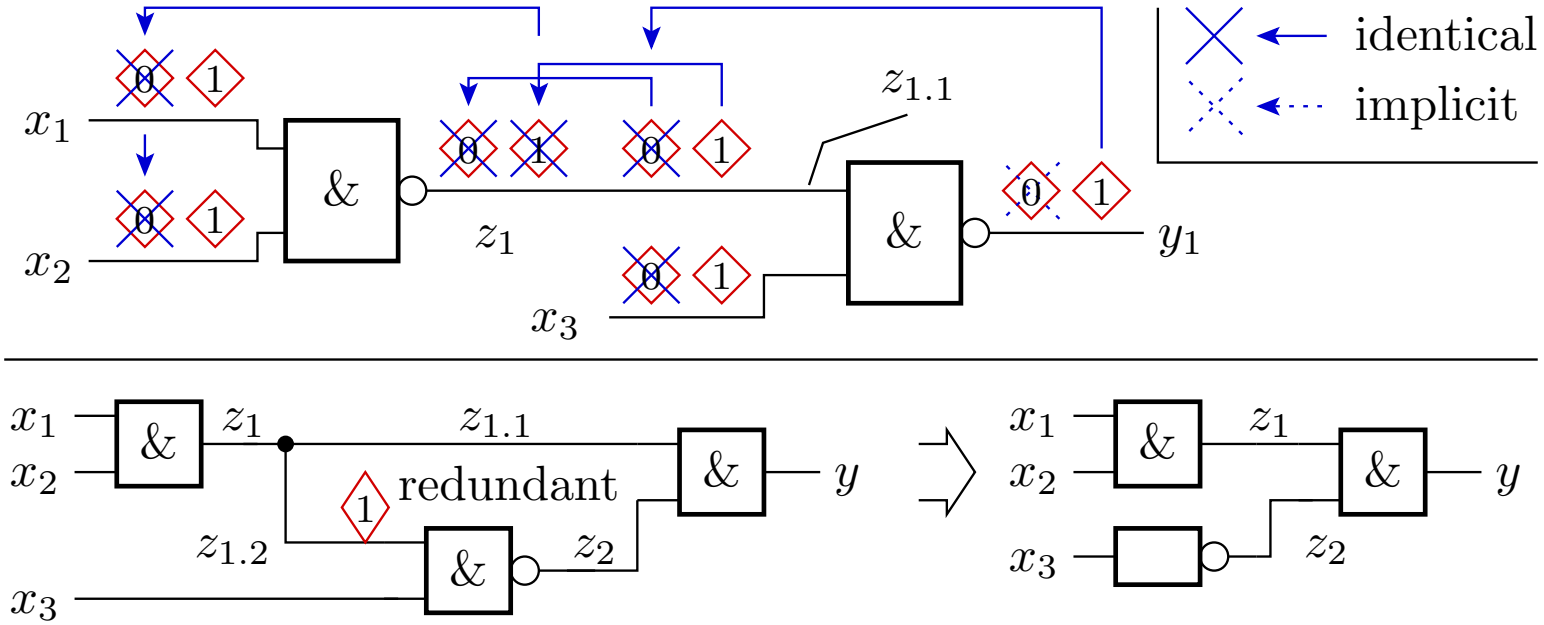
- fault-oriented: defining a set of model faults or mutations. Search inputs proving those faults.
- random: selection without regard to fault assumptions.
- Mixed forms.

Even with fault-oriented selection, the detection of actual faults is a matter of chance.

Fault model: Algorithm for calculating a set of possible falsifications from a design description.

Model fault: a single one of these postulated falsifications.

4.4 Stuck-at faults



Example of a fault model:

- Initial fault set: sa0 and sa1 for each gate connection.
- Grouping of identically detectable faults, deletion of redundant and implicitly detectable faults.
- The resulting fault set is used for fault simulation and test calculation.

4.5 Reliability after fault elimination

If all faults detected with n dynamic tests are eliminated, then the MF rate per uneliminated fault is on average $\leq 1/n$. MF rate due to fault not greater than:

$$\zeta_{\mathrm{F}} \leq \frac{\mu_{\mathrm{FNE}}}{n}$$

Typical decrease of the proportion of undetectable faults with a randomly selected number of tests:

$$1 - \mathbb{E}\left[FC\left(n\right)\right] = \left(\frac{n}{n_{\min}}\right)^{-k} \tag{1.36}$$

From this, we can infer the distribution and density of the MF rate and from this, in turn, the decrease in the number of faults and the MF rates with the number of random tests.

Before the random tests, static tests are carried out (e.g. reviews and syntax tests), followed by a few dynamic tests to see whether anything works and then, as a rule, some fault-oriented tests, e.g. for limit values.

The pre-tests find a total of FC_{PT} with a total of n_{PT} dynamic tests. After elimination, the expected number of faults and the MF rate are:

$$\mu_{\mathrm{FNE}}\left(n_{\mathrm{min}}\right) = \mu_{\mathrm{FCR}} \cdot \left(1 - FC_{\mathrm{PT}}\right) \tag{1.40}$$

$$\zeta_{\mathrm{F}} = \frac{k \cdot \mu_{\mathrm{FNE}}(n_{\mathrm{min}})}{n_{\mathrm{min}}} \tag{1.41}$$

Expected number of faults and the MF rate after elimination of the faults detectable with the subsequent random tests:

$$\mu_{\mathrm{FNE}}\left(n\right) = \mu_{\mathrm{FNE}}\left(n_0\right) \cdot \left(\frac{n}{n_0}\right)^{-k} \tag{1.42}$$

$$\zeta_{\mathrm{F}}\left(n\right) = \frac{k \cdot \mu_{\mathrm{FNE}}(n)}{n} \tag{1.43}$$

$$\zeta_{\mathrm{F}}\left(n\right) = \zeta_{\mathrm{F}}\left(n_0\right) \cdot \left(\frac{n}{n_0}\right)^{-(k+1)} \tag{1.44}$$

with:

$$\mu_{\mathrm{FNE}}\left(n_{\mathrm{PT}}\right) = \mu_{\mathrm{FPT}}$$

$$\zeta_{\mathrm{F}}\left(n_{\mathrm{PT}}\right) = \zeta_{\mathrm{PT}}$$

The form factor can be estimated from experimentally determined MF rates for two different test lengths:

$$k = \ln\left(\frac{\zeta_{\mathrm{F}}(n_0)}{\zeta_{\mathrm{F}}(n_1)}\right) / \ln\left(\frac{n_1}{n_0}\right) - 1 \tag{1.45}$$

