Safety-critical systems: Basic definitions

Ákos Horváth

Based on István Majzik's slides
Dept. of Measurement and Information Systems

Introduction

- Safety-critical systems
  - Informal definition: Malfunction may cause injury of people
- Safety-critical computer-based systems
  - E/E/PE: Electrical, electronic, programmable electronic systems
  - Control, protection, or monitoring
  - EUC: Equipment under control

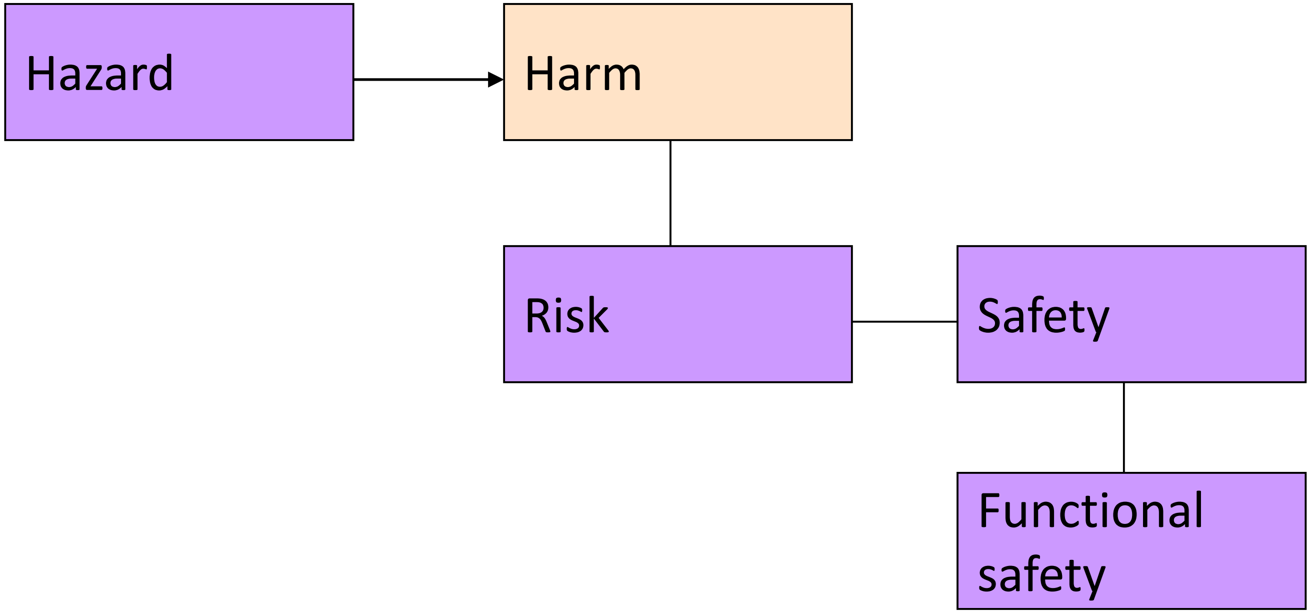
Railway signaling, x-by-wire, interlocking, emergency stopping, engine control, …

Specialities of safety critical systems

- Special solutions to achieve safe operation
  - Design: Requirements, architecture, tools, …
  - Verification, validation, and independent assessment
  - Certification (by safety authorities)
- Basis of certification: Standards
  - IEC 61508: Generic standard (for electrical, electronic or programmable electronic systems)
  - DO178B/C: Software in airborne systems and equipment
  - EN50129: Railway (control systems)
  - EN50128: Railway (software)
  - ISO26262: Automotive
  - Other sector-specific standards: Medical, process control, etc.

