

Test and dependability 1: Problem handling during operation

Prof. G. Kemnitz

Institut für Informatik, TU Clausthal
October 27, 2026

Dependability

- 1.1 Service-Modell
- 1.2 Availability
- 1.3 Reliability
- 1.4 Safety (security)

Problems under use

- 2.1 Monitoring
- 2.2 Format checks
- 2.3 Value checks, diversity
- 2.4 Time monitoring
- 2.5 Correction
- 2.6 Automated/manual

1.2 Foreword

The volume of specialist text grows exponentially over the years, whilst the time available for teaching remains constant. A major challenge for education.

»Test and dependability« summarises the fundamentals of diagnosis, test, maintenance, ... from this point of view in a new way.

Fields of knowledge are structured hierarchically. Knowledge modules can be thought of as stepping stones in an unfamiliar wilderness. Each stone mastered opens up new perspectives for the learner, paving the way to the next one.

Testing and dependability are central to computer science. IT systems are only useful if they can be trusted to

- deliver results on demand (availability),
- ensure that the results are correct (reliability) and
- prevent the system from causing catastrophic damage (security).

By choosing the right stepping stones (definitions, models, etc.), drastically simplifies the presentation of important concepts and, as a result, reduces the time needed to convey that knowledge:

Reliability tends, if all detectable faults are eliminated, to be proportional to the number of tests and inversely proportional to the number of undetected faults.

Where hazards can be ruled out for identified problems, safety is proportional to reliability and can be enhanced either through more testing, fault prevention, improved runtime checks, additional safety equipment or longer maturity periods.

In this conceptual framework, design and manufacturing are processes in which faults arise. Tests act as filters that reveal problems. Based on this, models are developed for qualitative and quantitative assessments. The course covers topics ranging from self-testing and hardware fault tolerance to a vision for solving current reliability and security issues in software.

1.3 Organisation

Lecture web page: <http://techwww.in.tu-clausthal.de/TestVerl>

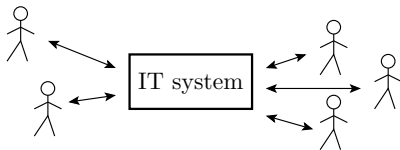
- Slide sets, handouts, homework, video recordings
- Submission of homework by e-mail to ha-tv@in.tu-clausthal.de as PDF. Deadlines see web page.
- Homework will be graded and returned. Additional publication of the number of points on the website.
- Exam admission 50% of the achievable points for all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

1.4 Examination

- Written exam for 10 or more participants.
- Permitted aids for the written examination: own elaboration including handouts with own comments and own homework, calculator.
- Permitted aids for oral examination: One A4 sheet (single-sided) with own elaborations.

For all further information see website.

1.6 Trust and dependability



IT systems automate intellectual tasks:

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

A precondition for use is confidence that

- the system will work when it is needed,
- performs its services correctly and in time,
- does not cause incalculable damage and costs.

Trust in an IT system depends on dependability.

1.6 Dependability

In everyday language, dependability (of people, computers, ...) means that you can trust them. Various aspects come together here (wishes, expectations, ...).

Individual factors influencing the perception of dependability:

- 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:

- for what reliable:
 - Lecturer reliable that on time,
 - Student reliable that exams are handed in, ...
- why reliable:
 - Doctor reliable because completed medical studies,
 - Car reliable because technical licence and valid MOT.

Passed tests are important, but also the error culture ...

1.7 Error-handling culture

The way in which societies, cultures and social systems deal with errors and their consequences.

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

Positive views: Learn from mistakes, eliminate mistakes. ...

- Pedagogy: positive climate for learning from mistakes.
- Quality management: minimising the cost of errors.
- Innovation management: striving for innovation. Errors as Opportunity / productive potential.

The lecture assumes an idealised error-handling culture:

- all identified problems are rectified and
- the rectification will be verified.

In human interactions with friends, superiors and partners and also in cost optimisation of design, production, ... less radical (more tolerant) error cultures are usually better.

*

A simplified model that neglects economic and cultural aspects.

1.10 Hazards and hazard prevention

dependability is described by the absence / prevention / exclusion of hazards, on three levels [2]:

1 Problems during operation:

- crash and other types of denial of service (no service, NS).
- malfunction (MF),

2 Root causes of the problems of (1):

- faults (eliminateable causes for NS and MF),
- disturbances (causes of randomly occurring problems),
- failures (faults occurring during operation).

3 Origins of the causes of problems:

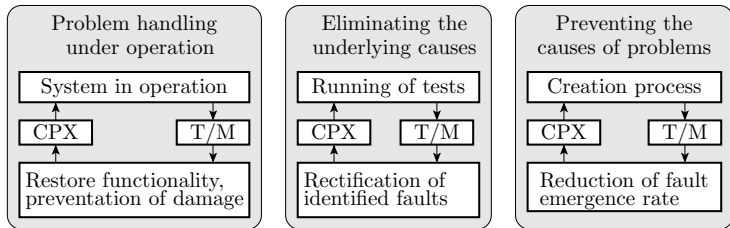
- shortcomings, faults,
- disturbances and failures

in the development and repair processes.

MF, NS

Malfunction. No service.

1.11 Levels of hazard avoidance



Hazard prevention in line with our idealised error culture, through a series of iterations comprising checks, attempts to resolve problems and success checks at three levels:

- Dealing with problems under use
- Testing and fault rectification before deployment.
- Fault prevention by improving creation processes.

T/M, CPX Testing and monitoring for problems. Check of problem fixing.

1.12 What is the cost of dependability?

Costs for hazard prevention:

- Monitoring and adequate response to detected malfunctions: up to more than 50 per cent of overall functionality, e.g. when doubling and comparing results.
- Testing, troubleshooting and debugging: For Hardware and Software typically more than 50% of total design cost.
- Fault prevention by improving the development processes: Costs for quality assurance and the ongoing development and improvement of the development processes.

Overall, measures to ensure dependability can account for well over 50 per cent of product costs.

dependability is the most expensive feature of an IT system.

1.13 The price of insufficient dependability

Since reliability is expensive, why not do without it? – Costs of damage:

- no or limited usability,
- data loss, backdoors for data misuse¹,
- 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. Retaliatory measures were immediately prepared. A review of the data from radar stations and satellites could not confirm the attack ... [1].

Cause of the almost nuclear strike: a defective circuit.

Only reliable IT systems are ready for use.

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

1.14 Hypothetical insurance against damage

Spectacular disaster scenarios raise awareness of problems, but are not suitable for cost-benefit analyses.

The cost of a lack of dependability can best be estimated by a hypothetical insurance policy covering all potential against damage caused by IT using insurance mathematics.

Hypothetical, as no such insurance policies exist and, from a legal perspective, it is difficult to offer due to the historically established distribution of risk:

- Separate business risks for IT manufacturers, operators and users, for each stakeholder, in terms of financial loss, damage to reputation and loss of customers, ...
- product liability in the case of quality assurance shortcomings,
- major damage for which the public has to pay.

Dependability measures are cost-effective if they are cheaper than the reduction in the insurance policy price.

In the case of entertainment IT, for example, the acceptable costs are

1.15 Naming the lecture

Dependability, as the most costly product characteristic of IT systems, is subject to many aspects, of which the **tests** and checks carried out are the most significant.

- **Dependability** is the umbrella term for availability, reliability and safety, and
- all three sub-aspects are ensured at all three levels – namely problem handling, troubleshooting and fault prevention – through checks (monitoring, testing, etc.).

In our idealised error-handling culture, all identified problems are rectified and the rectification is verified:

- only undetected problems cause damage that, hypothetically, would need to be insured against, and
- the frequency of undetected problems depends largely on the effectiveness of the checks and tests.

1.16 Learning objective and content

Learning objective

Basic understanding of what dependability is, of what measures it is influenced by and how. A subject-specific introduction to stochastics, methods of monitoring and testing, and key aspects and specific features of hardware and software.

Foliensätze:

- 1 Measurements. Problem handling during operation.
- 2 Test, Fault rectification. Fault prevention.
- 3 Stochastics I: Probabilities.
- 4 Stochastics II: Estimates and Forecasts.
- 5 More about tests and checks.
- 6 Hardware testing and self-test.
- 7 Software design and testing.



Dependability



1.17 Quantifying Dependability

Usage of IT is based on trust. dependability describes the extent to trust them. Quantifiable through counts of positive and negative experiences across different aspects:

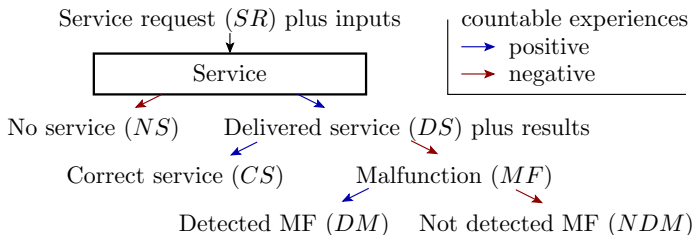
- Delivery: Positive: delivery on demand. Negative: no service.
- Correctness: Positive: correct results. Negative: malfunction.
- Hazards: Positive: no safety risk. Negative: safety risks of varying types and severity.

For a quantitative assessment based on count data IT systems must be modelled in such a way that services delivered and not delivered, correct and incorrect results, and other potential and avoided problems under consideration are countable.



Service-Modell

1.18 Service provider



Basic model for quantifying positive and negative experiences. **System that generates outputs from inputs upon request:**

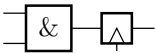
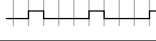
- Positive experience: delivered service (DS),
- Negative experience: no service (NS).

A service provided may be:

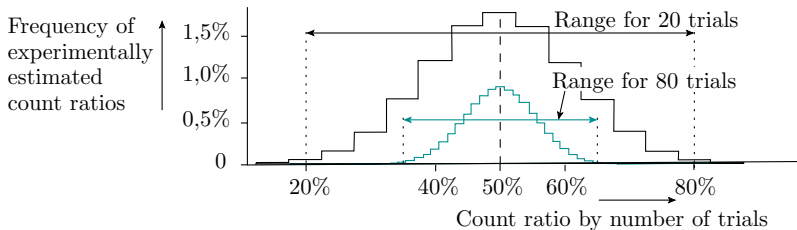
- Positive experience: correct service (CS),
- Negative experience: malfunction (MF).

1.19 Areas of application for the service model

The service model is applicable to different types and abstractions of IT systems, human and automated services, as well as manufacturing and design processes, ...

Clocked digital circuit		I:  O: 
Programme with IPO-structure	<pre>uint8_t up(uint8_t a){ return 23 * a; }</pre>	I: 10 101 ... O: 230 19 ...
Server	I: e.g. database enquiry O: Result dataset	
Manufacturing process	I: Production order, material, ... O: Manufactured product	
Design process	I: Design order O: Design	

1.20 Sufficiently large count values (SLC)



Defining dependency parameters using proportionate count values for positive or negative outcomes allows for a simple and clear assessment of actual or simulated counting experiments:

for 10^5 repetitions:

for $n = 20$ or 80 trials, respectively:

if $\text{rand}() < 50\%$:

increase the count by 1

Result:

n	ε
20	30%
80	15%

Minimum count value	$\varepsilon_{\text{rel}} = 10\%$	$\varepsilon_{\text{rel}} = 5\%$	$\varepsilon_{\text{rel}} = 2\%$
$\alpha = 4\%$	416	1665	10.404
$\alpha = 2\%$	543	2172	13.572
$\alpha = 1\%$	660	2642	16.512

The table shows the minimum count value as a function of the permissible relative interval radius ε_{rel} and the error probability α for independent counts and small occurrence frequencies (sec.4.2.7). The relative interval radius is the ratio of the interval radius to the occurrence frequency. The probability of error is the probability that, if the experiment were repeated, the result would lie outside the estimated interval.

Accurate estimates require large count values and therefore many counting trials. This will be covered in the topic in Slide Set 4. Until then, all parameters to be estimated via counting trials are labelled with $\dots|_{\text{SLC}}$:

Reliable estimates only with sufficient counting trials.

$\varepsilon_{\text{rel}}, \alpha$
SLC

Relative interval radius, error probability.
Useful estimates only with Sufficiently large count values.



Availability

1.22 Availability metrics

Definition of availability as the rate at which requested services are delivered:

$$A = \frac{\#DS}{\#SR} \Big|_{SLC} \quad (1.1)$$

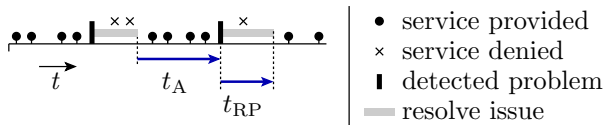
An alternative measure is the probability of failure on demand:

$$PFD = 1 - A \quad (1.2)$$

There are various reasons for unavailability, each of which requires a different reaction. Separate counting for partial availabilities that can be combined to obtain total availability (slide 1.27).

<i>A</i>	Availability.
<i>#SR</i>	Number of service requests.
<i>#DS</i>	Number of delivered services.
<i>PFD</i>	Probability of failure on demand.
<i>SLC</i>	Useful estimates only with Sufficiently large count values.
Rate	Relative frequency of occurrence of the event under consideration.

1.23 Availability, time taken to resolve issues



A system always operates for a period of time t_A without any visible problems (i.e. without denial of service, crashes, detected malfunctions, etc.), followed by a problem-resolution time t_{RP} for repair, reinitialisation, etc. Availability as the average proportion of time:

$$A = \frac{\bar{t}_A}{\bar{t}_A + \bar{t}_{RP}} \quad (1.3)$$

Significant dependence on the average problem resolution time.

A	Availability.
$t_A, \bar{t}_A$	Time and average time during which the system is available between issues.
$t_{RP}, \bar{t}_{RP}$	Time and average time taken to resolve the issue.

1.24 Problem resolution – tasks, effort involved

An IT system should respond appropriately to service denial, system crashes and detected malfunctions:

- Mitigation of damage: data recovery, fallback to a safe state, ...
- Restoring operational readiness: repair, reinitialisation, ...
- Logging and storing data to identify the cause of the problem,
- Termination without service provision or a new request.

In particular, repairs following hardware failures, but also data recovery, ensuring secure conditions, and re-ordering may take a considerable amount of time, during which the system is unavailable.

Availability can be improved by reducing the frequency of problems and by shortening problem-resolution times.