Fault-related partial reliability:

$$R_{\mathrm{F}}(n) = \frac{n}{k \cdot \mu_{\mathrm{FNE}}(n)} \tag{1.46}$$

$$R_{\mathrm{F}}(n) = R_{\mathrm{F}}(n_0) \cdot \left(\frac{n}{n_0}\right)^{k+1} \tag{1.47}$$

Reliability and safety in operation:

$$R = \frac{1}{\zeta_{\mathrm{F}} + \zeta_{\mathrm{D}}} \tag{1.48}$$

$$R_{\mathrm{MT}} = \frac{1}{(\zeta_{\mathrm{F}} + \zeta_{\mathrm{D}}) \cdot (1 - MC)} \tag{1.49}$$

$$S = \frac{1}{(\zeta_{\mathrm{F}} + \zeta_{\mathrm{D}}) \cdot (1 - MC) \cdot \eta_{\mathrm{SE}}} \tag{1.50}$$

The test number n in all these equations is the effective test set length, i.e. the number of tests for which all detected faults are eliminated:

$$n_{[\mathrm{eff}]} = c \cdot n_{\mathrm{T}} \quad \text{mit } c = \frac{\zeta}{\zeta_{\mathrm{T}}} \tag{1.51a}$$

and can deviate from n_{T} by a factor $c \neq 1$.

1.4.6 Maturing process

The operational reliability decreases inversely proportional to the size of the system or the effort required to create it, respectively:

- Required increase in the number of tests to compensate for the loss of reliability:

$$n = n_0 \cdot \left(\frac{C}{C_0}\right)^{\frac{1}{k+1}} \tag{1.54}$$

Continued troubleshooting under operation with users as testers:

- effective number of tests:

$$n_{\mathrm{M}} = p_{\mathrm{FE}} \cdot \mu_{\mathrm{NU}} \cdot \eta_{\mathrm{ST}} \cdot t_{\mathrm{M}} + n_{\mathrm{MR}} \tag{1.55}$$

- effective number of tests per update interval:

$$n_{\mathrm{M}} = n_{\mathrm{MV}} \cdot u + n_{\mathrm{MR}} \tag{1.56}$$

- decrease in the number of faults and MF rate if no new faults emerge when detected faults are eliminated:

$$\mu_{\mathrm{FNE}}\left(n_{\mathrm{M}}\right) = \mu_{\mathrm{FNE}}\left(n_{\mathrm{M0}}\right) \cdot \left(\frac{n_{\mathrm{M}}}{n_{\mathrm{M0}}}\right)^{-k} \tag{1.57}$$

$$\zeta_{\mathrm{F}}\left(n_{\mathrm{M}}\right) = \frac{k \cdot \mu_{\mathrm{FNE}}(n_{\mathrm{M}})}{n_{\mathrm{M}}} \tag{1.58}$$

$$\zeta_{\mathrm{F}}\left(n_{\mathrm{M}}\right) = \zeta_{\mathrm{F}}\left(n_{\mathrm{M0}}\right) \cdot \left(\frac{n_{\mathrm{M}}}{n_{\mathrm{M0}}}\right)^{-(k+1)}$$

(1.59)

- Reliability of mature systems neglecting the MF rate due to disturbances:

$$R\left(n_{\mathrm{M}}\right) = \frac{n_{\mathrm{M}}}{k \cdot \mu_{\mathrm{FNE}}(n_{\mathrm{M}})} \tag{1.60}$$

$$R\left(n_{\mathrm{M}}\right) = R\left(n_{\mathrm{M0}}\right) \cdot \left(\frac{n_{\mathrm{M}}}{n_{\mathrm{M0}}}\right)^{k+1} \tag{1.61}$$

- Reliability with MF treatment and safety in use:

$$R_{\mathrm{MT}}\left(n_{\mathrm{M}}\right) = \frac{R(n_{\mathrm{M0}})}{1-MC} \cdot \left(\frac{n_{\mathrm{M}}}{n_{\mathrm{M0}}}\right)^{k+1} \tag{1.62}$$

$$S\left(n_{\mathrm{M}}\right) = \frac{R(n_{\mathrm{M}})}{(1-MC) \cdot \eta_{\mathrm{SE}}} \tag{1.63}$$

- Long maturing processes over years and decades achieve unattainable reliability in other ways.
- Old, long-matured software is difficult to replace because equivalent replacements must also mature for a long time with many users.
- There are also maturation processes for user behaviour through which reliability increases with individual usage time

1.4.7 Modular test

Modularity is important for:

- Design process: splitting into subtasks, re-use of partial designs, ...
- Test: Thorough test of the components before insertion into the higher-level system.
- Repair: Replaceable components.
- Increase the effective test set length for component-internal faults by:

$$c = \frac{\zeta_{\mathrm{Mod}}}{\zeta_{\mathrm{Sys}}} = \frac{1}{p_{\mathrm{FE}} \cdot p_{\mathrm{FO}}} \gg 1$$

1.4.8 Defect level, yield

For non-repairable systems and replaceable components, instead of the expected number of faults, the defect level is of interest:

$$DL = \left.\frac{\#DP}{\#P}\right|_{\mathrm{ACR}} \tag{1.64}$$

Defect coverage as a proportion of detectable defective products:

$$DC = \left.\frac{\#IDP}{\#DP}\right|_{\mathrm{ACR}} \tag{1.65}$$

Yield: Proportion of products found to be good.

$$Y = 1 - \left.\frac{\#IDP}{\#P}\right|_{\mathrm{ACR}} \tag{1.66}$$

$$Y = 1 - DL_{\mathrm{M}} \cdot DC \tag{1.67}$$

Defect level after replacement of the detected defective parts:

$$DL_{\mathrm{R}} = \frac{DL_{\mathrm{M}} \cdot (1 - DC)}{1 - DL_{\mathrm{M}} \cdot DC} \tag{1.68}$$

$$DL_{\mathrm{R}} = \frac{(1 - Y) \cdot (1 - DC)}{DC \cdot Y} \tag{1.69}$$

Number of faults in a system of tested components:

$$\mu_{\mathrm{FSys}} = \mu_{\mathrm{FCon}} \cdot (1 - FC_{\mathrm{Con}}) + \sum_{i=1}^{\#Prt} DL_i \tag{1.70}$$

For $\mu_{\mathrm{FSys}} \ll 1$:

$$DL_{\mathrm{Sys}} = \mu_{\mathrm{FSys}} \tag{1.71}$$

For tested PCBs, $FC_{\mathrm{Con}} = 1$ usually applies. The defect count and, for small values, also the defect level are approximately the sum of the defect levels of all components:

$$\mu_{\mathrm{FSys}} = \sum_{i=1}^{\#Prt} DL_i \tag{1.72}$$

Fault prevention

Planned topics

fault prevention	fault elimination	malfunction treatment
elimination of the causes of faults	test and elimination of detected faults	monitoring, robust resposnce MF tolerance interferences

5.1 Fault creation
Modelling of emergence processes as a service provider and fault as its MF.

5.2 Determinism and randomness
Fault prevention is a maturing process for an emergence process. However, manual steps in particular lack determinism. How do non-deterministic emergence processes mature?