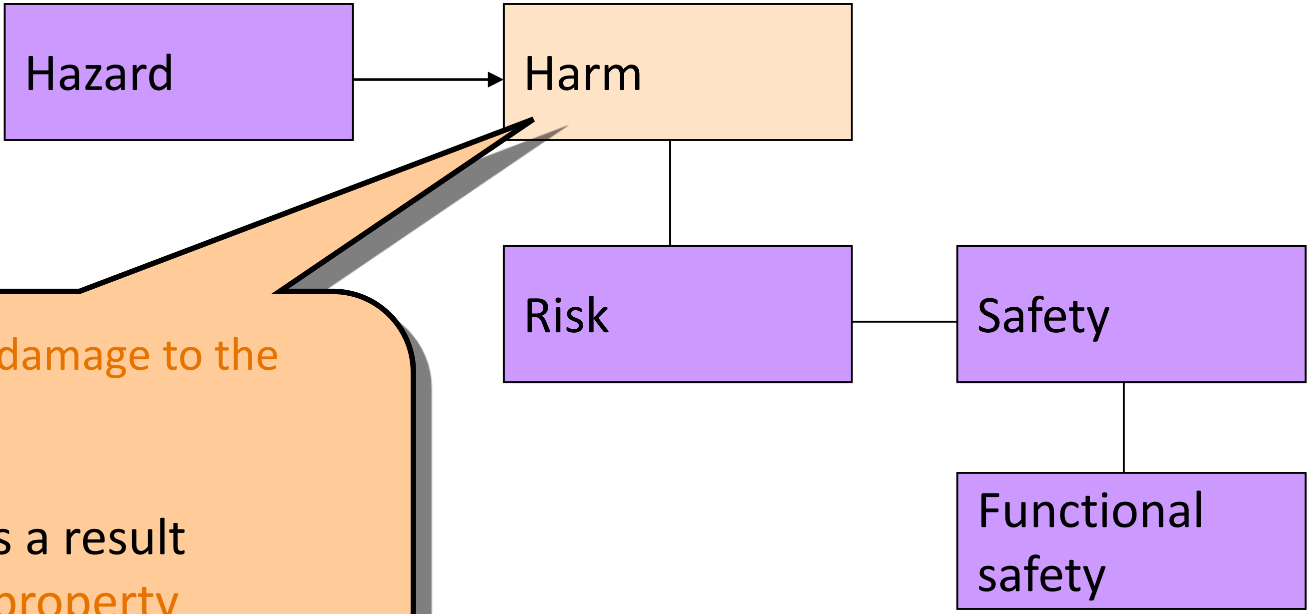
Definition of safety

- Central concepts: Hazard, risk and safety



Definition of safety

- Central concepts: Hazard, risk and safety

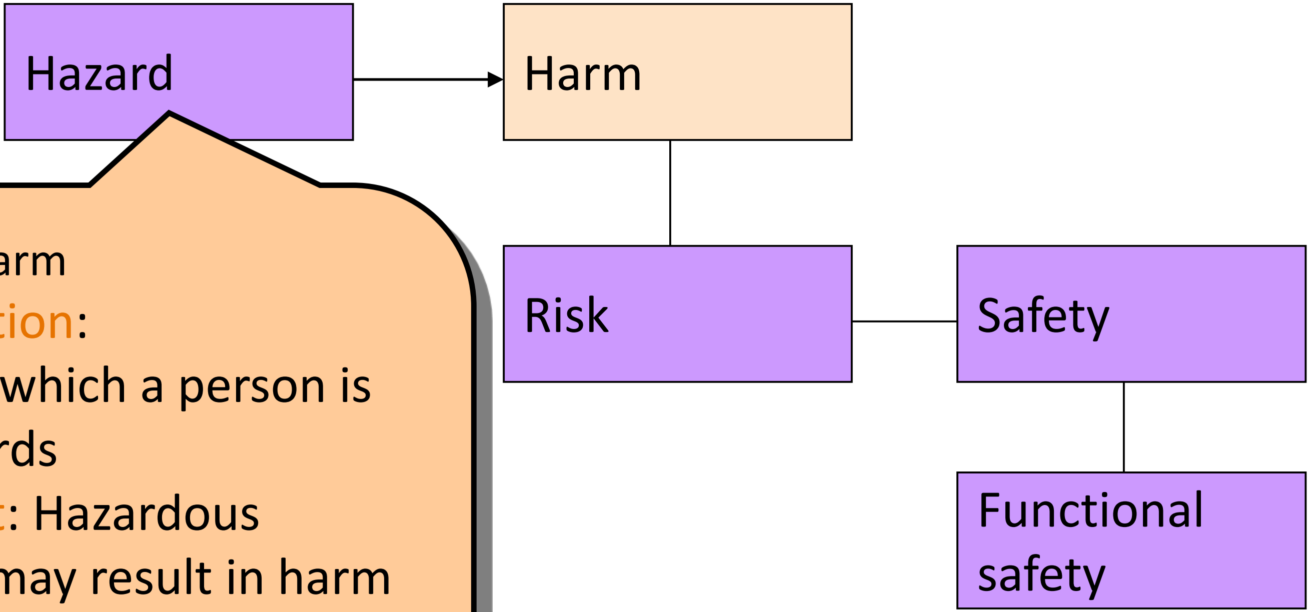


Physical injury or damage to the health of people

- either directly
- or indirectly as a result of damage to property or to the environment