1.25 Partial availabilities

Unavailability can have various causes:

- 1 Hardware failure (FL)
- 2 Denial of acceptance (DA),
- 3 Crash (CR),
- 4 Detected malfunction (DM).

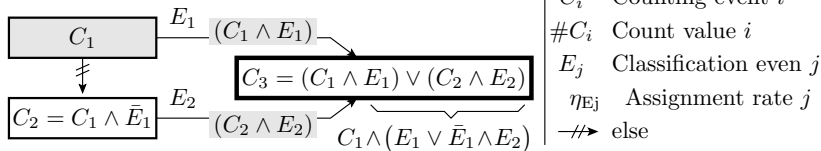
Each has its own count values, its own response measures, and its own occurrence and tolerance rates.

	Failure	DA	Crash	DM
Count of negative events	$\#FL$	$\#DA$	$\#CR$	$\#DM$
Count of positive events	$\#HA$	$\#RA$	$\#DR$	$\#DS$
Occurrence rate	ζ_{FL}	ζ_{DA}	ζ_{CR}	ζ_{DM}
Tolerance rate	ν_{FL}	ν_{DA}	ν_{CR}	ν_{DM}

Introduction of a graphical representation of complex relationships between measured values, enabling a clear visualisation and easy interpretation of measured values and rates.

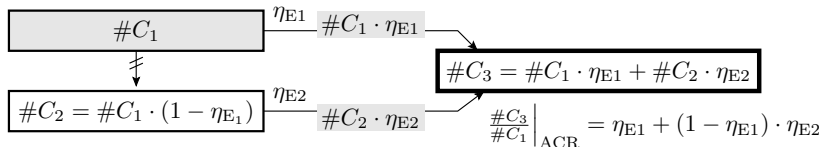
1.26 CVA (count-value assignment) graph

Counting and classification events:



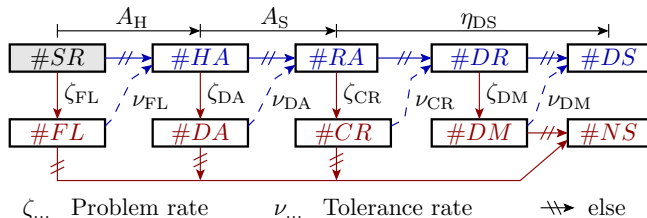
Count value assignment graph:

- all $(\xrightarrow{E_i})$ -Assignments: independent events $\Rightarrow$ Product: $\#_i = \#C_i \cdot \eta_i$
- all $(\hookrightarrow)$ -Reconvergences: mutual exclusion $\Rightarrow$ Sum: $\#C_j = \sum \#_i$



See »Chained Events« below, section 3.1.2.

1.27 Breakdown into three partial availabilities



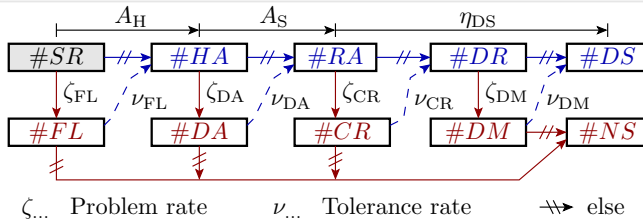
	Failure	DA	Crash	DM
Count of negative events	$\#FL$	$\#DA$	$\#CR$	$\#DM$
Count of positive events	$\#HA$	$\#RA$	$\#DR$	$\#DS$
Occurrence rate	ζ_{FL}	ζ_{DA}	ζ_{CR}	ζ_{DM}
Tolerance rate	ν_{FL}	ν_{DA}	ν_{CR}	ν_{DM}

- $\#HA$ counts requests for which hardware is available,
- $\#RA$ counts requests, for which the service is also available,
- $\#DR$ counts the accepted services provided, ...

Pronounce: ζ : zeta, ν : ny.

1. Dependability

2. Availability



Hardware availability

$$A_H = \left. \frac{\#HA}{\#SR} \right|_{SLC}$$

$$A_H = (1 - \zeta_{FL}) + \zeta_{FL} \cdot \nu_{FL} \quad (1.4)$$

Service availability:

$$A_S = \left. \frac{\#RA}{\#HA} \right|_{SLC}$$

$$A_S = (1 - \zeta_{DA}) + \zeta_{DA} \cdot \nu_{DA} \quad (1.5)$$

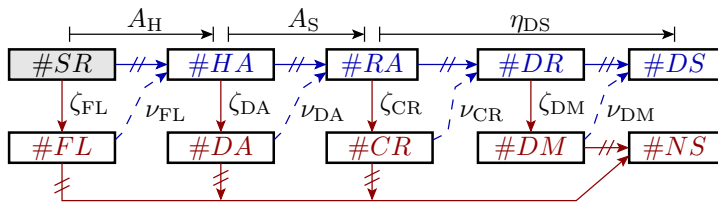
Delivery rate:

$$\eta_{DS} = \left. \frac{\#DS}{\#RA} \right|_{SLC} = \dots$$

Overall availability:

$$A = \left. \frac{\#DS}{\#SR} \right|_{SLC}$$

$$A = A_H \cdot A_S \cdot \eta_{DS} \quad (1.6)$$

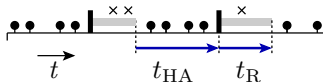


For independent events: failure, failure tolerance, ... along all paths to $\#HA$, then on to $\#RA$, ... only

- independent events are combined using »AND« and
- mutually exclusive events are combined using »OR«.

$\# \langle evt \rangle$	Number of count events:.
SR, HA	Service request. Hardware available.
RA, DR	Service Request accepted. Delivered result.
DS, NS	Delivered Service. No service.
FL, DA	Non-availability due to hardware failure or denial of acceptance.
CR, DM	Unavailability due to crash or recognised malfunction.
ζ_{FL}, ν_{FL}	HW unavailability rate. Tolerance rate for unavailable HW.
ζ_{DA}, ν_{DA}	Service denial rate. Tolerance rate for service denials.
ζ_{CR}, ν_{CR}	Crash rate. Rate of tolerance of crashes.
ζ_{DM}, ν_{DM}	Rate of detected malfunctions. Tolerance rate for detected malfunctions.

1.30 Hardware availability



$$(1.4) \quad A_H = (1 - \zeta_{FL}) + \zeta_{FL} \cdot \nu_{FL}$$

Also, the proportion of time the system is available (eq. 1.3):

$$A_H = \frac{\bar{t}_{HA}}{\bar{t}_{HA} + \bar{t}_R} \stackrel{\bar{t}_{HA} \gg \bar{t}_R}{=} 1 - \frac{\bar{t}_R}{\bar{t}_{HA}} \quad (1.7)$$

High availability requires a low rate of (total) failures, short repair times and, where possible, redundancies that take over the tasks of failed components (sec.6.5.2). Total failure as opposed to minor failures, after which the system remains usable, but with reduced reliability (see section 6.5.2 below).

A_H	Hardware availability.
ζ_{FL}, ν_{FL}	HW unavailability rate. Tolerance rate for unavailable HW.
$\bar{t}_{HA}$	Service life and average service life between total hardware failures.
$t_R, \bar{t}_R$	Repair time. Average repair time.

Total failure: Failure after which a component is no longer usable and must be repaired.

Minor failure: Failure-induced fault that reduces only reliability.

1.31 Service availability

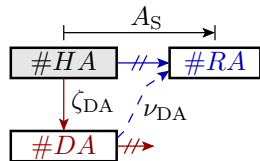
$$(1.5) \quad A_S = (1 - \zeta_{DA}) + \zeta_{DA} \cdot \nu_{DA}$$

Service refusal when hardware is available, e.g. due to:

- Troubleshooting following a crash or a detected malfunction,
- Exceeding the permitted processing time,
- Overload due to too many service requests, ...

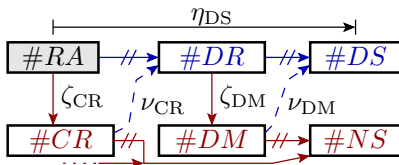
Can be avoided by ensuring sufficient capacity. Can be tolerated by allowing for delays in processing, redistributing tasks, etc.

Closely intertwined with functional aspects. Further on, it is usually neglected ($A_S = 1$).



A_S	Service availability.
$\#HA$	Number of requests to available hardware.
$\#RA$	Number of accepted service requests.
$\#DA$	Number of denials of acceptance.
ζ_{DA}, ν_{DA}	Service denial rate. Tolerance rate for service denials.

1.32 Delivery of accepted requirements



Service delivery rate for accepted requirements

$$\eta_{DS} = \left. \frac{\#DS}{\#RA} \right|_{SLC}$$

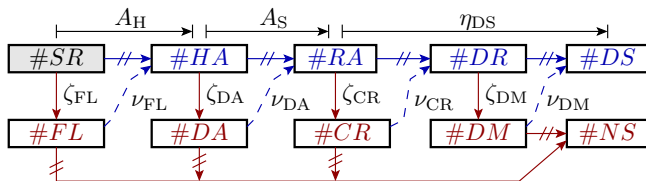
depends on

- the crash rate ζ_{CR} , the rate of detected malfunctions ζ_{DM} ,
- the associated tolerance rates ν_{CR} and ν_{DM} , and
- how detected malfunctions are handled.

Continued in section 1.2: Troubleshooting.

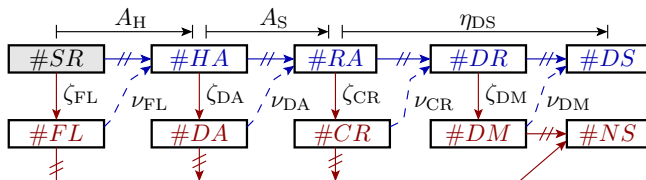
<i>RA, DR</i>	Service Request accepted. Delivered result.
<i>DS, NS</i>	Delivered Service. No service.
<i>CR, DM</i>	Unavailability due to crash or recognised malfunction.

Example 1.1: CVA graph and availability

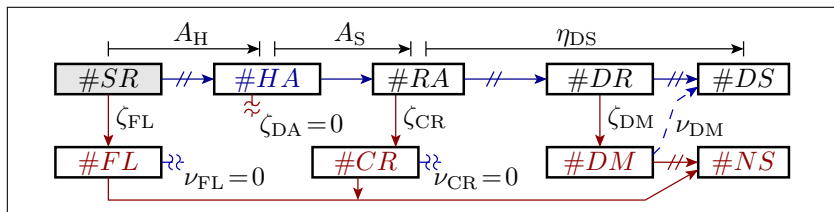


- Adjust the CVA graph so that failures are never tolerated, service requests are always accepted when hardware is available, and no service is provided in the event of a crash.
- What are the formulas for hardware availability, service availability, service delivery rate and overall availability, as shown in the CVA graph from part (a) of the question?

SR, HA	Service request. Hardware available.
RA, DR	Service Request accepted. Delivered result.
DS, NS	Delivered Service. No service.
FL, DA	Non-availability due to hardware failure or denial of acceptance.
CR, DM	Unavailability due to crash or recognised malfunction.



- a) *Adjust the CVA graph so that failures are never tolerated, service requests are always accepted when hardware is available, and no service is provided in the event of a crash.*

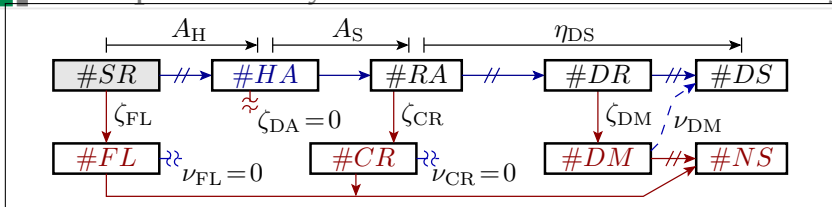


- ζ_{FL}, ν_{FL} HW unavailability rate. Tolerance rate for unavailable HW.
 ζ_{DA}, ν_{DA} Service denial rate. Tolerance rate for service denials.
 ζ_{CR}, ν_{CR} Crash rate. Rate of tolerance of crashes.



1. Dependability

2. Availability



- b) What are the formulas for hardware availability, service availability, service delivery rate and overall availability, as shown in the CVA graph from part (a) of the question?

$$A_H = (1 - \zeta_{FL}) ; A_S = 1$$

$$\eta_{DS} = (1 - \zeta_{CR}) \cdot (1 - \zeta_{DM} + \zeta_{DM} \cdot \nu_{DM})$$

$$A = (1 - \zeta_{FL}) \cdot (1 - \zeta_{CR}) \cdot (1 - \zeta_{DM} \cdot (1 - \nu_{DM}))$$

ζ_{DM}, ν_{DM} Rate of detected malfunctions. Tolerance rate for detected malfunctions.
 A_H, A_S Hardware availability. Service availability.
 η_{DS}, A Rate of delivered services, Overall availability.



Reliability

1.34 A look back

The use of IT requires trust. Dependability describes the extent to which this trust is justified. An objective description based on metrics for positive and negative experiences. Classification by aspects:

Service delivery:

- Positive experiences: Service provided upon request.
- Negative experiences: Refusal of service.
- Performance indicators: Availability, which can be broken down into the product of partial availabilities.

Accuracy:

- Positive experiences: the results produced were correct.
- Negative experiences: the results produced were incorrect.
- Key figures: [<continue here>](#)

Damage:

- Positive experiences: no damage when requesting a service.
- Negative experiences: damage of various types and degrees.
- Key figures: (sec. 1.1.4)

1.35 Key figures for accuracy

Reliability*: Number of delivered services (DS) per delivered malfunction (DMF):

$$R = \frac{\#DS}{\#DMF} \Big|_{SLC} \quad (1.8)$$

Malfunction rate: the reciprocal of reliability:

$$\zeta = \frac{1}{R} = \frac{\#DMF}{\#DS} \Big|_{SLC} \quad (1.9)$$

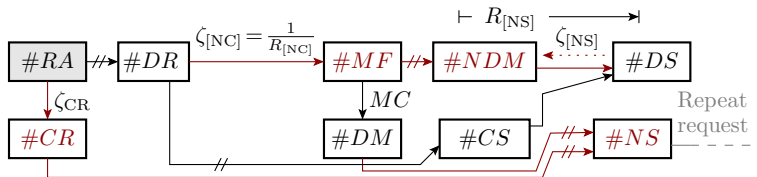
Probability of reliability as the counter-probability of the Malfunction rate:

$$p_R = 1 - \zeta$$

The preferred reliability measure is R , because in this case a tenth of the failures equates to ten times the reliability $10 \cdot R$, rather than an increase in the probability of reliability from 99% to 99.9%.