5.3 Projects, process models
Maturing processes require a large number of repetitions of the same process in order to be able to learn from recognised mistakes. Projects are unique processes. Process models standardise the procedure in order to still be able to learn from mistakes.

Fault emergence

Faults as MF of the creation process

A creation process is also a service

- with design specifications or material (properties) as input
- and design results or products (or their properties) as output.

It thus also inherits the parameters describing reliability:

- availability, MF rate as fault generation rate,
- reliability, safety, ...

and the measures to ensure reliability.

MF avoidance is a maturing process for a creation service:

- Monitoring of the resulting designs or products.
- Elimination of identified causes of fault emergence.

NS	no service.
CS	correct service.
MF	malfunction.

Fault generation rates and metrics

The expected number of emerging faults is, if the process conditions do not change, the product of a fault emergence rate and a parameter for the emergence effort:

$$\mu_{\mathrm{FCP}} = \xi \cdot C \tag{69}$$

With generation expense proportional to system size, C can also be a metric for system size. Estimation of a fault generation rate tailored to a metric:

$$\xi = \frac{\mu_{\mathrm{FCP}}}{C} \tag{70}$$

Examples:

- Documentations: Faults per page,
- programme code: Faults per 1000 NLOC (Net Lines of Code),
- Circuits: Faults per 10^6 transistors, ...

μ_{FCP}	expected number of **f**aults from **c**reation **p**rocess.
ξ	fault generation rate **c**reation process.
C	metric for **c**reation effort or scale.

Example 1.9: programme faults

$\xi = 30$ faults / 1000 NLOC, programme size $C = 2000$ NLOC. Expected number of programme faults?:

$$\mu_{\mathrm{FCP}} = \xi \cdot C = \frac{30\,\mathrm{Fehler} \cdot 2000\,\mathrm{NLOC}}{1000\,\mathrm{NLOC}} = 60\,\mathrm{Fehler}$$

Example 1.10: faults per circuit

$\xi = 1$ faults per 10^6 transistors. Circuit with $C = 10^5$ transistors. Expected number of faults per circuit?

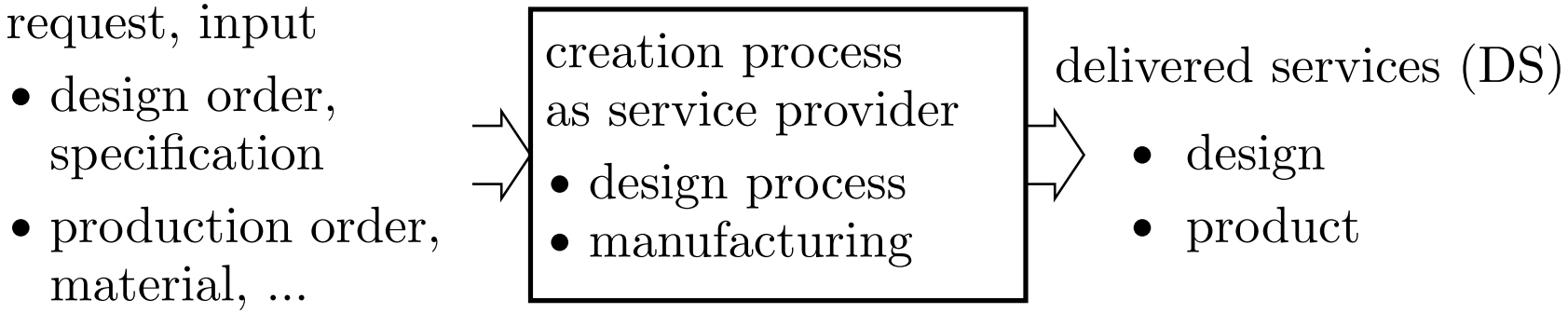
$$\mu_{\mathrm{FCP}} = \xi \cdot C = \frac{1\,\mathrm{Fehler} \cdot 10^5\,\mathrm{Transistoren}}{10^6\,\mathrm{Transistoren}} = 0{,}1\mathrm{Fehler}$$

There are also empirical models that postulate a disproportionate increase in the number of faults with the size of the system. For software modules, for example, it is assumed that the number of faults per NLOC increases disproportionately from 3 source code pages for a function module, because the designers lose track of them.

A reasonably designed creation process avoids all known negative influences on fault generation rates.

Determinism and randomness

Fault creation



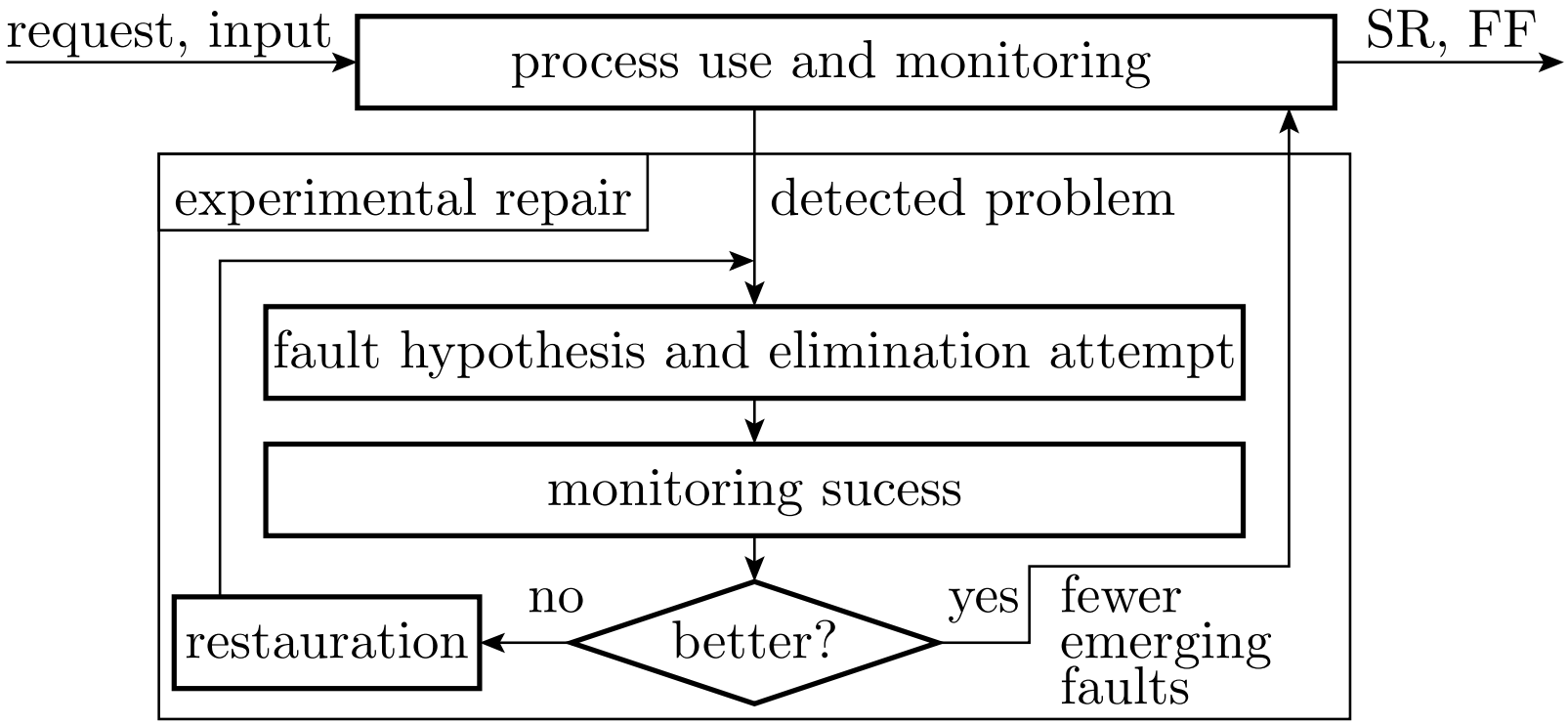
Sources of fault emergence::