Definition of safety

- Central concepts: Hazard, risk and safety

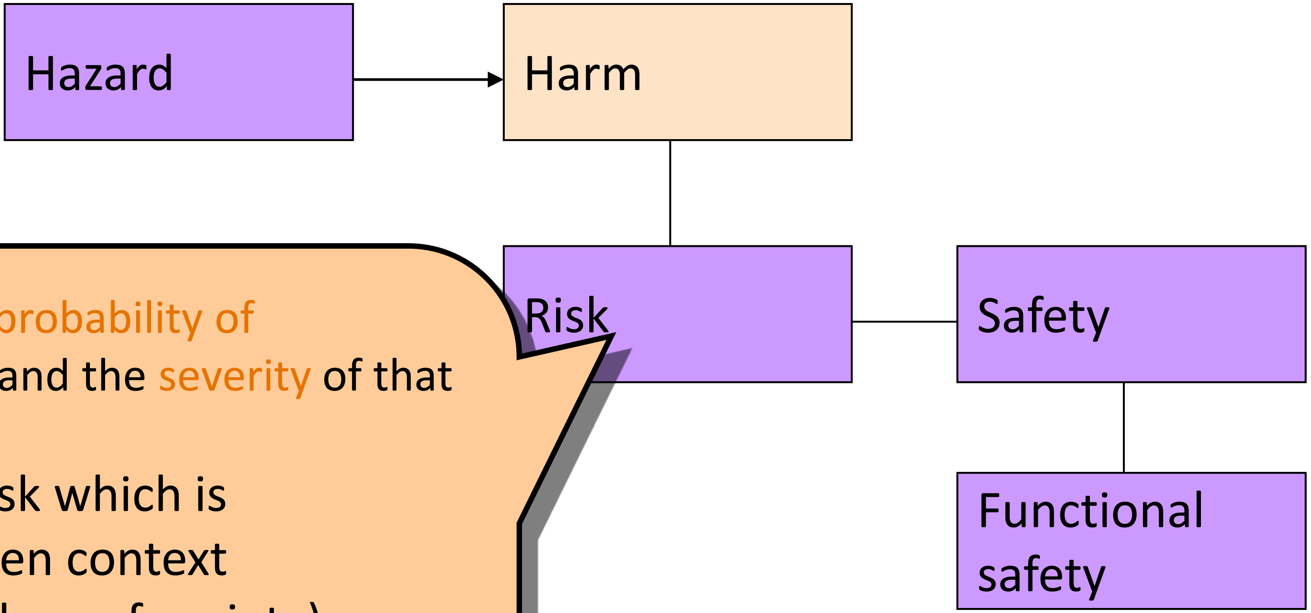


Potential cause of harm

- Hazardous situation: Circumstance in which a person is exposed to hazards
- Hazardous event: Hazardous situation which may result in harm
- Accident: Unintended event that results in harm
- Incident (near miss): Event that has the potential of harm