R, ζ	Reliability. Malfunction rate.
$\#DS$	Number of delivered services.
$\#DMF$	Number of malfunctions delivered.
SLC	Useful estimates only with Sufficiently large count values.
*	Purposeful but still uncommon definitions in the professional world.

1.36 Revision of the CVA graph



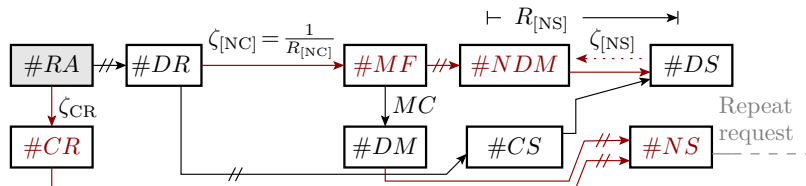
Adding count values for undetected faults: Malfunctions (MF) are detected with frequency MC erkannt (DM) and otherwise remain undetected (NDM).

In the example CVA graph, no service delivery (NS) occurs in the event of a crash (CR) or a detected malfunction (DM)..

Later, this also applies with at most one re-request [SR] and re-requests until service delivery [RR] in the case of a problem (sec.1.2.5).

$\# \langle evt \rangle$	Number of count events:.
RA, DR	Service Request accepted. Delivered result.
MF, NDM	Malfunction. Undetectable malfunction.
DS, NS	Delivered Service. No service.

1.37 Delivery rate



Delivery rate if no service provision in the event of problems

$$\eta_{DS} = (1 - \zeta_{CR}) \cdot (1 - \zeta_{[NS]} \cdot MC) \quad (1.10)$$

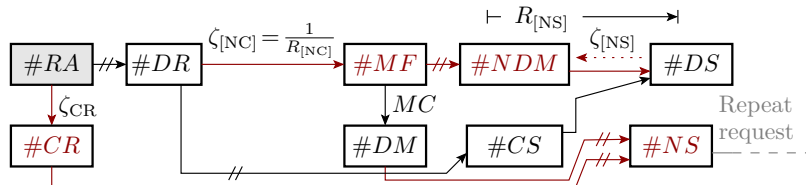
With an observable malfunction rate:

$$\zeta_{OMF} = \zeta_{[NC]} \cdot MC \quad (1.11)$$

$$\eta_{DS} = (1 - \zeta_{CR}) \cdot (1 - \zeta_{OMF}) \quad (1.12)$$

$\zeta_{[NC/NS]}$	malfunction rate without checks / with no service for identified problems.
$R_{[NC/NS]}$	Reliability without checks / with no service for identified problems.
ζ_{CR}, MC	Crash rate. Malfunction coverage.
η_{DS}	Rate of delivered services.
ζ_{OMF}	Rate of observable malfunctions.

1.38 Improvement to reliability



Malfunction rate with monitoring and troubleshooting:

$$\zeta_{DMF} = \frac{1}{R} = \frac{\#NDM}{\#DS} \Big|_{SLC} = \frac{\cancel{\#DR} \cdot (1 - \zeta \cdot MC)}{\cancel{\#DR} \cdot \zeta \cdot (1 - MC)}$$

$$\zeta_{DMF} = \frac{\zeta \cdot (1 - MC)}{1 - \zeta_{OMF}} \quad (1.13)$$

Improvement to reliability:

$$R_{[NS]} = \frac{R_{[NC]} \cdot (1 - \zeta_{OMF})}{(1 - MC)} \quad (1.14)$$

ζ_{OMF}	Rate of observable malfunctions.
$R_{[NC/NS]}$	Reliability without checks / with no service for identified problems.
MC, MT	Malfunction coverage. Malfunction leakage.



Where the observable failure rate is generally very low ($\zeta_{\text{OMF}} \ll 1$), monitoring and the refusal to perform a service in the event of detected problems improve reliability in inverse proportion to the failure detection rate of the checks carried out during operation:

$$R_{[\text{NS}]} = \frac{R_{[\text{NC}]}}{1 - MC} = \frac{R_{[\text{NC}]}}{MT} \quad (1.15)$$

The reduction of the delivery rate due to refusal to perform

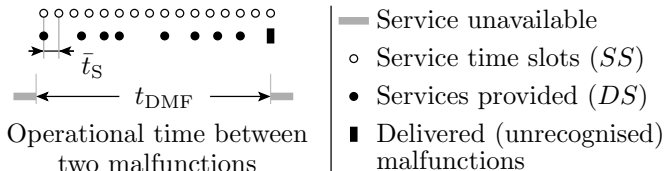
$$(1.12) \quad \eta_{\text{DS}} = (1 - \zeta_{\text{CR}}) \cdot (1 - \zeta_{\text{OMF}})$$

is generally negligible and can be remedied by repeated service requests.

Confirmation of the argument presented on slide 1.15, namely that reliability is determined primarily by the quality of the checks carried out.

ζ_{OMF}	Rate of observable malfunctions.
$R_{[\text{NC/NS}]}$	Reliability without checks / with no service for identified problems.
MC, MT	Malfunction coverage. Malfunction leakage.
η_{DS}	Rate of delivered services.
ζ_{OMF}	Rate of observable malfunctions.
ζ_{CR}	Crash rate.

1.40 Estimation from time-based averages



$$(1.8) \quad R = \frac{\#DS}{\#DMF} \Big|_{SLC}$$

Estimation of reliability based on the mean time between delivery of undetected faults and the mean service life:

$$R = \frac{\eta_{SS} \cdot \#SS}{\#DMF} \quad \text{with} \quad \frac{\#SS}{\#DMF} = \frac{\bar{t}_{DMF}}{t_S}$$

$$R = \frac{\eta_{SS} \cdot \bar{t}_{DMF}}{t_S} \quad (1.16)$$

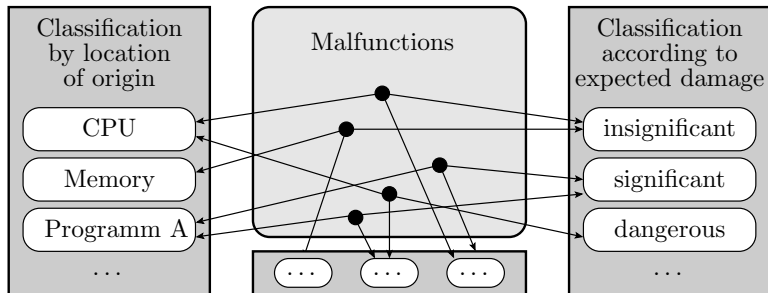
$\#SS, \eta_{SS}$ Number of service time slots per delivered malfunction.

$\#DMF$ Number of malfunctions delivered.

$\bar{t}_{DMF}, \bar{t}_S$ Mean service time between unrecognised malfunctions. Mean service time.

R Reliability.

1.41 Partial reliabilities



The problems within a system can also be counted separately according to different criteria, e.g.

- by location, cause, damage,
- separately for subsystems (CPU, memory, ...),
- separated into faults, disruptions and failures,
- only issues that compromise operational safety or data security.



If each malfunction is assigned to exactly one class* , the sum of all malfunctions is equal to the sum of the malfunctions of all classes:

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

Total malfunction rate equal to the sum of all partial malfunction rates:

$$\zeta = \sum_{i=1}^{\#MFC} \zeta_i \quad (1.17)$$

The reciprocal of the overall reliability is equal to the sum of the reciprocals of the partial reliabilities:

$$\frac{1}{R} = \sum_{i=1}^{\#MFC} \frac{1}{R_i} \quad (1.18)$$

This applies regardless of how malfunctions are handled (without checking, with checking, and refusal to perform in the event of detected malfunctions, ...) and also to sub-types such as detected, avoided, ... malfunctions, crashes, ...

$\#MFC$	Number of malfunction classes.
MF, MF_i	Malfunction. malfunction of class i .
ζ, ζ_i	Total failure rate. Partial failure rate i .
R, R_i	Overall reliability. Partial reliability i .

Example 1.2: Partial reliabilities

In a system, malfunctions are caused by the memory, the processor, the software or other components. Mean time between detected (unrecognised) malfunctions in the subsystems:

Subsystem i	Memory	Processor	Software	others
$\bar{t}_{\text{DMF},i}$	1,000 h	6,000 h	2,000 h	4,000 h

Average service duration $\bar{t}_S = 1$ min. System utilisation $\eta_{SS} = 50\%$.

- What are the values of the four partial failure rates ζ_i and partial reliabilities that can be estimated from the time data? R_i ?*
- What are the failure rate ζ and reliability R of the overall system?*

$\bar{t}_{\text{DMF},i}$	Mean service time between delivery of malfunctions of subsystem i .
$\bar{t}_S, \eta_{SS}$	Average service duration. System utilisation rate.
ζ, ζ_i	Total failure rate. Partial failure rate i .
R, R_i	Overall reliability. Partial reliability i .

Subsystem i	Memory	Processor	Software	others
$\bar{t}_{DMF.i}$	1,000 h	6,000 h	2,000 h	4,000 h

Average service duration $\bar{t}_S = 1$ min. System utilisation $\eta_{SS} = 50\%$.

a) What are the values of the four partial failure rates ζ_i and partial reliabilities that can be estimated from the time data? R_i ?

(1.16)

$$R = \frac{\eta_{SS} \cdot \bar{t}_{DMF}}{\bar{t}_S}$$

Subsystem i	Memory	Processor	Software	others
$\eta_{SS} \cdot \bar{t}_{DMF}$ in min	$3 \cdot 10^4$	$18 \cdot 10^4$	$6 \cdot 10^4$	$12 \cdot 10^4$
R_i in $\left[\frac{DS}{DMF}\right]$	$3 \cdot 10^4$	$18 \cdot 10^4$	$6 \cdot 10^4$	$12 \cdot 10^4$
$\zeta_i = \frac{1}{R_i}$ in $\left[\frac{DMF}{DS}\right]$	$3,33 \cdot 10^{-5}$	$5,56 \cdot 10^{-6}$	$1,67 \cdot 10^{-5}$	$8,33 \cdot 10^{-6}$

R_i, ζ_i

$\left[\frac{DS}{DMF}\right]$

$\left[\frac{DMF}{DS}\right]$

Partial reliability i . Malfunction rate of subsystem i .

Counting value ratio in elivered services per malfunction.

Counting value ratio in malfunctions per delivered service.

Subsystem i	Memory	Processor	Software	others
$\bar{t}_{DMF.i}$	1,000 h	6,000 h	2,000 h	4,000 h

Average service duration $\bar{t}_S = 1$ min. System utilisation $\eta_{SS} = 50\%$.

b) *What are the failure rate ζ and reliability R of the overall system?*

$$(1.9) \quad \zeta = \frac{1}{R} = \frac{\#DMF}{\#DS} \Big|_{SLC}$$

$$(1.17) \quad \zeta = \sum_{i=1}^{\#MFC} \zeta_i$$

$$(1.18) \quad \frac{1}{R} = \sum_{i=1}^{\#MFC} \frac{1}{R_i}$$

$$\zeta = (3,33 \cdot 10^{-5} + 5,56 \cdot 10^{-6} + 1,67 \cdot 1 + 8,33 \cdot 10^{-6}) \left[\frac{DMF}{DS} \right]$$

$$= 6,39 \cdot 10^{-5} \left[\frac{DMF}{DS} \right]$$

$$\frac{1}{R} = \left(\frac{1}{3 \cdot 10^4} + \frac{1}{18 \cdot 10^4} + \frac{1}{6 \cdot 10^4} + \frac{1}{12 \cdot 10^4} \right) \left[\frac{DMF}{DS} \right]$$

$$R = \frac{1}{\zeta} = 1,57 \cdot 10^4 \left[\frac{DS}{NDM} \right]$$



Safety (security)

1.44 Damage caused by malfunctions

Safety for industrial equipment is currently described in terms of safety levels. SIL (**S**afety **I**ntegrity **L**evel) in accordance with IEC 61508:

- SIL1: Minor damage to facilities and property,
- SIL2: Serious damage to equipment, harm to persons,
- SIL3: Injuries, some fatalities,
- SIL4: Disasters, many deaths, serious environmental damage.

set key performance indicators for availability:

- *PFH* (probability of **f**ailure per **h**our),
- *PDF* (probability of **f**ailure on **d**emand), ...

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

Mixing with availability is unfavourable. Here, a subdivision is made according to the type of risk. Definition of (partial) **safety** as the **number of service requests (SR) per safety-related problem (SP)**.

1.45 Safety-related issues

Safety (security) refers to assumed risks:

Safety (security)	Safe from what?
Operational safety	Injury, environmental damage
Access security (data protection)	Data theft
Data security	Data loss
...	...

Rate of problems posing a safety risk / compromise security:

$$\zeta_S = \frac{\#SP}{\#SR} \Big|_{SLC} \quad (1.19)$$

If each problem is assigned to exactly one problem class, the overall rate is the sum of the sub-problem rates of all problem classes:

$$\zeta_S = \sum_{i=1}^{\#SPC} \zeta_{S.i} \quad (1.20)$$

SR, SP Services requests. Safety-related problem.

$\#SPC$ Number of safety and security problem classes.

$\zeta_S, \zeta_{S.i}$ Rate of safety endangering malfunctions in total, for each problem class individually.

1.45 Overall and partial Safety

Safety is the reciprocal of the rate of safety-compromising problems:

$$S = \frac{\#SR}{\#SP} \Big|_{SLC}$$
$$S = \frac{1}{\zeta_S} \quad (1.21)$$

If each problem is assigned to exactly one problem class, the reciprocal of the overall safety is equal to the reciprocal of the sum of all partial safeties:

$$\frac{1}{S} = \sum_{i=1}^{\#SPC} \frac{1}{S_i} \quad (1.22)$$

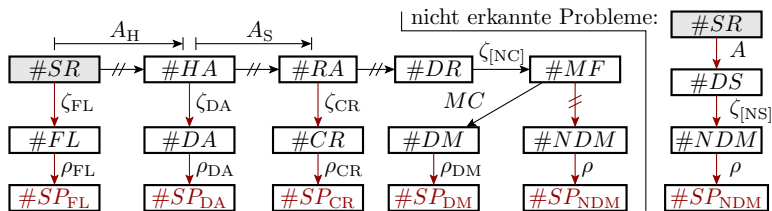
All problems in an IT system

- Hardware failure, denial of service,
- System crash, detected/undetected malfunction.

may pose a security risk at a rate of $\rho_{[.]}$.