- Faults: deterministic cause-effect relationship
  - eliminable causes,
  - success control by single test repetition, ...
- disturbances: random cause-effect relationship
  - MF correction by recalculation,
  - success control after cause elimination difficult„ ...
- Failures: faults occurring in the deployment phase, ...

Fault prevention is achieved by eliminating faults and by reducing the vulnerability to disturbances in the creation processes.

############ hier weiter ######

Fault prevention as a maturing process



Fault prevention is a Maturing process for an creation process with experimental repair for problem elimination. Iteration of:

- Problem detection, localisation, fix attempts,
- success control through rerun of the creation process, and
- deconstruction of changes after unsuccessful elimination attempts.

Experimental repair and determinism

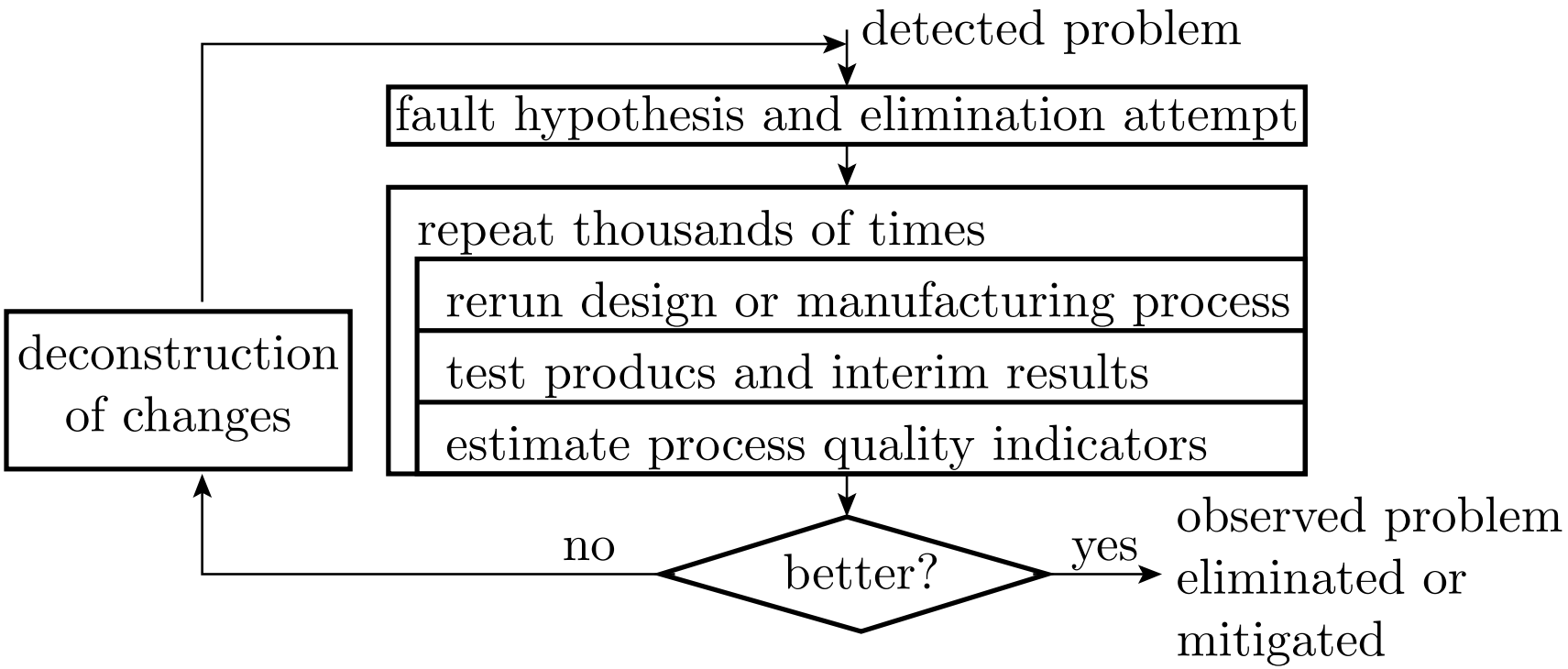
Determinism means that for the same design or manufacturing job (same specification, same material, ...), the fault-free system always produces the same outputs (the same design result, an identical product, ...).

For faults in deterministic processes, it is usually possible to find process sequences that allow checks on intermediate results and end products that provide clear yes/no statements about the presence/elimination of faults.

For non-deterministic processes, faults with non-deterministic effect and process disturbances, the check of successful problem elimination usually requires

- a statistically significant sample of typically 1000 process runs to determine process performance indicators and
- decisions with a certain probability of misinterpretation.