Definition of safety

- Central concepts: Hazard, risk and safety

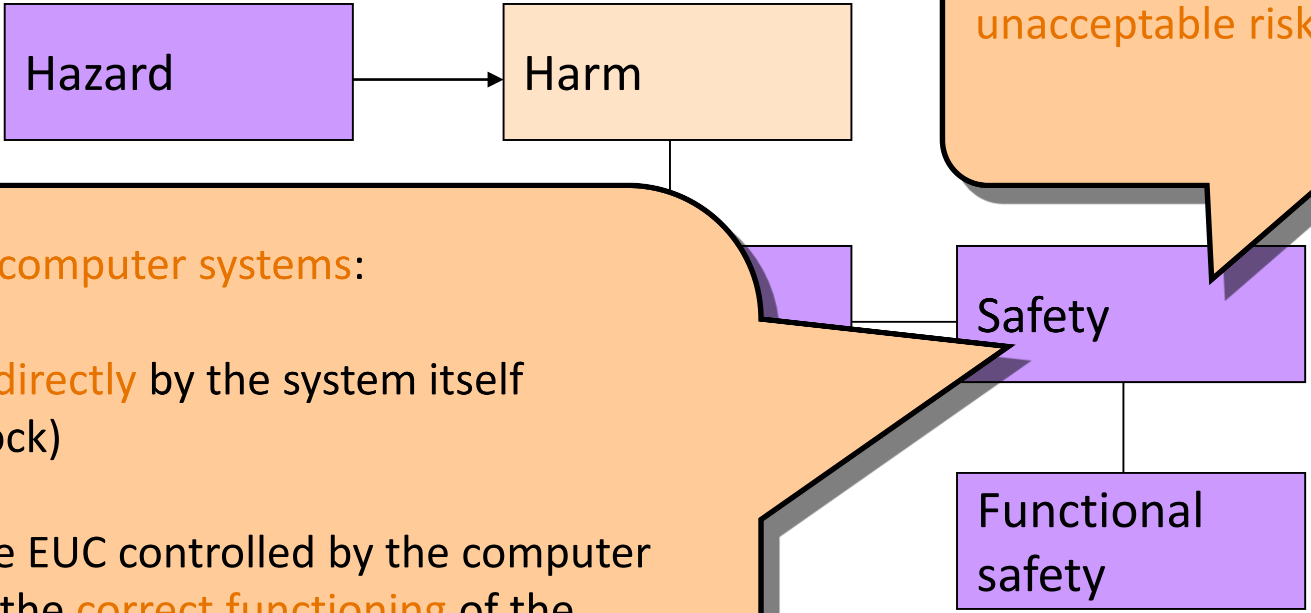


Combination of the probability of occurrence of harm and the severity of that harm

- Tolerable risk: Risk which is accepted in a given context (based on the values of society)
- Residual risk: Risk remaining after protective measures have been taken

Definition of safety

- Central concepts: Hazard, risk and safety



Freedom from unacceptable risk

Forms of safety in computer systems:

Primary safety:

- Dangers caused directly by the system itself (e.g., electric shock)

Functional safety:

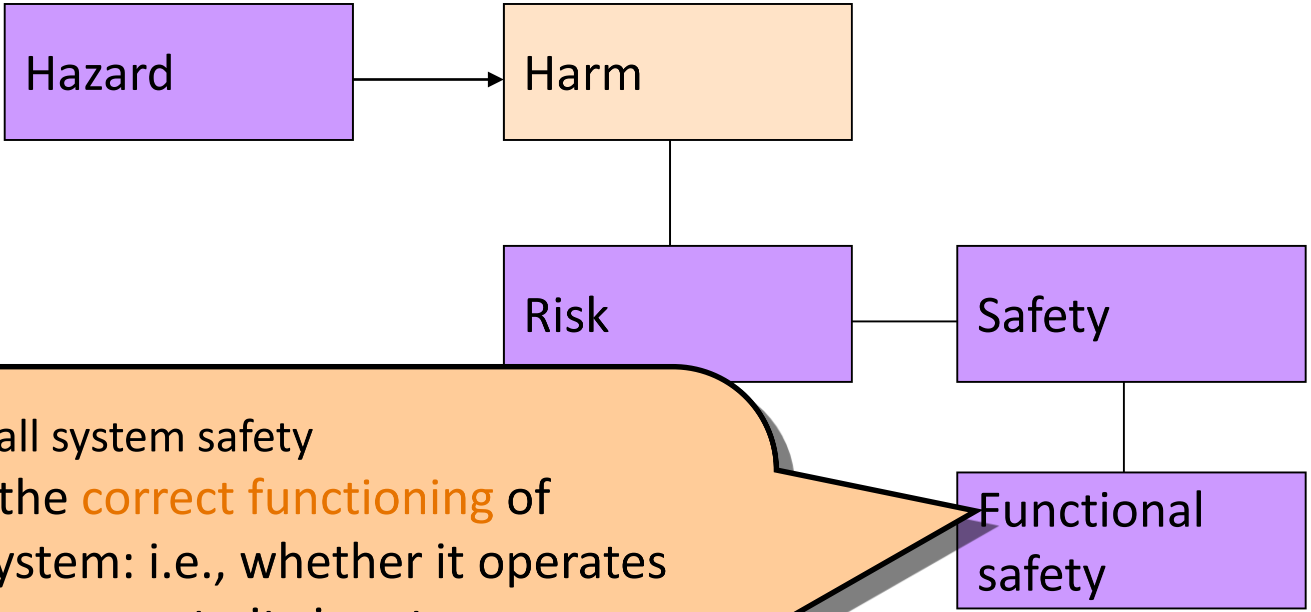
- This concerns the EUC controlled by the computer and is related to the correct functioning of the computer and software.