S, S_i	Overall safety. Partial safety i .
$\#SPC$	Number of safety and security problem classes.
SLC	Useful estimates only with Sufficiently large count values.

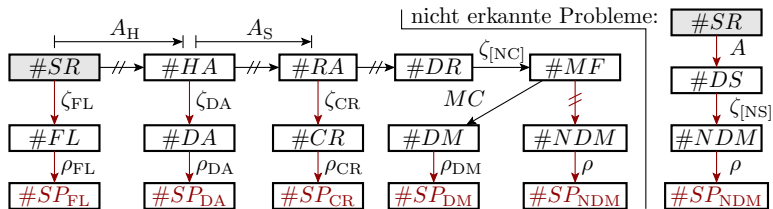
1.46 CVA graph of safety (security) hazards



Rates of safety hazards caused by hardware failures:

$$\zeta_{S.FL} = \zeta_{FL} \cdot \rho_{FL} \quad (1.23)$$

SR, FL	Service request. Hardware failed.
HA, DA	Hardware available. Acceptance denied.
RA, CR	Request accepted. Crash.
DS, MF	Delivered services. Malfunctions.
ζ_{FL}, ζ_{DA}	HW unavailability rate. Service denial rate.
ζ_{CR}, MC	Crash rate. Malfunction coverage.
$\zeta_{[NC/NS]}$	malfunction rate without checks / with no service for identified problems.
$\rho_{[..]}, \#SP_{[..]}$	Proportion and count value of safety-related problems.
Pronounce:	ζ : zeta, ρ : rho.



Rates of security risks arising from refusal to accept delivery:

$$\zeta_{S.DA} = A_H \cdot \zeta_{DA} \cdot \rho_{DA}$$

Incidence rates of safety hazards caused by crashes (CR):

$$\zeta_{S.CR} = A_H \cdot A_S \cdot \zeta_{CR} \cdot \rho_{CR}$$

Rates of safety risks caused by detected malfunctions:

$$\zeta_{S.DM} = A_H \cdot A_S \cdot (1 - \zeta_{CR}) \cdot \zeta_{[NC]} \cdot MC \cdot \rho_{DM}$$

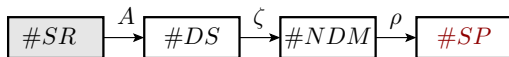
Rates of safety risks caused by undetected malfunctions:

$$\zeta_{S.NDM} = A_H \cdot A_S \cdot (1 - \zeta_{CR}) \cdot \zeta_{[NC]} \cdot (1 - MC) \cdot \rho = A \cdot \zeta_{[NS]} \cdot \rho$$

Overall rate of all problems posing a risk:

$$\zeta_S = \underbrace{\zeta_{S.DA} + \zeta_{S.CR} + \zeta_{S.DM}}_{\text{Risk prevention possible}} + \underbrace{A \cdot \zeta_{[NS]} \cdot \rho}_{\text{Risk prevention impossible}}$$

1.49 Safe State



Ideally, the system switches to a safe state as soon as a problem is detected. Security risks are then limited to those arising from undetected malfunctions:

$$\zeta_S = A \cdot \zeta \cdot \rho$$

For availability $A \approx 1$ and no safety risk arising from detected problems, safety is the quotient of reliability and the proportion of malfunctions that pose a safety risk:

$$S = \frac{R}{\rho} \quad (1.24)$$

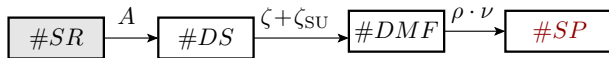
SR, DS Service request. Delivered service.

DS, NDM Unrecognised malfunction. Safety-critical issue.

A, Z, S Availability. Reliability. Safety or security.

ζ, ρ Malfunction rate. Proportion of malfunctions that pose a safety risk.

1.50 Improved safety with safety unit



Reduction of the safety-critical proportion ρ by means of an additional safety unit:

- Emergency power supply, emergency programmes,
- Schutzvorrichtungen, Sicherheitseinrichtungen, ...

which reduce the proportion of safety-critical problems by a factor ν , but which themselves contribute to the number of malfunctions in the delivered product at a rate of ζ_{SU}^* :

$$S_{\text{ASU}} = \frac{1}{(\zeta + \zeta_{\text{SU}}) \cdot \rho \cdot \nu} \quad (1.25)$$

SR, DS	Service request. Delivered service.
DS, NDM	Unrecognised malfunction. Safety-critical issue.
ζ, ζ_{SU}	Rate of delivery malfunctions: system / safety unit.
ρ, ν	Share of safety-critical malfunctions. Reduction of this share by the safety unit.
S_{ASU}	Overall safety (security) of a system with an additional safety unit.
*	Availability one and switching to a safe state whenever a problem is detected.

Example 1.3: Improving vehicle safety

A vehicle has a mean time between undetected malfunctions of 1000 h. The proportion of these that compromise operational safety is 1%, and the mean service duration (mean journey duration) is 1 h. System utilisation is one. An additional electronic control unit with reliability R_{SU} reduces the proportion of problems that compromise safety to one-tenth. Safety is compromised only by undetected malfunctions, availability one.

$$\bar{t}_{\text{NDM}} = 1000 \left[\frac{\text{h}}{\text{NDM}} \right], \bar{t}_{\text{S}} = 1 \left[\frac{\text{h}}{\text{DS}} \right], \eta_{\text{SS}} = 1, \rho = 1\% \left[\frac{\text{SP}}{\text{NDM}} \right], \nu = 0,1.$$

- Reliability and safety without the additional control unit?*
- Minimum reliability R_{SU} of the auxiliary control unit, such that safety is increased at least fivefold ($S_{\text{ASU}} \geq 5 \cdot S$)?*

$\bar{t}_{\text{NDM}}, \bar{t}_{\text{S}}$	Mean service time between unrecognised malfunctions. Mean service time.
η_{SS}	System utilisation rate, percentage of available service time used.
ρ, ν	Share of safety-critical malfunctions. Reduction of this share by the safetyty unit.
S, S_{ASU}	Safety without/with additional unit for safety improvement.



$$\bar{t}_{\text{NDM}} = 1000 \left[\frac{\text{h}}{\text{NDM}} \right], \bar{t}_{\text{S}} = 1 \left[\frac{\text{h}}{\text{DS}} \right], \eta_{\text{SS}} = 1, \rho = 1\% \left[\frac{\text{SP}}{\text{NDM}} \right], \nu = 0,1.$$

a) *Reliability and safety without the additional control unit?*

$$(1.16) \quad R = \frac{\eta_{\text{SS}} \cdot \bar{t}_{\text{DMF}}}{\bar{t}_{\text{S}}}$$

$$(1.24) \quad S = \frac{R}{\rho}$$

Reliability:

$$R = \frac{\eta_{\text{SS}} \cdot \bar{t}_{\text{NDM}}}{\bar{t}_{\text{S}}} = 1000 \left[\frac{\text{DS}}{\text{NDM}} \right]$$

Of which $\sigma = 1\%$ poses a safety risk:

$$S = \frac{R}{\rho} = 10^5 \left[\frac{\text{DR}}{\text{SP}} \right]$$



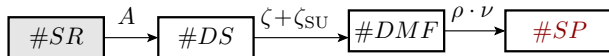
1. Dependability

4. Safety (security)

$$\bar{t}_{\text{NDM}} = 1000 \left[\frac{h}{\text{NDM}} \right], \bar{t}_S = 1 \left[\frac{h}{\text{DS}} \right], \eta_{\text{SS}} = 1, \rho = 1\% \left[\frac{\text{SP}}{\text{NDM}} \right], \nu = 0,1.$$

b) *Minimum reliability R_{SU} of the auxiliary control unit, such that safety is increased at least fivefold ($S_{\text{ASU}} \geq 5 \cdot S$)?*

$$(1.25) \quad S_{\text{ASU}} = \frac{1}{(\zeta + \zeta_{\text{SU}}) \cdot \rho \cdot \nu}$$



Maximum reduction in reliability caused by the control unit:

$$S_{\text{ASU}} = \frac{1}{(\zeta + \zeta_{\text{SU}}) \cdot \rho \cdot \nu} \geq 5 \cdot S = \frac{1}{\zeta \cdot \rho}$$

$$(\zeta + \zeta_{\text{SU}}) \cdot \nu = 0,1 \cdot (\zeta + \zeta_{\text{SU}}) \leq \frac{\zeta}{5}$$

$$\zeta_{\text{SU}} \leq \zeta$$

$$R_{\text{SU}} \geq R = 10^3 \left[\frac{\text{DS}}{\text{NDM}} \right]$$

The auxiliary control unit must be at least as reliable as the vehicle.

R, R_{SU}

Reliability of the system. Reliability of the safety unit.

1.52 Notes on the exercise

There are currently ethical debates as to whether autonomous vehicles should run over children, pensioners, ... in critical driving situations. This is not a constructive approach:

- Smart responses, such as detecting and deciding who should be run over, require complex additional systems that tend to be unreliable.
- Unreliable safety devices are not a good idea.
- It is not possible to programme any emergency response at all for undetected malfunctions..

High levels of safety (security) require:

- high system reliability, high fault coverage,
- Safety measures and equipment designed to prevent safety risks arising from identified problems.
- Insurance for the system to cover the residual risk arising from unforeseeable – and therefore untreatable – problems.



Summary

1.53 Lecture: Test and dependabilityt

This lecture aims to reorganise the field of study that deals with the most costly feature of IT systems – dependability – for the purpose of presenting knowledge in a concise manner:

- building blocks of knowledge that build on one another,
- concise definitions and models,
- which cover many of the issues important in practice.

(L) Important for the exam are the simple fundamental concepts and formulae, and the conditions under which they apply. After all, you will come across these in practice.

(B) More complex examples are often used merely for illustrative purposes. In the exam, it is sufficient to correctly identify the underlying concepts when presented with slides or images, and to apply the formulas correctly when working through example questions.

1.54 Key points (L)

Dependability describes the extent to which trust in an IT system is justified by counting positive and negative experiences: malfunctions, crashes, correct performance, ...

Estimates are reliable only if the count values are large enough.

Breaking down into countable experiences via the service model:

A system that generates output from input on request.

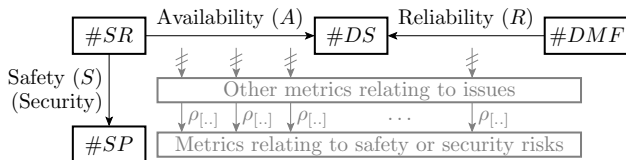
Dependability is ensured at three levels:

- 1 Dealing with problems during service,
- 2 Troubleshooting before and during use,
- 3 Fault prevention by means of improved design and manufacturing processes,

By iteration of checks, attempts to resolve problems and success checks. An idealised error culture in which all identified problems are resolved.

Definition of parameters using count-value assignment graphs.

1.55 Measures of dependability ($\mathbb{L}$)



Measures for delivery:

■ Availability: (1.1)

$$A = \frac{\#DS}{\#SR} \Big|_{SLC}$$

Measures of correctness:

■ Reliability: (1.8)

$$R = \frac{\#DS}{\#DMF} \Big|_{SLC}$$

■ MF rate: (1.9)

$$\zeta = \frac{1}{R} = \frac{\#DMF}{\#DS} \Big|_{SLC}$$

Characteristics of safety (security) risks:

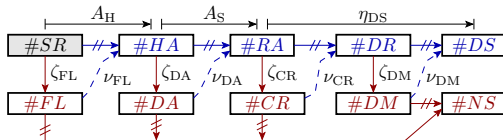
■ Rate safety probl.: (1.19)

$$\zeta_S = \frac{\#SP}{\#SR} \Big|_{SLC}$$

■ safety: (1.21)

$$S = \frac{1}{\zeta_S}$$

1.56 Overall and partial availability ($\mathbb{B}^*$)



Categorised by cause of unavailability (and reaction to it):

- FL: Hardware failure,
- DA: Denial of acceptance,
- CR: Crash whilst running a service,
- DM: detected malfunction.

Hardware availability: $(1.4) \quad A_H = (1 - \zeta_{FL}) + \zeta_{FL} \cdot \nu_{FL}$

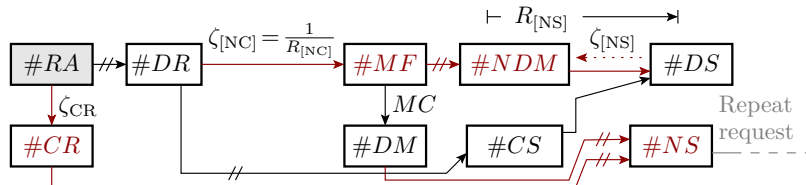
Service availability: $(1.5) \quad A_S = (1 - \zeta_{DA}) + \zeta_{DA} \cdot \nu_{DA}$

Overall availability: $(1.6) \quad A = A_H \cdot A_S \cdot \eta_{DS}$

*

Understanding the CVA graph. Derivation and application of the formulae.