... Non-deterministic processes or fault effects or Disturbances as MF causes:



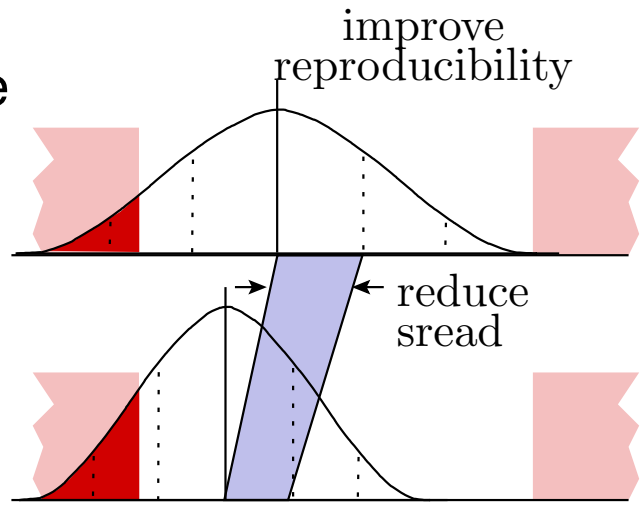
Non-deterministic processes require orders of magnitude more process runs for the same reduction in fault generation rate and have a significantly higher risk of new faults emerging that are not eliminated by deconstruction.

Process centring and improvement

There are causes of faults that are easy and those that are difficult to eliminate, example process centring/improvement.

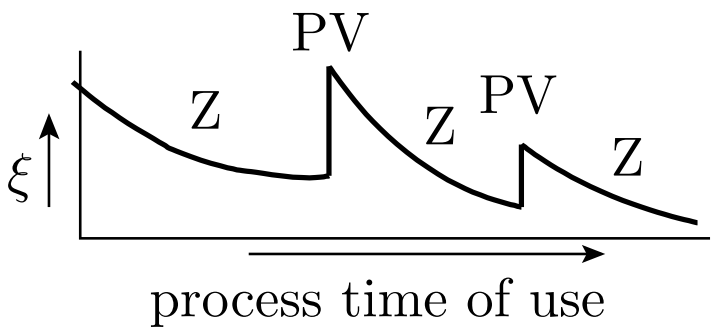
In mechanical production, the target parameters, e.g. diameter and depth of a hole, have a distribution and a tolerance range. The frequency of occurrence of a parameter error is approximately the probability of the parameter being outside the tolerance range:

- Process centring: Shifting the distribution to the centre of the tolerance range using adjustment options.
- Process improvement: Reduction of parameter deviation by technological innovations, new machines, processes, ...



With a technological innovation, the centration gets lost and the fault creation rate increases abruptly.

Sawtooth course of the fault creation rate



Z	process centring
PV	process improvement with lost of centring
ξ	fault generation rate

Technological improvements (new machines, programmes, technologies, ...) occur in larger time periods (months, years) and have the potential to reduce the expected number of errors..

- With each technological reorganisation, the centring is lost and the number of faults increases abruptly.
- The potentially smaller fault creation rate is only achieved by re-centring, i.e. after a longer period of use.
- As process centring progresses, the rate of fault creation decreases.

Also in other manufacturing processes and design processes

- there are technological innovations in larger time steps that reduce the achievable fault creation rate through lower sensitivity to faults, higher reproducibility, ... However, new process faults emerge with innovations, which increase the observable fault creation rate in the products abruptly.
- In between, a continuous search and elimination of fault creation causes, starting with those that cause the most faults. Effect on the process similar to centring.

Conclution

Products tend to have the lowest fault generation rates shortly before next technological innovations (maxima of process reliability).

A dark side of innovation

Technological maturation processes can be observed in every type of service today:

- Improved repeatability of the processes,
- improved / predictable material and product properties,
- less arising faults, increased yield, lower costs.

Downside:

- Innovations almost inevitably lead to new »teething troubles« that are only eliminated after a certain maturing period.
- More emerging faults means not only poorer yield and more costs, but also more faults in deployed systems, more early failures, ...

Linux, for example, differentiates in its version management:

- »Innovative« beta versions with many teething problems, ...
- and reliable stable versions.

Projects, process models

The idea of technology

Technology: The principles of reproducible processes for the creation of products.[4].

The idea of technology

A technological process should be designed in such a way that, when repeated under the same conditions, the same products with (almost) the same properties emerge.

The technological development towards

- automated human-free production and
- computer-aided / automated design processes

dient nicht nur zur Kostensenkungung, sondern ist auch wesentliche Grundlage für die Fehlervermeidung.

[4]The term »technology« was first used by the Göttingen professor Johann Beckmann (1739-1811) in his textbook "Grundsätze der teutschen Landwirthschaft". Today an interdisciplinary field.

Transferring the idea of technology to projects

Technologies mature by repeating the same process very often in order to detect as many faults as possible and to control the success of elimination.

What about projects:

- Manual creative parts of the design processes[5] und
- Production of prototypes, demonstrators, ... ?

A project is a purposeful, one-time undertaking consisting of a set of coordinated, directed activities. ...

From the point of view of fault prevention, projects lack reproducibility and frequent repetition.

Does that exclude projects from fault prevention by learning from faults?

[5] Here in particular the software and hardware design.

Process models

Unification of the workflow for large classes of projects

- for cost minimisation, better predictability and
- fault prevention by »learning from faults«.

Typical process models for the design and manufacture of IT components include:

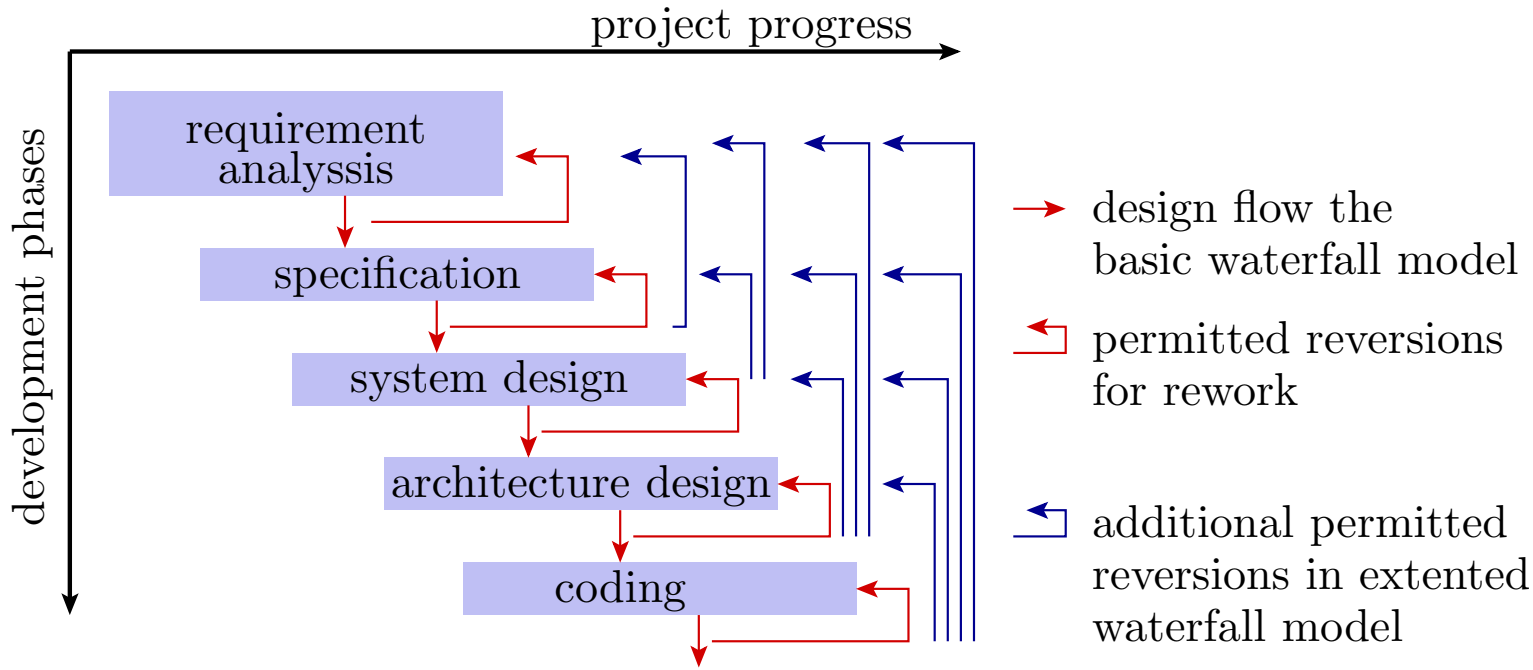
- Splitting into steps and phases,
- reference workflows,
- Definition of intermediate and final controls, ...

The classic process models for software design are stage models. They divide development processes into phases:

- Requirements analysis,
- specification ,
- architecture design, coding, testing, ...

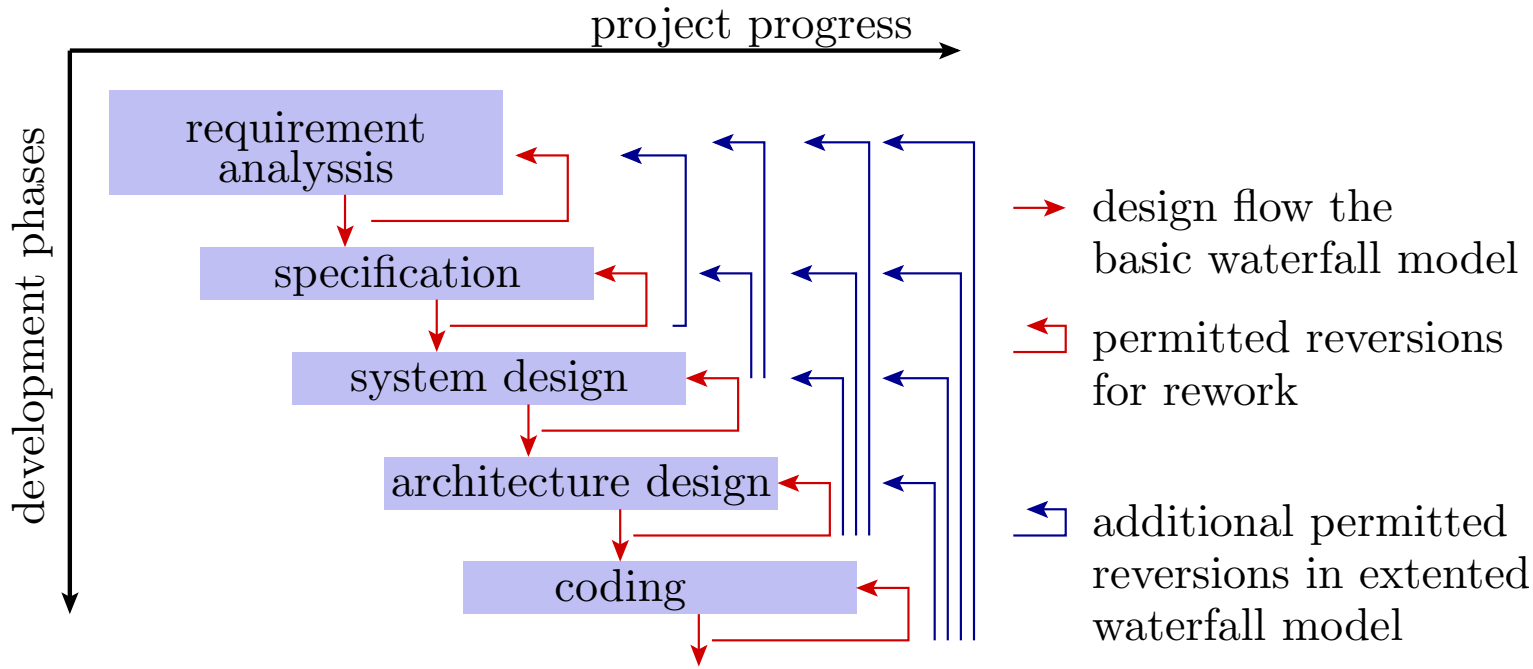
Fault prevention in projects is the empirical search for a good procedural model and its enforcement.

Stage models



Stage models vary:

- in the definition of the design phases,
- Documentation and check for phase transitions,
- in allowing fallbacks (retroactive changes to the results of already completed phases), ...



Adjustable parameters influencing quality and costs:

- workflow within the phases,
- required tests and documentation for phase transitions,
- rules for rework, ...

Fallbacks increase the number of steps in the creation process and with it the number of arising faults. The alternative, a workaround for a fault from a former stage, can also significantly increase the amount of work and thus the number of arising faults. Difficult trade-off.

Evaluation of process models

Each kind of fault prevention requires a check on success:

Resulting question

Which measurable or assessable parameters can be used to recognise an improvement in a process model?