Indirect safety:

- This relates to the indirect consequences of a computer failure or the production of incorrect information.

Definition of safety

- Central concepts: Hazard, risk and safety

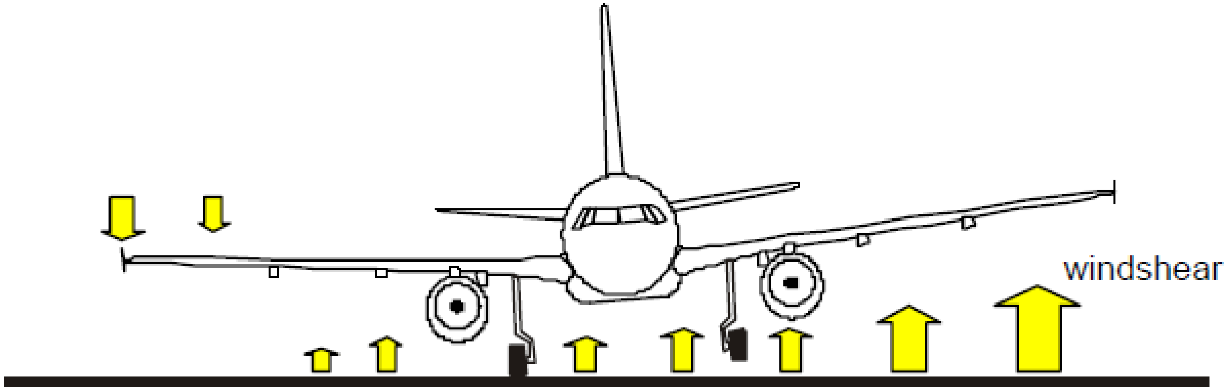


Part of the overall system safety

- depends on the correct functioning of the E/E/PE system: i.e., whether it operates correctly in response to its inputs
- depends on other technology safety-related systems
- depends on external risk reduction facilities

Accident examples

- A320-211 Accident in Warsaw (14 September 1993)
  - Windshear
  - Left gear touched the ground 9 sec later than the right
  - Intelligent braking is controlled by shock absorber + wheel rotation -> delayed braking -> hitting the embankment
- Is the control system ”too intelligent”?
- Correct functioning but not safe behaviour!



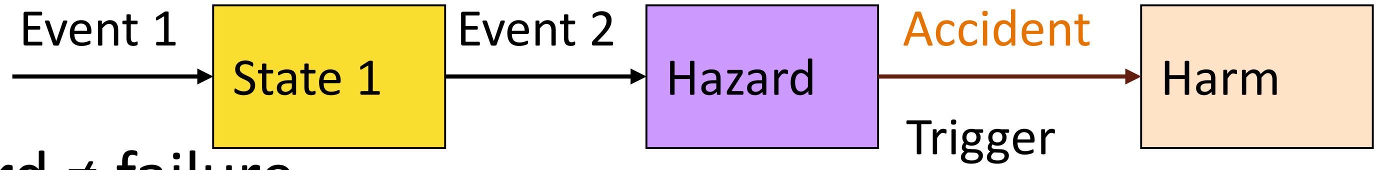
Accident examples

- Toyota car accident in San Diego, August 2009
- Hazard: Stuck accelerator (full power)
  - Floor mat problem

- Hazard control: What about…
  - Braking?
  - Shutting off the engine?
  - Putting the vehicle into neutral? (gearbox: D, P, N)