1.57 Delivery rate and reliability



(B) Delivery rate (without re-requests):

$$(1.10) \quad \eta_{DS} = (1 - \zeta_{CR}) \cdot (1 - \zeta_{[NS]} \cdot MC)$$

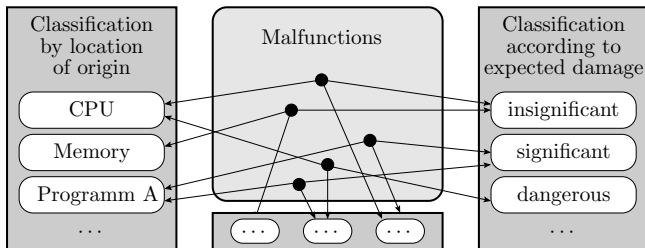
(B) Reliability:

$$(1.14) \quad R_{[NS]} = \frac{R_{[NC]} \cdot (1 - \zeta_{OMF})}{(1 - MC)}$$

(L) Where the rate of observable malfunctions is low $\zeta_{OMF} \ll 1$:

$$(1.15) \quad R_{[NS]} = \frac{R_{[NC]}}{1 - MC} = \frac{R_{[NC]}}{MT}$$

1.58 Partial reliabilities ($\mathbb{L}$)



If all malfunctions are assigned to classes in such a way that each one appears exactly once:

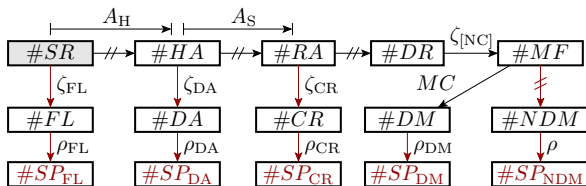
- Total failure rate equals the sum of the partial failure rates:

$$(1.17) \quad \zeta = \sum_{i=1}^{\#MFC} \zeta_i$$

- The reciprocal of the overall reliability is equal to the reciprocal of the sum of the partial reliabilities:

$$(1.18) \quad \frac{1}{R} = \sum_{i=1}^{\#MFC} \frac{1}{R_i}$$

1.59 Safety (Security) ($\mathbb{B}$)



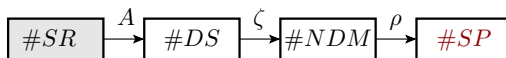
($\mathbb{L}$) Classification of security measures by type of risk into operational, access and data safety/security.

($\mathbb{B}$) Classification by cause and risk management:

- Unavailability due to failure, refusal, crashes,
- Malfunction functions, detected and undetected.

All these problems can lead to safety-critical issues with a low rate ρ . Partial and overall safety levels can be read from the CVA graph. Furthermore, it is assumed that a safe state is established whenever a problem is detected, so that the only remaining risks stem from undetected malfunctions.

1.60 Safe state, safety equipment ($\mathbb{L}$)



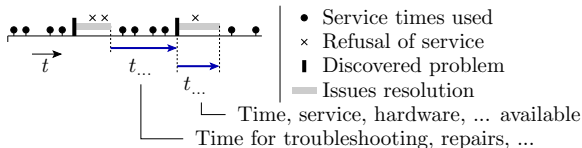
When transitioning to a safe state in the event of any detected problem (elimination of the risk of failures, crashes, detected malfunctions, etc.), **safety is proportional to reliability:**

$$(1.24) \quad S = \frac{R}{\rho}$$

Safety equipment designed to reduce the proportion of safety-critical issues is itself a source of problems that compromises reliability. Overall safety improvement:

$$(1.25) \quad S_{ASU} = \frac{1}{(\zeta + \zeta_{SU}) \cdot \rho \cdot \nu}$$

1.61 Estimation based on measured times (B)



Availability in terms of mean time between problems and mean time to resolution:

$$(1.3) \quad A = \frac{\bar{t}_A}{\bar{t}_A + \bar{t}_{RP}}$$

Hardware availability as a function of the mean time between failures and the mean time to repair:

$$(1.7) \quad A_H = \frac{\bar{t}_{HA}}{\bar{t}_{HA} + \bar{t}_R} \quad \bar{t}_{HA} \gg \bar{t}_R \quad 1 - \frac{\bar{t}_R}{\bar{t}_{FL}}$$

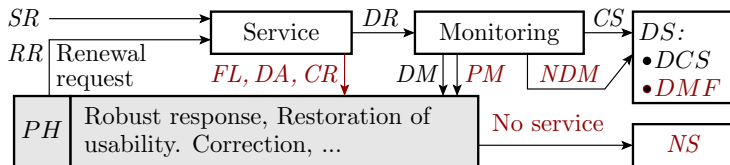
Reliability as a function of the mean time between malfunctions and average service duration, etc.:

$$(1.16) \quad R = \frac{\eta_{SS} \cdot \bar{t}_{DMF}}{\bar{t}_S}$$



Problems under use

1.62 Dealing with problems under use



Iterations comprising checks, attempts to resolve problems and success check.

- For crashes (CR) and detected malfunctions (DM), no result (NS) or repeat request (RR).
- Undelivered services (NS) reduce the delivery rate η_{DS} .
- Delivered Malfunctions (DMF) compromise reliability R and safety (or security) S .

SR, RR Service request. Repeat request.

DR, CS Delivered result. Correct service.

NDM, PM Not detected malfunction. Phantom malfunction.

DS, NS Delivered service. No service.

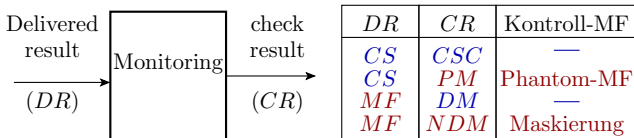
FL, DA Non-availability due to hardware failure or denial of acceptance.

CR, TS Crash. Troubleshooting.



Monitoring

1.63 Monitoring



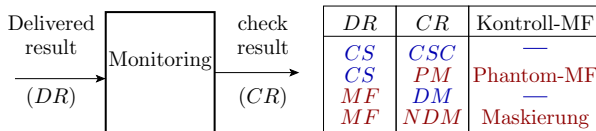
Responding to problems that arise during use (service denial, crashes, malfunctions, etc.) requires monitoring. In the simplest case, a monitoring system provides a binary ‘good/bad’ result, with the possible classification errors:

- Masking a detectable malfunction and
- Phantom malfunction.

Service refusals, e.g. due to a system crash, are always detected.

<i>DS, CR</i>	Delivered service. Check result.
<i>CS, MF</i>	Correct service. Malfunction.
<i>DM, CSC</i>	Detected malfunction. Correct service detected as correct.
<i>NDM, PM</i>	Not detected malfunction, phantom malfunction.

1.64 Monitoring parameters



Malfunction coverage, proportion of detectable malfunctions:

$$MC = \frac{\#DM}{\#MF} \Big|_{SLC} \quad (1.26)$$

Malfunction leakage as a counter frequency:

$$MT = \frac{\#NDM}{\#MF} \Big|_{SLC} \quad (1.27)$$

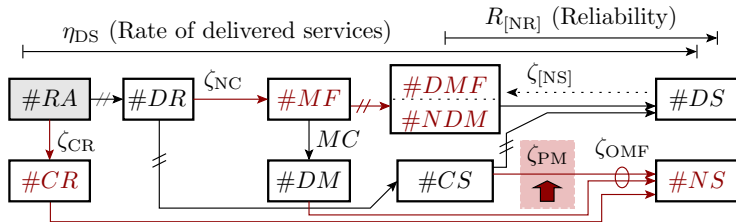
Phantom malfunction rate, the proportion of correct service operations that are classified as malfunctions:

$$\zeta_{PM} = \frac{\#PM}{\#CS} \Big|_{SLC} \quad (1.28)$$

MC, MT Malfunction coverage. Malfunction leakage.

ζ_{PM} Phantom malfunction rate.

1.65 CVA graph without repeat request

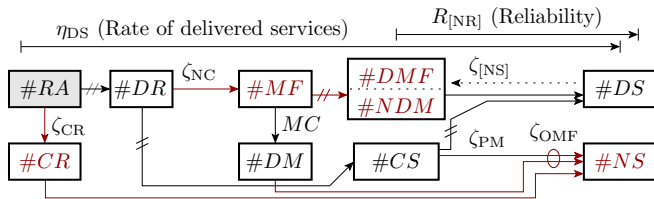


Compared with slide 1.36, an additional edge for phantom malfunctions:

- Malfunctions (MF) are with a frequency of MC detected (DM) and otherwise undetected malfunction (NDM).
- Unless a new request is made, crashes (CR), detected malfunctions (DM) and phantom malfunctions (MF) are non-delivered services (NS).

ζ_{CR}, ζ Crash rate. Malfunction rate.
 MC, ζ_{PM} Malfunction coverage. Phantom malfunction rate.
 $\zeta_{[NC/NS]}$ malfunction rate without checks / with no service for identified problems.

1.66 Delivery rate without re-requests



$$\eta_{DS} = \frac{\#DS}{\#RA} \Big|_{ACR} = (1 - \zeta_{CR}) \cdot (\zeta \cdot (1 - MC) + (1 - \zeta) \cdot (1 - \zeta_{PM}))$$

$$= (1 - \zeta_{CR}) \cdot (\zeta - \zeta \cdot MC + 1 - \zeta - \zeta_{PM} + (\zeta \cdot \zeta_{PM})^*) \quad (1.29)$$

With the rate of observable malfunctions,

$$\zeta_{OMF} = \zeta_{PM} + \zeta \cdot MC \quad (1.30)$$

which is greater than that of (eq. 1.11) by ζ_{PM} , the following still holds:

$$(1.12) \quad \eta_{DS} = (1 - \zeta_{CR}) \cdot (1 - \zeta_{OMF})$$

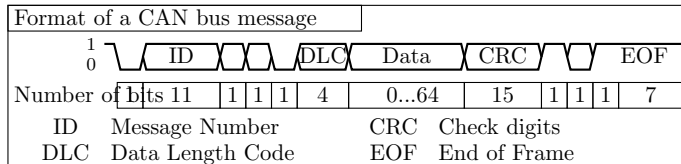
*

Quadrat sehr kleiner Werte.



Format checks

1.68 Format and Value Checks



A service includes data embedded in a format:

- Format: value-independent attributes (value ranges, ...).
- Data: Attributes related to value representation.

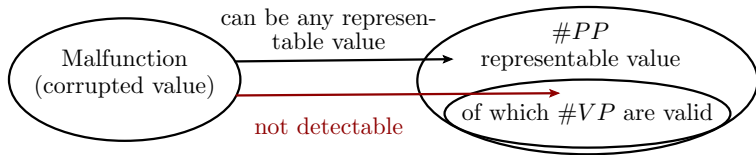
Classification of monitoring Procedures for digital Services:

- 1 Format Checks: Check only characteristics that are independent of value. Service results with format errors are always incorrect. Results with the correct format may still contain incorrect values. Check for admissibility only.
- 2 Value Checks: (Additional) verification of values.

Format Checks are easier to implement and often result in higher *MC* and lower phantom malfunction rates than value checks.

1.69 Information redundancy

Format checks (error-detecting codes, check digits, valid range checks, etc.) are based on information redundancy.



Malfunction coverage tends to be higher the smaller the proportion of valid bit patterns is. If all possible corruption scenarios occur with equal frequency, and all invalid patterns are recognized as invalid:

$$\mu_{MC} = 1 - \frac{\#VP}{\#PP} \quad (1.31)$$

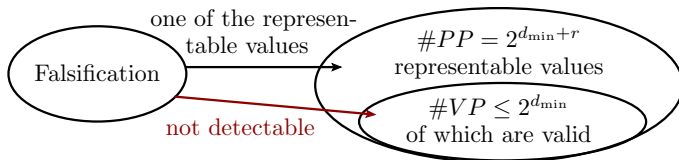
$$\mu_{MT} = \frac{\#VP}{\#PP} \quad (1.32)$$

$\#VP$	Number of valid bit patterns.
$\#PP$	Number of presentable bit patterns.
μ_{MC}	Expected malfunction coverage of monitoring.

1.70 Redundant Bits

Suppose that $d_{\min}$ data bits are sufficient to distinguish all possible values. When represented using r additional (redundant) bits:

$$b = r + d_{\min}$$



$$\mu_{MC} = 1 - \frac{\#VP}{\#PP} \geq 1 - \frac{2^{d_{\min}}}{2^{d_{\min}+r}}$$

Minimum coverage for malfunctions:

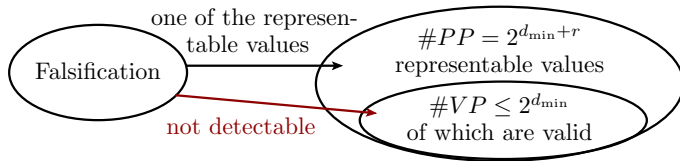
$$\mu_{MC} = 1 - 2^{-r} \quad (1.33)$$

$\#VP, \#PP$ Number of valid bit patterns. Number of presentable bit patterns.

μ_{MC}, r Expected malfunction coverage. Number of redundant bits.

$d_{\min}, b$ Required number of bits to differ between all permissible values. Total number of bits.

μ_{MC}, r Expected malfunction coverage. Number of redundant bits.



Maximum expected malfunction leakage:

$$\mu_{MT} = 2^{-r} \quad (1.34)$$

r	10	20	30
μ_{MT}	10^{-3}	10^{-6}	10^{-9}

If all valid values are recognized as valid, there will be no phantom malfunctions during format checks:

$$\zeta_{Phan} = 0 \quad (1.35)$$

$\#VP, \#PP$	Number of valid bit patterns. Number of presentable bit patterns.
d_{min}, b	Required number of bits to differ between all permissible values. Total number of bits.
r	Number of redundant bits.
μ_{MT}	Expected malfunction leakage.
ζ_{PM}	Phantom malfunction rate.

1.72 Reliability Improvement

Reliability improvement through **ideal format checking** with r redundant bits. No re-request, low rate of observable malfunctions $\zeta_{\text{OMF}} \ll 1$:

(1.32)

$$\mu_{\text{MC}} = 1 - 2^{-r}$$

(1.15)

$$R_{[\text{NS}]} = \frac{R_{[\text{NC}]}}{1 - \mu_{\text{MC}}} = \frac{R_{[\text{NC}]}}{\mu_{\text{MT}}}$$

$$R_{[\text{NS}]} = 2^r \cdot R_{[\text{NC}]} \quad (1.36)$$

Ideal: Distortion not disproportionately often into valid values. This generally applies to error-correcting codes and check digits used in the transmission and storage of data (sec. 5.3).

Most **other format checks** (value range, syntax, etc.) do not have comparably good features, but they still offer a very good cost-benefit ratio (sec. 5.4).

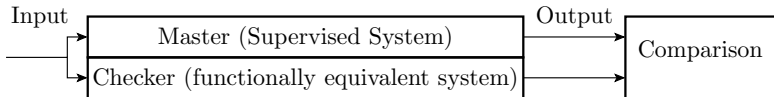
ζ_{OMF}	Rate of observable malfunctions.
$R_{[\text{NC/NS}]}$	Reliability without checks / with no service for identified problems.
$\mu_{\text{MC}}, \mu_{\text{MT}}$	Expected malfunction coverage. Expected malfunction leakage.
r	Number of redundant bits.