These parameters must be comparable between different real projects and process models:

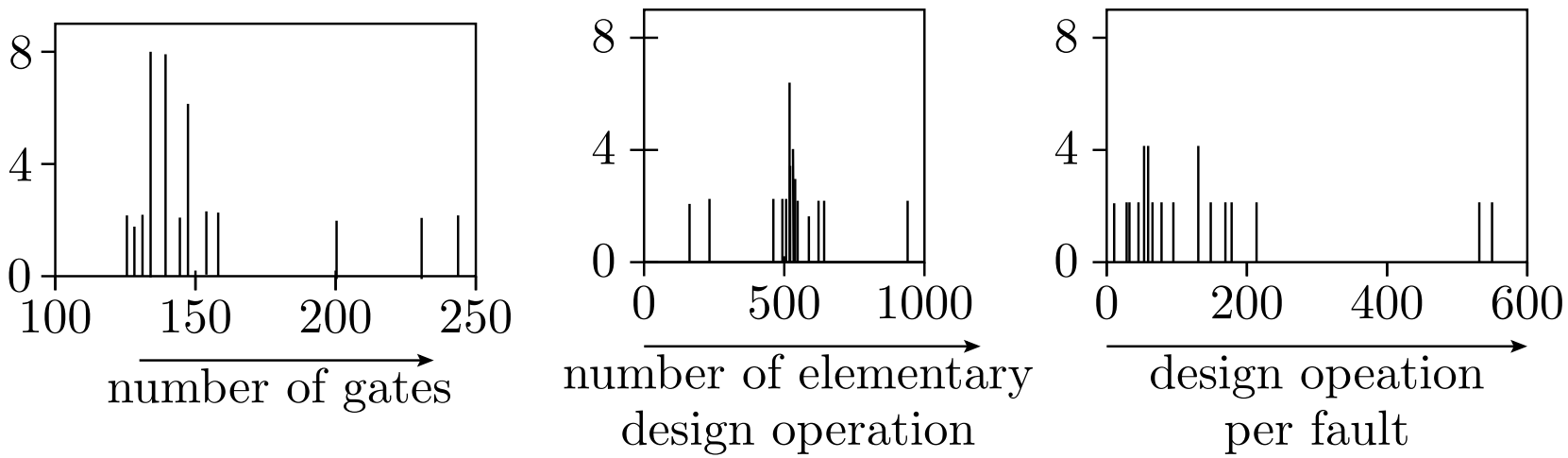
- Duration, costs related to the project size,
- Work steps per emergence of faults, survey results, ...

Expected values, scatter, scalability to project size, difficulty, ...

Significant statements about process models require the observation of thousands of projects with comparable workflows.

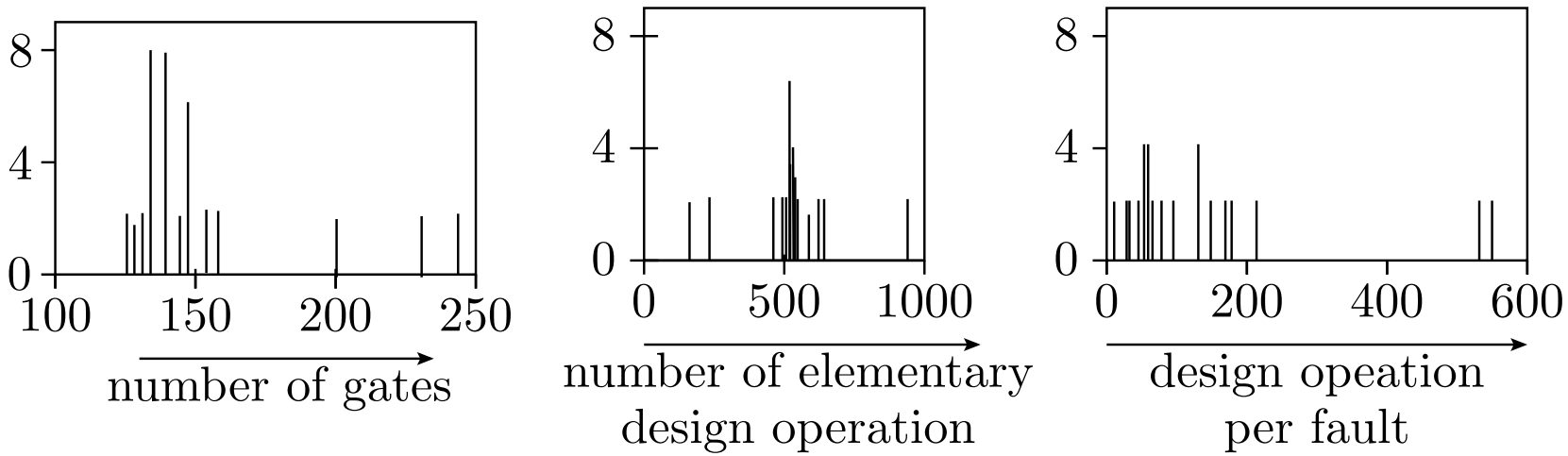
An experiment [6]

A group of 72 students were asked to design a gate circuit from a PLA (Programmable Logic Array) description and enter it into a CAD system via a GUI. For each design, the elementary design operations, the number of gates and the design errors were counted. The elementary design operations were the arrangement of a gate on the screen, the drawing of a connection, ...



[6] Aas, J. E., Sundsbo, I.: Harnessing the Human Factor for Design Quality, IEEE Circuits and Devices Magazine, 3/1995, S. 24-28

What conclusions does the experiment allow?



Assume that the experiment is repeated in exactly the same way at other universities:

- Here, too, determination of the same parameters for each student,
- Comparison of the distributions, expected values and variances.
- Statistically significant differences?

From the findings of the comparison, it could be concluded whether and at which higher education institution students are better trained for this task.

Quality and creativity

Quality and creativity

Quality requires fault prevention. Fault prevention requires:

- a high repetition rate of the same or similar tasks,
- workflows to be followed with reproducible results,
- logging of all irregularities and problems, ...

Creativity requires »uniqueness«:

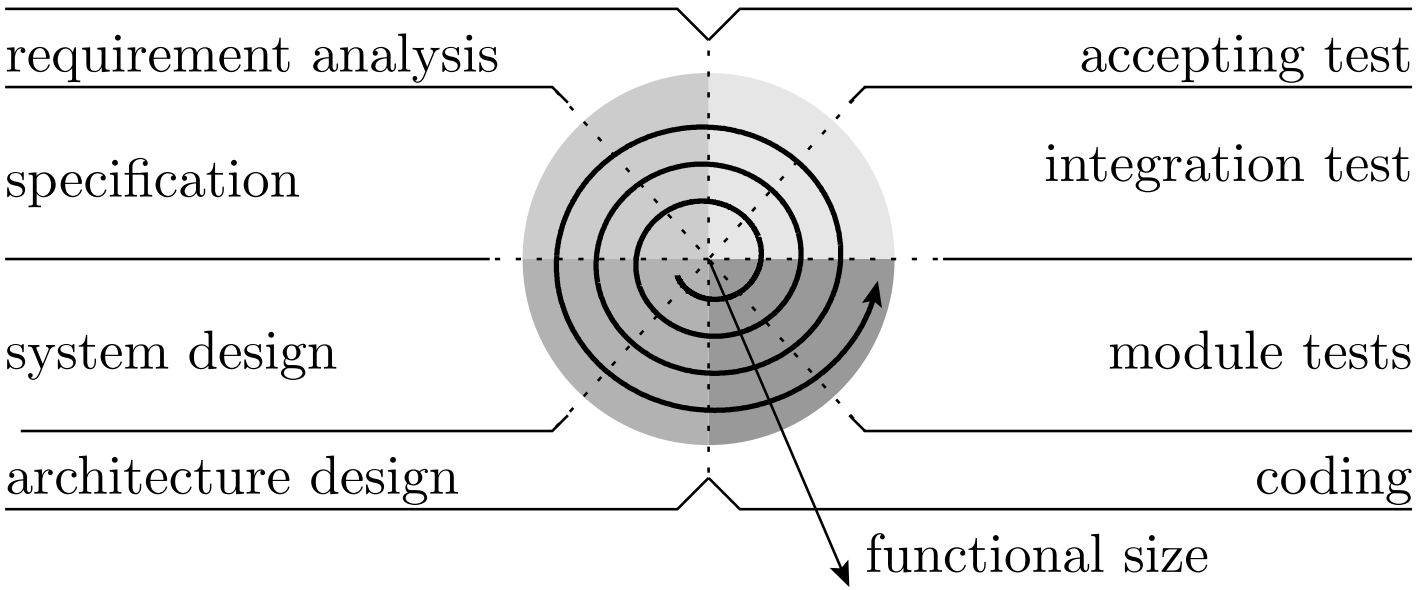
- bringing in new concepts,
- trying out new solutions,
- flexible adaptation to changing requirements.

Conclusion

Quality and creativity have opposite requirements on workflow design. IT design requires quality and creativity. How can both be combined in a process model?

Spiral model as an example of evolutionary models

Evolutionary process models try to provide a framework for projects in which customer requirements, goals, procedures, ... evolve with the project. Less rigid processes. More creative freedom. Example spiral model:



- Breakdown of a development into a repeated run through of a stage model.

Breakdown of a development into a repeated run through of a stage model.

- Run 1: Specification of basic requirements, design, coding, testing, ..., acceptance and deployment.
- Run 2 to n: Idea collection and selection of desired additional requirements and changes. Design to deployment.

Intended goal:

- Minimise the number of creation steps and the number of faults arising per stage model run.
- Creative freedom in the form of collecting ideas for the next stage model run.

Ideally, no changes should be made to already implemented features after each stage model run, except for bug fixes.

Basic idea good, the actual achievable benefit is in the implementation details.

Cross-connections to everyday academic life

Process models are also used for the design of learning processes. The Bolonia process (Bachelor-Master) strives to establish reference processes.

Behind this is the hope that similar spectacular progress as in science and technology can be achieved with the idea of technology in the education system:

- Unification of the workflows.
- Improving predictability and comparability of educational outcomes and costs.
- Adoption of the »processes« from educational institutions with better results from educational institutions with poorer results

How is it at our university:

- Is the organisation of teaching and research processes maturing?
- Which types of creativity are restricted and which are not? ...

Side trip to learning processes

In school and when learning practical activities, largely process models are taught and trained.:

- Arithmetic, writing, handicrafts, programming, ...
- Rating in performed services per MF and time.

Learning phases

1. knowledge transfer: reading, being explained, ...
2. training until results are predictable.
3. professionalisation: process monitoring; elimination of weaknesses and bugs in the processes.

At universities:

- Stage 1: Lecture, seminars, self-study, ...
- Stage 2: Phase 2: Exercise, exam preparation, lab courses.
- Stage 3: For reasons of time, not until professional practice for one's own limited field of activity.

Cross-linking to third-party funded projects

- The professionalisation phase is first undergone by graduates in practice.
- Academics and students are not trained for »low fault-creation rate« workflows.
- In industrial software projects, academics tend to create more faults per code size.
- The industry partner bears the costs for troubleshooting.
- That is why it does not normally pay off for industry to involve universities and students in their daily business.
- Industrielle Studenten-Projekte dienen der Ausbildung.
- Third-party funded research is valuable for know-how transfer, literature studies, demostrators, ... but in the IT sector unsuitable for joint product development.

Fault prevention opens up interesting perspectives on technologies, institutions, authorities, ... and their further development.

Summary

fault prevention	fault elimination	malfunction treatment
elimination of the causes of faults	test and elimination of detected faults	monitoring, robust resposnce MF tolerance interferences

Section 5.1: Fault emergence

Modelling of creation processes as service provider and faults as its MF. Expected number of faults and fault generation rate:

$$\mu_{\mathrm{FCP}} = \xi \cdot C \tag{1.73}$$

$$\xi = \frac{\mu_{\mathrm{FCP}}}{C} \tag{1.74}$$

Section 5.2: Determinism and randomness

If determinism is lacking, the control of success requires statistical studies on thousands of products that have been created. This slows down the maturation processes. Resulting from the usual procedure

- process improvement every few years and
- continuous search for possibilities to reduce the causes of defects

defect generation rate follows a sawtooth pattern with peaks at the time of process changes.

Section 5.3: Projects, process models

Maturing processes require a large number of repetitions of the same procedures. In order to be able to learn from detected faults in projects, project work is carried out according to process models. Classics are the stage models, which divide designs into phases and define checks and activities at the transition from one stage to the next. Checking whether a change leads to an improvement is problematic.

Section 5.4: Quality and creativity

Process models can be found everywhere where constant learning from faults is the goal, i.e. also in administrations, schools, ... In evolutionary process models, creativity is integrated in such a way that new ideas are collected for the specification of subsequent projects, which then ideally run according to a non-recourse stage model.

Fault prevention opens up interesting perspectives on how and where technologies, institutions, authorities and, for example, education at our university are developing.

Kontrolle Seitenreferenzen

(see slide 1.11 *The price of lacking dependability*) on page ??
(see slide 1.96 *Experimental repair*) on page ??
(see slide 1.98 *Backtracing of data falsifications*) on page 104
(see slide 1.128 *Distribution function of the MF rate*) on page ??
(see slide 1.129 *MF rate density, mean MF rate*) on page 131
(see slide 1.130 *Expected number of faults and MF rate*) on page 132
(see slide 1.135 *After a total of n random tests*) on page ??
(see slide 1.159 *Defect level after replacement*) on page ??
(see slide 1.83 *check-point rollback recovery $\eta_{\mathrm{DS}} = 0$*) on page ??