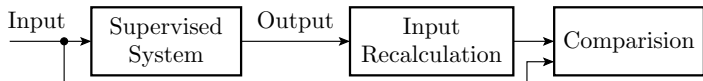
Value checks, diversity

1.73 Universal Value-Checking Methods

- **Duplication and Comparison** (Master-Checker Principle). Applicable to all deterministic services.

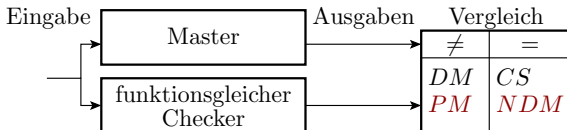


- **Loop test** (input recalculation and comparison), e.g., monitoring transmission by receiving data and comparing the received data with the transmitted data. Only for reversible, one-to-one functions.



- **Proof of correctness:** For example, when searching for a path from A to B through a graph, all paths from A to B are valid solutions. Assume a correctness condition whose satisfaction proves correctness.

1.74 Eigenschaften Verdopplung und Vergleich



Maskiert werden identische Fehlfunktionen durch übereinstimmende Fehler in Master und Checker. Zu erwartende Fehlfunktionsabdeckung sei definiert als die Diversitätsrate (siehe auch nächste Folie):

$$\mu_{MC} = \eta_{Div} \quad (1.37)$$

Die Phantom-Fehlfunktionsrate ist die Rate der Checker-Fehlfunktionen ohne gleichzeitig gleiche Master-Fehlfunktion:

$$\zeta_{PM} = \eta_{Div} \cdot \zeta_{[Chk]} \quad (1.38)$$

DM, PM Detected malfunction. Phantom malfunction.

CS, NDM Correct service. Undetected malfunction.

η_{Div} Diversity rate, defined as fault coverage by doubling and comparison.

$\zeta_{[Mst/Chk]}$ Malfunction rate: Master/Checker.



1.75 Diversität und Diversitätsrate

Diversität von IT-Systemen beschreibt die **Verschiedenartigkeit** der Problemwirkung bei Mehrfachberechnung. Praktisch gilt:

- Probleme (Fehlfunktionen, Abstürze) sind sehr selten und
- es gibt sehr viele Verfälschungsmöglichkeiten.

Gleichzeitige gleiche Wirkung durch Zufall praktisch ausgeschlossen. Übereinstimmung hat immer eine gemeinsame Ursache:

- falsche Eingaben, gleiche Fehler,
- Hardware-Ausfall, ...

Die **Diversitätsrate** η_{Div} nach (eq. 1.36) definiert als **Fehlfunktionsabdeckung** durch Verdopplung und Vergleich ist gleichzeitig die **Korrekturrate durch wiederholte Berechnung**.

Die Korrekturrate für Abstürze und erkannte Fehlfunktionen und damit die Diversität durch Zählen der versuchten und erfolgreichen Korrekturversuche bestimmbar.

1.76 Zuverlässigkeitsgewinn, beobachtbare MF, ...

$$(1.14) \quad R_{[NS]} = \frac{R_{[NC]} \cdot (1 - \zeta_{OMF})}{(1 - MC)}$$

$$(1.12) \quad \eta_{DS} = (1 - \zeta_{CR}) \cdot (1 - \zeta_{OMF})$$

$$(1.30) \quad \zeta_{OMF} = \zeta_{PM} + \zeta \cdot MC$$

Mit $\mu_{MC} = \eta_{Div}$ und $\zeta_{PM} = \eta_{Div} \cdot \zeta_{[Chk]}$ ist die Rate der beobachteten Fehlfunktionen die Summe der Raten der diversitären Master- und Checker-Fehlfunktionen:

$$\zeta_{OMF} = \eta_{Div} \cdot (\zeta_{[Mst]} + \zeta_{[Chk]}) \quad (1.39)$$

Nichterbringungsrate ist die Summe der Raten für Abstürze plus der diversitären Master- und Checker-Fehlfunktionen:

$$\begin{aligned} \eta_{DS} &= (1 - \zeta_{CR}) \cdot (1 - \eta_{Div} \cdot (\zeta_{[Mst]} + \zeta_{[Chk]})) \\ 1 - \eta_{DS} &\approx \zeta_{CR} - \eta_{Div} \cdot (\zeta_{[Mst]} + \zeta_{[Chk]}) \end{aligned} \quad (1.40)$$

Zuverlässigkeitsgewinn umgekehrt proportional zur Nicht-Diversitätsrate:

$$R_{[NS]} = \frac{R_{[NC]} \cdot (1 - \zeta_{OMF})}{(1 - \eta_{Div})} \quad (1.41)$$

1.77 Natürliche Diversität

Mehrfachberechnung und Vergleich mit derselbe Hard- und Software erkennt praktisch alle Fehlfunktionen durch Störungen, aber kaum Fehlfunktionen durch Fehler.

Diversitätsrate gleich Fehlfunktionsanteil durch Störungen. Bestimmbar durch wiederholte Ausführung mit demselben System nach jedem Problem.

Minderung der Fehlfunktionsraten auf maximal die durch Fehler:

$$\zeta \leq \zeta_F$$

Erhöhung der Zuverlässigkeit mindestens auf die fehlerbezogene Teilzuverlässigkeit:

$$R \geq R_F$$

ζ, ζ_F	Overall failure rate. Failure rate due to faults.
R, R_F	Overall reliability. Partial reliability by faults.



1.78 Erweiterte Diversität

Konstruktive und organisatorische Maßnahmen zur Erhöhung der Diversitäten durch Vermeidung gemeinsamer Problemursachen:

Erweiterte Diversität	konstr. und org. Maßnahmen	Erhöhung der Diversitätsrate für
Hardware-Diversität	Ausführung auf verschiedener HW	Fertigungsfehler, Ausfälle
Hardware-Entwurfsdiversität	unabhängig entworfene HW	<u>zusätzlich</u> HW-Entwurfsfehler
Syntaktische Diversität	unterschiedlich übersetzte SW	<u>zusätzlich</u> SW-Übersetzungsfehler
Algorithmische Diversität	unabhängige Programmierung	<u>zusätzlich</u> SW-Entwurfsfehler
diversitäre Nutzung	Wiederholung mit anderen SR*	<u>zusätzlich</u> Eingabefehler

HW, SW, SR Hardware. Software. Service request.

* Oft abweichenden Sollwerte. Ungeeignet für Mehrfachberechnung und Vergleich.

Nachweis von Fehlfunktionen durch Software-Fehler verlangt **diversitäre Entstehung**, im Extremfall:

- kompletter Software-Entwurf mindestens zweimal,
- durch getrennte Teams, keine Kommunikation,
- aus einer gemeinsamen Spezifikation, ...

Die ursprüngliche euphorische Meinung, dass so Diversität gegenüber allen Fehlern, außer denen in der Spezifikation erzielbar ist, nicht bestätigt. Die direkte oder indirekte Kommunikation der Entwicklungsteams über die Interpretation der Spezifikation, während des Test etc. trägt Gemeinsamkeiten in die Entwürfe. Neigung von Menschen, gewisse Fehler zu wiederholen, ... erzielbare Diversitätsrate nach [3] max.

$$\eta_{\text{Div}} \leq 90\%$$

Nach (eq. 1.40) Zuverlässigkeitsverbesserung max. 10. Eine Kontrolle mit $r = 10$ Bit Informationsredundanz erreicht nach Gl. 1.32 eine Verbesserung um 10^3 mit viel weniger Entwurfsaufwand und ohne Phantomfehlfunktionen

1.80 Diversitäre Nutzung

Reale IT-Systeme bieten für die meisten Aufgaben viele Lösungswege (funktionale Redundanz), bei denen vorhandene Fehler oft unterschiedliche und zum Teil gar keine Wirkung haben.

Diversitäre Nutzung lässt eine viel höherer Diversität als Wiederholung mit denselben Eingaben erwarten. Jedoch oft unterschiedliche richtige Ergebnisse und damit **Ungeeignet für Verdopplung und Vergleich**.

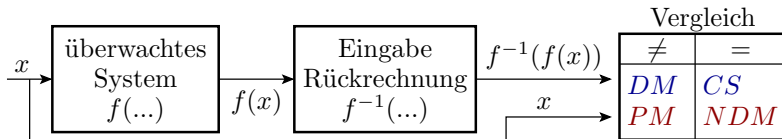
Standardverfahren für die händische Korrektur durch Neuaufrufung. Intuitiv probieren Nutzer solange alternative Lösungswege, bis Problem umgangen (slide 1.105).

Händischen Probieraufwand abschätzungsweise proportional zur Nichterbringungsrate

$$(1.39) \quad 1 - \eta_{DS} \approx \zeta_{CR} - \eta_{Div} \cdot (\zeta_{[Mst]} + \zeta_{[Chk]})$$

die damit ein wichtiges Maß für die Wirtschaftlichkeit von IT-Einsatz ist.

1.81 Eigenschaften Loop-Test



Da $f(\dots)$ und $f^{-1}(\dots)$ sich im Algorithmus unterscheiden, ist auch ohne zusätzliche konstruktive und organisatorische Maßnahmen größere Verschiedenartigkeit der Wirkung gleicher Fehlern, und damit eine höhere Fehlfunktionsabdeckung und Zuverlässigkeit zu erwarten.

Loop-Test nur anwendbar, wenn zu $f(\dots)$ eine Umkehrfunktion $f^{-1}(\dots)$ konstruierbar ist, idealerweise eine einfachere, wie z.B. die Quadratbildung zur Kontrolle der Wurzelberechnung.

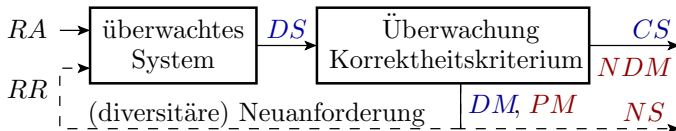
$f(\dots)$ Function of the system to be monitored.

$f^{-1}(\dots)$ Inverse function.

DM, *PM* Detected malfunction. Phantom malfunction.

CS, *NDM* Correct service. Undetected malfunction (product fault).

1.82 Spezifizierter Korrektheitsnachweis



Wenn eine hinreichend **Kontrollbedingung** für den Korrektheitsnachweis existiert (vorgegeben ist):

- Sortieren einer Liste $\Rightarrow$ Liste sortiert und enthält alle Elemente,
- Suche Weg durch einen Graphen $\Rightarrow$ Wegkontrolle auf Zulässig.,
- Suche Test für Fehlernachweis $\Rightarrow$ Fehlersimulation, ...

Fehlfunktionsdurchlässigkeit MT der Kontrolle oft sehr gering, aber ein Suchalgorithmus »Probiere, bis Korrektheitsnachweis erbracht« versucht falsche Ergebnisse μ_{Trl} -mal durch die Kontrolle zu schleusen.

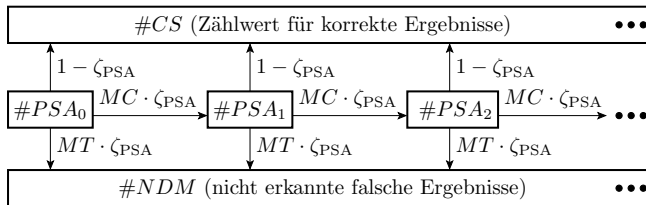
RA, DS Accepted request. Delivered service.

CS, NDM Correct service. Undetected malfunction (product fault).

DM, PM Detected malfunction. Phantom malfunction.

NS, RR Cancellation with no service. Renewed request.

1.83 Zufallsuche mit Korrektheitsnachweis



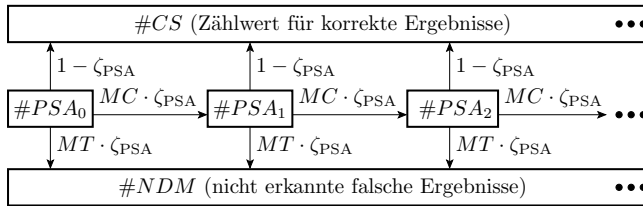
Jede Zählwert $\#PSA_i$ ist je mit der Rate von

- $1 - \zeta_{PSA}$ ein korrektes Ergebnis $\#CS$,
- $MC \cdot \zeta_{PSA}$ ein weiterer Lösungsversuch $\#PSA_{i+1}$ und
- $(1 - MC) \cdot \zeta_{PSA}$ eine nicht erkennbare Fehlfunktion $\#NDM$.

Erbringungsrate je Lösungsanforderung ist eins:

$$\eta_{DS} = \frac{\#CS + \#NDM}{\#PSA_0} = (1 - \eta_{PSA} + (1 - MC) \cdot \zeta_{PSA}) \cdot \sum_{i=0}^{\infty} (MC \cdot \zeta_{PSA})^i = 1$$

$$\stackrel{(SGS)}{=} \frac{1 - MC \cdot \zeta_{PSA}}{1 - MC \cdot \zeta_{PSA}} = 1$$



Dafür im Mittel erforderliche Versuchsanzahl:

$$\mu_{\text{Trl}} = \sum_{i=0}^{\infty} \eta_{\text{PSA}}^i = \frac{\sum_{i=0}^{\infty} \#DR_i}{\#DR_0} = \sum_{i=0}^{\infty} (MC \cdot \zeta_{\text{PSA}})^i \stackrel{(\text{SGS})}{=} \dots$$

$$\mu_{\text{Trl}} = \frac{1}{1 - MC \cdot \zeta_{\text{PSA}}} \quad (1.42)$$

Fehlfunktionsrate der Lösungen:

$$\zeta = \frac{\#NDM}{\#PSA_0} = (1 - MC) \cdot \zeta_{\text{PSA}} \cdot \sum_{i=0}^{\infty} (MC \cdot \zeta_{\text{PSA}})^i \stackrel{(\text{SGS})}{=} \frac{(1 - MC) \cdot \zeta_{\text{PSA}}}{1 - MC \cdot \zeta_{\text{PSA}}}$$

$$\zeta = \mu_{\text{Trl}} \cdot MT \cdot \zeta_{\text{PSA}} \quad (1.43)$$

$\zeta_{\text{PSA}}, \mu_{\text{Trl}}$ Problem-solving rate. Expected number of trials.
 $\#PSA_i, \zeta$ Count for solution attempt i . Overall failure rate.
 MC, MT Malfunction coverage. Malfunction leakage.

1.87 Lange Suche, Kontrollgüte, KI

$$(1.41) \quad \mu_{\text{Trl}} = \frac{1}{1 - MC \cdot \zeta_{\text{PSA}}}$$

$$(1.42) \quad \zeta = \mu_{\text{Trl}} \cdot MT \cdot \zeta_{\text{PSA}}$$

Die Rate der Fehlfunktionen der insgesamt durch die Kontrolle geschleusten Lösungen nimmt proportional mit der erforderlichen Versuchsanzahl für die Lösungsfindung zu.

Aktuelle KI-Dienste durchmustern große Datenmengen mit unscharf formulierten Suchkriterien. Es ist weniger verwunderlich, dass die Antworten zum Teil falsch sind, als dass auch brauchbare Ergebnisse dabei sind. Zuverlässige KI verlangt gute Strategien, um die Lösungen mit wenigen Versuchen zu finden und vor allem gute Kontrollen.

$\zeta_{\text{PSA}}, \mu_{\text{Trl}}$ Problem-solving rate. Expected number of trials.

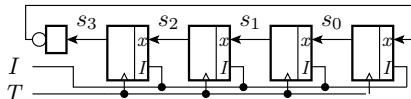
MC, MT Malfunction coverage. Malfunction leakage.

ζ Overall failure rate.

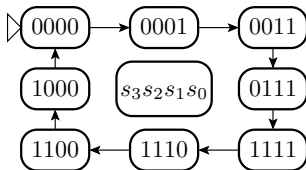


Time monitoring

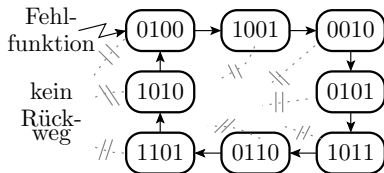
1.89 Absturz und Zeitüberwachung



Soll-Funktion

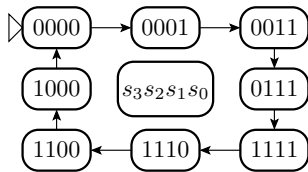


Funktion nach einem Absturz

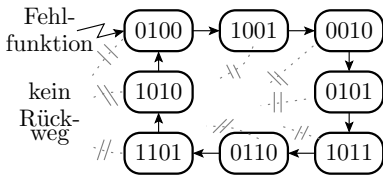


- Automaten und Programme nutzen nur einen kleinen Teil der $2^{\#SB}$ möglichen Zustände ($\#SB$ – Anzahl Zustandsbits).
- Der 4-Bit-Johnson-Zähler durchläuft zyklisch 8 der 16 Zustände.
- ...

Soll-Funktion

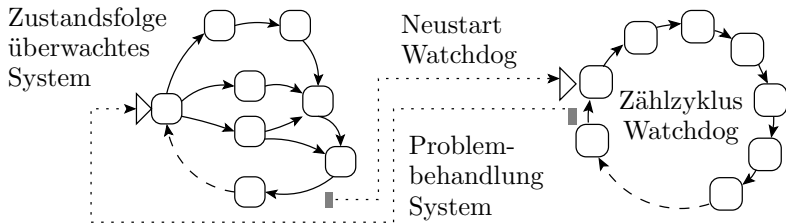


Funktion nach einem Absturz



- Die restlichen (redundanten) 8 Zustände bilden einen Zyklus, der nicht mehr verlassen wird.
- Der Übergang in unzulässige Zustände, die ohne Neuinitialisierung nicht verlassen werden, ist ein Absturz.
- Komplexer Hardware und Software hat Millionen von Zustandsbits und unüberschaubar viele Absturzmöglichkeiten.
- **Ein Absturz** (CR) **wird** immer daran **erkannt**, dass das System keine Ergebnisse (DR) mehr liefert.
- Absturzbehandlung: Zeitüberwachung, Abbruch, Leistungsverzicht oder Wiederholung. Erfolgsrate der Wiederholung gleich der Diversitätsrate gegenüber Abstürzen (verg. Folie 1.75).

1.91 Watchdog



Ein Watchdog ist ein Zeitzähler, der vom überwachten System, bevor er überläuft, zurückgesetzt werden muss. Bei Überlauf Bearbeitungsabbruch und Problembehebung.

Ein Hardware-Watchdog ist ein Zählregister im Prozessor

- einmal gestartet vom Testobjekt selbst nicht mehr ausschaltbar
- und einem speziellen Maschinenbefehl rücksetzbar.

Bei Überlauf Neustart oder Interrupt mit Problembehebungsroutine.



Correction

1.92 Umgang mit erkannten Problemen

Bei ausreichend hoher Erbringungsrate, Leistungsverzicht oder händische Versuche, die gewünschten Leistungen zu erhalten:

- 1 Wiederholte identische Anforderung an denselben Service-Anbieter.
- 2 Geänderte Anforderung an denselben Anbieter oder
- 3 anderer Anbieter.

(2) und (3) diversitäre Nutzung, kaum automatisierbar.

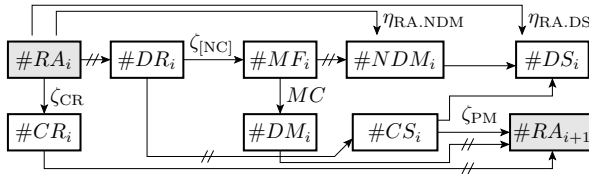
Automatismen für (1) unterteilbar in:

- Neuanforderung bis zur Erbringung und
- begrenzte Anzahl von Neuanforderung (typ. max. eins).

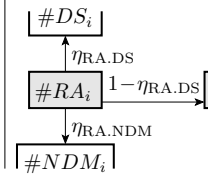
Einen weiteren ganz anderen Korrekturautomatismus bieten fehlerkorrigierende Codes, allerdings nur für Datenübertragung und Speicherung.

1.93 CVA-Graphvereinfachung

CVA-Graph für akzeptierte Service-Anforderungen



Vereinfacht



Rate der erbrachte Leistungen je akzeptierte Service-Anforderung

$$\frac{\#DS_i}{\#RA_i} = \eta_{RA.DS} = (1 - \zeta_{CR}) \cdot (\zeta_{[NC]} \cdot (1 - MC) + (1 - \zeta) \cdot (1 - \zeta_{PM}))$$

RA_i, DS_i Request i accepted. Service provided for request i .

NDM_i Not detected (delivered) malfunction for accepted request i .

CR_i, DR_i Overall failure rate.

DM_i, CS_i Crash at request i . Result delivered for request i .

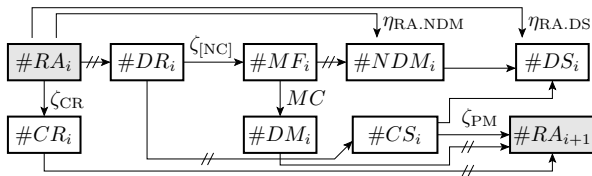
$\eta_{RA.DS}$ The rate at which accepted requirements are delivered as services.

$\eta_{RA.NDM}$ The rate of accepted requirements that are undetected faults.

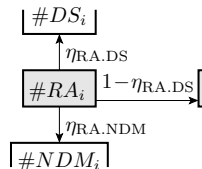
$\zeta_{CR}, \zeta_{[NC]}$ Crash rate. Malfunction rate without monitoring.

MC, ζ_{PM} Malfunction coverage. Phantom malfunction rate.

CVA-Graph für akzeptierte Service-Anforderungen



Vereinfacht



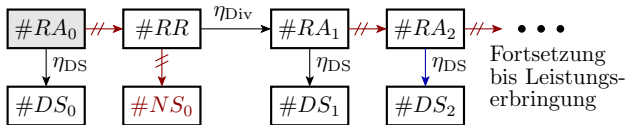
Rate der nicht erkannten (ausgelieferten) Fehlfunktionen je akzeptierte Service-Anforderung (Siehe auch Folie 1.66):

$$\frac{\#NDM_i}{\#RA_i} = \eta_{RA.DS} = (1 - \zeta_{CR}) \cdot (\zeta_{[NC]} \cdot (1 - MC))$$

Für die Untersuchung mit Wiederholanforderungen Ersatz linker CVA-Graph für eine einzelne Leistungsanforderung durch den vereinfachten Graphen rechts. (Symbolliste siehe Folie).

RA_i, DS_i	Request i accepted. Service provided for request i .
NDM_i	Not detected (delivered) malfunction for accepted request i .
CR_i, DR_i	Overall failure rate.
DM_i, CS_i	Crash at request i . Result delivered for request i .
$\eta_{RA.DS}$	The rate at which accepted requirements are delivered as services.
$\eta_{RA.NDM}$	The rate of accepted requirements that are undetected faults.

1.95 Wiederholung bis Erbringung [RR]



Die erste akzeptierte Anforderung (RA_0) ist mit der Rate $\eta_{RA.DS}$ eine erbrachte Leistung (DS_0) und sonst eine Wiederholanforderung bis Ergebniserbringung (RR).

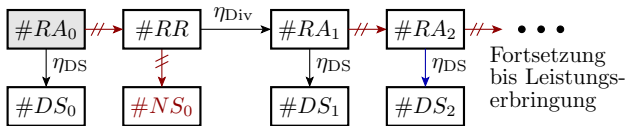
Bei identischem (nicht diversitären) Wiederholergebnis Abbruch NS_0 , weil keine Korrektur möglich. Sonst diversitäre akzeptierte Anforderung (RA_1), die mit $\eta_{RA.DS}$ eine erbrachte Leistung (DS_1) ist.

RA_i, DS_i Request i accepted. Service provided for request i .

NDM_i Not detected (delivered) malfunction for accepted request i .

$\eta_{RA.DS}$ The rate at which accepted requirements are delivered as services.

$\eta_{RA.NDM}$ The rate of accepted requirements that are undetected faults.

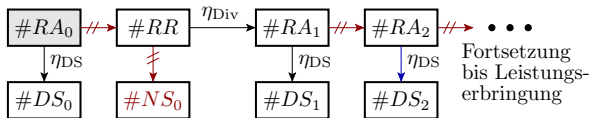


Bei Nichterbringung weitere (diversitäre) Wiederholversuche bis Erbringung. Akzeptierte Anforderung (RA_0) und alle diversitären Wiederholungsergebnisse (RA_i mit $i \geq 1$) sind mit $\eta_{RA.NDM}$ nicht erkannte (ausgelieferte) Fehlfunktionen (NDM_i).

Für Wiederholungsversuche ist unterstellt das die Anforderung immer akzeptiert wird, entweder Hardware- und Service-Verfügbarkeit ein oder, das Versuche ohne akzeptierte Anforderung nicht zählen.

RA_i, DS_i	Request i accepted. Service provided for request i e.
NDM_i	Not detected (delivered) malfunction for accepted request i .
$\eta_{RA.DS}$	The rate at which accepted requirements are delivered as services.
$\eta_{RA.NDM}$	The rate of accepted requirements that are undetected faults.

1.97 Erbringungsrate insgesamt



Erbringungsrate ist eins abzüglich der Rate der NS_0 -Zählwerte:

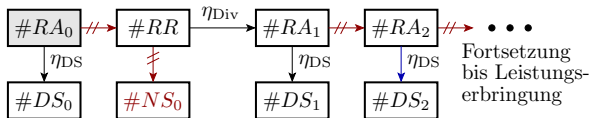
$$\eta_{DS[RR]} = 1 - \left. \frac{\#NS_0}{\#RA_0} \right|_{SLC}$$

$$\eta_{DS[RR]} = 1 - (1 - \eta_{RA.DS}) \cdot (1 - \eta_{Div}) \quad (1.44)$$

Leistungserbringung für alle erkennbaren diversitären Probleme (Abstürze, Fehlfunktionen). Die restlichen Probleme treten bei Wiederholung immer in derseben Weise auf und sind dadurch nicht korrigierbar.

RA_i, DS_i	Request i accepted. Service provided for request i e.
NDM_i	Not detected (delivered) malfunction for accepted request i .
$\eta_{RA.DS}$	The rate at which accepted requirements are delivered as services.
$\eta_{RA.NDM}$	The rate of accepted requirements that are undetected faults.
$\eta_{DS[RR]}$	Delivery rate, in the case of repeated requests, until delivery.

1.98 Zu erwartende Versuchsanzahl



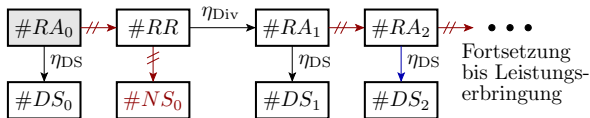
Summe aller $\#RA_i$ Zählwerte je Leistungsanforderung $\#RA_0$:

$$\mu_{Tr1} = \frac{\sum_{i=0}^{\infty} \#RA_i}{\#RA_0} \Big|_{ACR} = 1 + \eta_{Div} \cdot \sum_{i=1}^{\infty} (1 - \eta_{RA.DS})^i \stackrel{(SGS)}{=} 1 - \eta_{Div} + \frac{\eta_{Div}}{\eta_{RA.DS}}$$

Für ideale Diversität (z.B. nur Probleme durch Störungen) Kehrwert der Erbringungsrate je Versuch (vergl. Gl. 1.41). Geringe Erbringungsrate große Versuchsanzahl.

RA_i, DS_i	Request i accepted. Service provided for request i .
NDM_i	Not detected (delivered) malfunction for accepted request i .
$\eta_{RA.DS}$	The rate at which accepted requirements are delivered as services.
$\eta_{RA.NDM}$	The rate of accepted requirements that are undetected faults.
μ_{Tr1}	Expected number of trials.

1.99 Zuverlässigkeit



Rate der nicht erkannten Fehlfunktionen vervielfacht sich gegenüber ohne Wiederholversuche um die zu erwartende Versuchsanzahl:

$$\eta_{\text{NDM}[\text{RR}]} = \frac{\sum_{i=0}^{\infty} \# \text{NDM}_i}{\# \text{RA}_0} \Big|_{\text{SLC}} = \eta_{\text{RA.NDM}} \cdot \underbrace{\left(1 - \eta_{\text{Div}} + \eta_{\text{Div}} \cdot \sum_{i=1}^{\infty} (1 - \eta_{\text{RA.DS}}) \right)}_{\mu_{\text{Tr1}}}$$

Zuverlässigkeit als das Verhältnis aus der Erbringungsrate und der Raten der nicht erkannten Fehlfunktionen ...

$\eta_{\text{MDM}[\text{RR}]}$ Overall rate of undetected malfunctions.

1.100 Zuverlässigkeitsabnahme mit der η_{Tr1}

Die Zuverlässigkeit ist das Verhältnis aus der Erbringungsrate und der Raten der nicht erkannten Fehlfunktionen:

$$R_{[RR]} = \frac{\eta_{DS[RR]}}{\eta_{NDM[RR]}} = \frac{1 - (1 - \eta_{RA.DS}) \cdot (1 - \eta_{Div})}{\eta_{RA.NDM} \cdot \mu_{Tr1}}$$

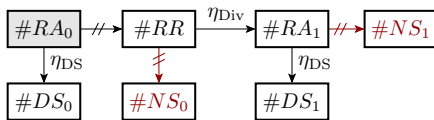
Bei hoher Erbringungsrate $\eta_{NDM[RR]}$ nahe eins ist die Zuverlässigkeit mit Wiederholung etwa der Quotient aus der Zuverlässigkeit ohne Wiederholanforderungen und der zu erwartenden Versuchsanzahl bis zur Erbringung:

$$R_{[RR]} = \frac{R_{[NS]}}{\mu_{Tr1}} \quad (1.45)$$

Mit einer Begrenzung der maximalen Anzahl der Wiederholungen lässt sich die zu erwartende Anzahl der Wiederholungen verringern. Verbessert die Zuverlässigkeit, mindert aber die Erbringungsrate.

$R_{[RR]}$	Reliability, in the case of repeated requests, until delivery.
$\eta_{DS[RR]}$	Delivery rate, in the case of repeated requests, until delivery.
$\eta_{MDM[RR]}$	Overall rate of undetected malfunctions.
$\eta_{RA.DS}$	The rate at which accepted requirements are delivered as services.
$\eta_{RA.NDM}$	The rate of accepted requirements that are undetected faults.
μ_{Tr1}	Expected number of trials.
$R_{[NS]}$	Reliability without request (no service) in case of problems.

1.101 Maximal ein Wiederholversuch [SR]



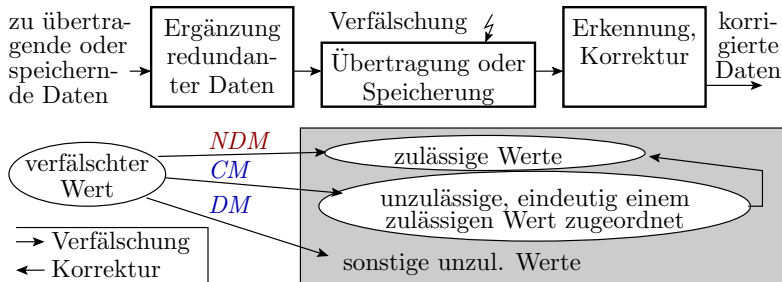
Mit einem Wiederholversuch beträgt die Erbringungsrate:

$$\begin{aligned}
 \eta_{DS[SR]} &= \frac{\#DS_0 + \#DS_1}{\#RA_0} \Big|_{SLC} = \eta_{RA.DS} + (1 - \eta_{RA.DS}) \cdot \eta_{Div} \cdot \eta_{RA.DS} \\
 \eta_{DS[SR]} &= 1 - \underbrace{(1 - \eta_{RA.DS}) \cdot (1 - \eta_{Div})}_{\text{nicht erbrachte nicht diversitäre Leitungen}} - \underbrace{\eta_{Div} \cdot (1 - \eta_{RA.DS})^2}_{0 \text{ for } \eta_{RA.DS} \rightarrow 1} \quad (1.46)
 \end{aligned}$$

Bei hoher Erbringungsrate je Anforderung praktisch auch wie bei Anforderung bis Erbringung (eq. 1.43) Leistungserbringung für alle diversitären Probleme.

$\eta_{DS[SR]}$ Delivery rate in case of a maximum one repeated request.

1.102 Fehlerkorrigierende Codes (Datenverlust)



Korrektur verfälschter Daten nach Übertragung und Speicherung:

- Ergänzung zusätzlicher (redundanter) Bits vor der Übertragung oder Speicherung, mehr als für fehlererkennende Codes.
- Ersatz verfälschter korrigierbarer Werte durch korrekte Wert.

Einzelheiten der Umsetzung siehe später Abschn. 5.3.5.



Automated/manual

1.103 Reaktion auf erkannte Probleme

Automatisierte Reaktion auf Abstürze und erkannte Fehlfunktionen:

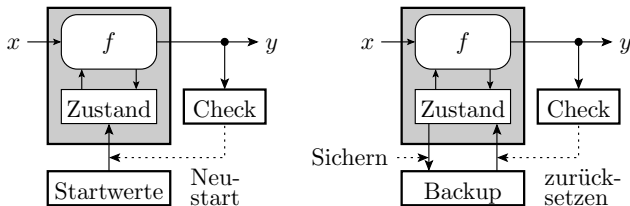
- Abbruch bei Zeitüberschreitung (slide 1.89).
- Sicherer Systemzustand (slide 1.49).
- Fehlermeldung an Nutzer für den händischen Umgang (slide 1.105) und an den Hersteller zur Fehlerbeseitigung (sec.2.3.6).
- Wiederherstellung Funktionsfähigkeit: Bei Vermutung Hardware-Ausfall Reparatur/Rekonfiguration. Neuinitialisierung.
- Korrektur oder Abbruch ohne Leistungserbringung.

Händische Reaktion auf keine automatische Leistungserbringung:

- Verzicht, identische/diversitäre Neuansforderung an dasselbe System oder ein alternatives System.
- Ursachensuche und Beseitigung und Neuansforderung, ...

Erheblicher Anteil an der Funktionalität einer Software und am händische Arbeitsaufwand bei der Nutzung von IT-Service-Leistungen.

1.104 Neuinitialisierung



Bei einer Fehlfunktion werden oft interne Daten verfälscht. Zur Rückkehr in einen funktionsfähigen Zustand sind die internen Daten erneut mit zulässigen Werten zu initialisieren:

- Statische Neuinitialisierung (Reset): fester Anfangszustand,
- Dynamische Neuinitialisierung: Regelmäßiges Backup während des Betriebs. Laden des letzten Backups nach erkannter MF.

Oft werden nur Daten gesichert, die sich nicht problemlos neu berechnen lassen, bei Editoren, Logistiksysteme, Datenbanken, ... die Eingaben seit dem letzten kompletten Backup.

1.105 Händische Fehlerumgehung

Bei Leistungsverweigerung nach Absturz oder erkannter Fehlfunktion mit oder ohne automatische Wiederholversuche probiert ein Nutzer i.allg. händisch, die Aufgabe anders (diversitär) zu lösen. Hinweise liefern die Fehlermeldungen.

Nutzer-Neuanforderung idealerweise bis Erfolg:

- mit unterschiedlichen Eingabedaten,
- unterschiedlicher Art und Reihenfolge der Eingabe,
- Optionen für den Berechnungsweg, die Ausgabe, ...

Ausgenutzt wird die funktionale Redundanz von IT-Systemen (slide 1.80). Für jede Aufgabe gibt es in der Regel viele Lösungswege, von denen die meisten Fehler nur einen Teil beeinträchtigen.

Der Nutzer lernt mit der Zeit, die beobachteten Probleme zu vermeiden oder zu umgehen (siehe später »Reifen des Nutzerverhaltes«, Folie 2.100).

1.106 Problemvermeidung

Vermeidung schwer zu behandelnder, insbesondere sicherheitsgefährdender Probleme:

- Robustheit gegen Störungen, Fehlerisolation,
- Verzicht auf unsichere Beschreibungsmittel,
- Anwendungs- und Arbeitsschutzrichtlinien,
- Sicherheitsvorkehrungen, Trennung vom Internet,
- Ruhestromprinzip, ...

Ruhestromprinzip: Systemgestaltung so, dass sich bei erwarteten Problemen automatisch ein sicherer Zustand einstellt:

- Eisenbahnsignaltechnik: bei fehlendem Ruhestrom Störungsmeldung.
- Brandmeldeanlage: bei Drahtbruch Alarm.
- Fahrzeugbremse: Bremsen, wenn Bremsschlauch platzt, ...



Zusammenfassung

1.107 Probleme während der Nutzung ($\mathbb{L}$)

Iteration aus Überwachung

- Zeitüberwachung auf Abstürze, Watchdog-Prinzip,
- Kontrolle der Formate, d.h. der wertunabhängigen Bestandteile,
- zusätzlich Wertekontrolle.

und Behandlung erkannter Probleme:

- Herstellen eines sicheren (gefährdungsfreien) Zustands,
- Problemdokumentation,
- Neuinitialisierung (dynamisch, statisch),
- wenn erforderlich, Reparatur. Praktisch immer Neuinitialisierung.
- Leistungsverweigerung (NS) oder Wiederholversuch (RR).

Vermeidung schwer zu handhabender Probleme:

- Ruhestromprinzip,
- Trennung vom Internet, ...

Vermeidung von Datenverlust durch fehlerkorrigierende Codes.

1.108 Kennzahlen der Überwachung ($\mathbb{L}$)

$$(1.26) \quad MC = \frac{\#DM}{\#MF} \Big|_{SLC}$$

$$(1.28) \quad \zeta_{PM} = \frac{\#PM}{\#CS} \Big|_{SLC}$$

Ideale Formkontrolle mit r redundanten Bits:

$$(1.32) \quad \mu_{MC} = 1 - 2^{-r}$$

$$(1.34) \quad \zeta_{Phan} = 0$$

Andere Formatkontrollen (Wertebereichstests, Syntaxtests, ...) schlecht modellierbar, schlechtere Eigenschaften, dennoch oft besseres Aufwand-Nutzen-Verhältnis als Wertekontrollen.

Wertekontrolle durch Verdopplung und Vergleich:

$$(1.36) \quad \mu_{MC} = \eta_{Div}$$

$$(1.37) \quad \zeta_{PM} = \eta_{Div} \cdot \zeta_{[Chk]}$$

Andere Formatkontrollen (Loop-Test, spezifizierter Korrektheitsnachweis) nur für spezielle Systeme geeignet, dafür bessere Eigenschaften.

1.109 Erbringungsrate, ($\mathbb{B}$)

Erbringungsrate ohne Neuanforderung:

$$(1.12) \quad \eta_{\text{DS}} = (1 - \zeta_{\text{CR}}) \cdot (1 - \zeta_{\text{OMF}})$$

$$(1.30) \quad \zeta_{\text{OMF}} = \zeta_{\text{PM}} + \zeta \cdot MC$$

Wiederholung, bis kein Problem mehr erkennbar, korrigiert aller diversitären Probleme:

$$(1.43) \quad \eta_{\text{DS}[\text{RR}]} = 1 - (1 - \eta_{\text{RA.DS}}) \cdot (1 - \eta_{\text{Div}})$$

Bei hoher Erbringungsrate, auch bereits bei Begrenzung auf max. eine Neuanforderung, Korrektur aller diversitären Probleme.

($\mathbb{L}$) Bei geringen Problemraten genügt Aussortieren erkannter Probleme oder max. eine Neuanforderung. Fehlfunktionsbehandlung mit bis zu mehreren Neuanforderungen lohnen nur bei diversitärer Erbringung und Erbringungsraten je Anforderung deutlich unter eins.

1.111 Zuverlässigkeitsverbesserung ($\mathbb{L}$)

Ohne Wiederholanforderung für erkannte Probleme

$$(1.14) \quad R_{[\text{NS}]} = \frac{R_{[\text{NC}]} \cdot (1 - \zeta_{\text{OMF}})}{(1 - MC)}$$

$$(1.30) \quad \zeta_{\text{OMF}} = \zeta_{\text{PM}} + \zeta \cdot MC$$

und geringe Rate beobachtbarer Fehlfunktionen $\zeta_{\text{OMF}} \ll 1$ ist die Zuverlässigkeit die ohne Überwachung geteilt durch die Fehlfunktionsdurchlässigkeit der Überwachung:

$$(1.15) \quad R_{[\text{NS}]} = \frac{R_{[\text{NC}]}}{1 - MC} = \frac{R_{[\text{NC}]}}{MT}$$

Wiederholte Anforderungen für diversitäre Leistungen verringern der Zuverlässigkeit um die zu erwartende Anforderungsanzahl:

$$(1.44) \quad R_{[\text{RR}]} = \frac{R_{[\text{NS}]}}{\mu_{\text{Tr1}}}$$

Mit einer Begrenzung der maximalen Anzahl der Wiederholungen, z.B. auf eine, lässt sich die zu erwartende Anzahl der Wiederholungen verringern. Verbessert die Zuverlässigkeit, mindert aber die Erbringungsrate.

1.112 Diversität ($\mathbb{L}$)

Diversität: Verschiedenartigkeit der Problemwirkung bei mehrfacher Lösung gleicher Aufgaben.

Diversitätsrate: Fehlfunktionsabdeckung durch Verdopplung und Vergleich, Korrekturrate Neuberechnung und Anteil der Fehlfunktionen ohne gemeinsame Ursache.

Mehrfachberechnung und Vergleich mit derselbe Hard- und Software erkennt praktisch alle Fehlfunktionen durch Störungen, aber kaum Fehlfunktionen durch Fehler. Grobtest, ob Fehler oder Störung.

Erweiterung der Diversität auf

- auf Fertigungsfehler und Ausfälle: unterschiedliche Hardware,
- Hardware-Entwurfsfehler: unabhängig entworfene Hardware, ...
- Getrennte Software-Entwürfe aus einer gemeinsamen Spezifikation erzielen etwa bis $\eta_{\text{Div}} \leq 90\%$.

(Händische) Fehlerumgehung basiert auf diversitärer Nutzung.

1.113 Literatur

- [1] J. Hartmann.
Analyse und Verbesserung der probabilistischen Testbarkeit kombinatorischer Schaltungen.
PhD thesis, Diss. Universität des Saarlandes, 1992.
- [2] J.C. Laprie.
Dependable computing and fault tolerance: Concepts and terminology.
In 15th IEEE Int. Symp. on Fault-Tolerant Computing, 1985.
- [3] U. Voges.
Software-Diversität und ihre Modellierung - Software-Fehlertoleranz und ihre Bewertung durch Fehler- und Kostenmodelle.
In Informatik-Fachberichte. Springer, 1989.