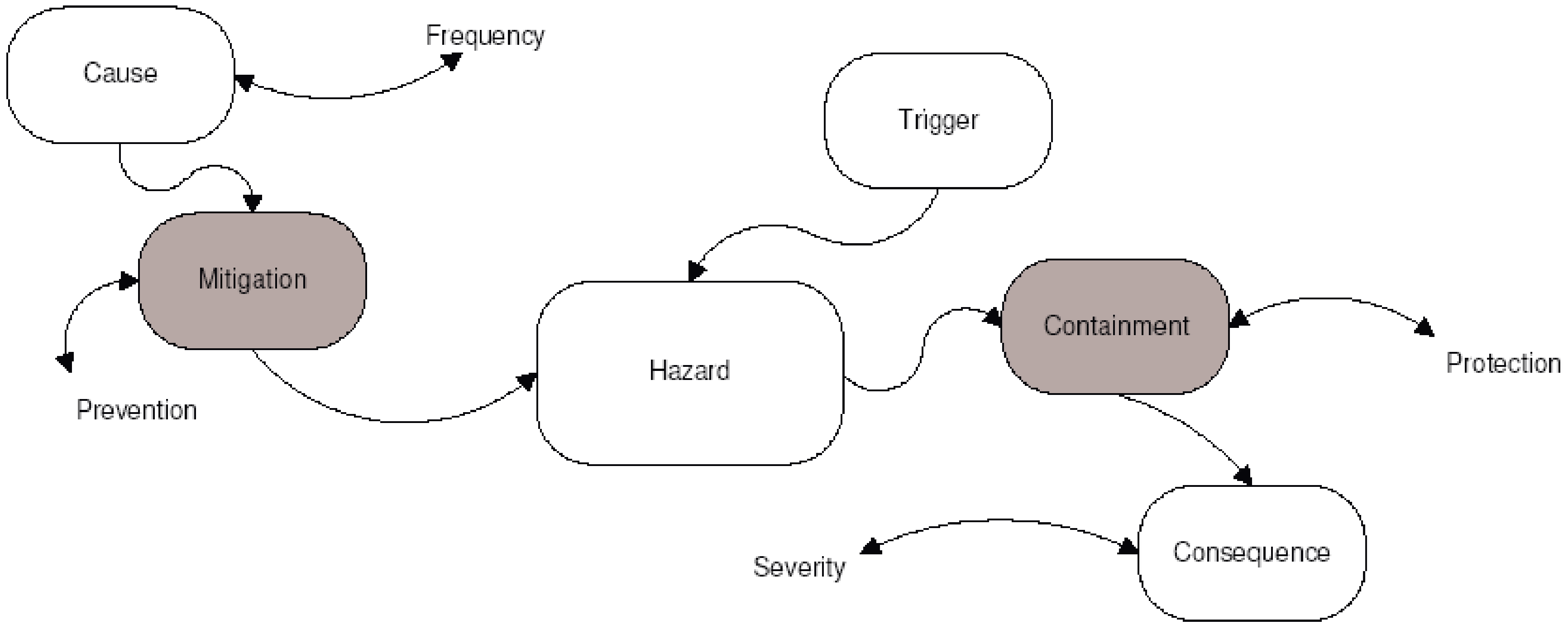
Experiences

- Harm is typically a result of a complex scenario
  - (Temporal) combination of failure(s) and/or normal event(s)
  - Hazards may not result in accidents



- Hazard ≠ failure
  - Undetected (and unhandled) error is a typical cause of hazards
  - Hazard may also be caused by (unexpected) combination of normal events
- Central problems in safety-critical systems:
  - Analysis of hazards
  - Assignment of functions to avoid hazards → accidents → harms

Hazard control



- Risk characteristics:
  - Frequency of occurrence
  - Severity of its consequence
- Mitigation: Eliminate or decrease the chance of a hazard
- Containment: Reduce the consequence of a hazard

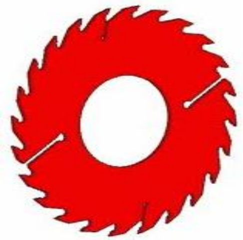
Safety-related system

- Safety function:
  - Function which is intended to achieve or maintain a safe state for the EUC
- Safety-related system:
  - Implements the required safety functions necessary to achieve or maintain a safe state for the EUC,
  - and is intended to achieve the necessary safety integrity for the required safety functions
- Requirements for a safety-related system:
  - What is the safety function: Safety function requirements
  - What is the likelihood of the correct operation of the safety function: Safety integrity requirements

Safety integrity

- Safety integrity:
  - Probability of a safety-related system satisfactorily performing the required safety functions (i.e., without failure)
    - under all stated conditions
    - within a stated period of time
- Types of safety integrity:
  - Random (hardware): Related to random hardware failures
    - Occur at a random time due to degradation mechanisms
  - Systematic: Related to systematic failures
    - Failures related in a deterministic way to faults that can only be eliminated by modification of the design / manufacturing process / operation procedure / documentation / other relevant factors
- Safety integrity level (SIL):
  - Discrete level for specifying safety integrity requirements of the safety functions (i.e., probabilities of failures)

Example: Safety function



- Machine with a rotating blade
  - Blade is protected by a hinged solid cover
- Cleaning of the blade: Lifting of the cover is needed
- Hazard analysis: Avoiding injury of the operator when cleaning the blade
  - If the cover is lifted more than 5 mm then the motor should be stopped
  - The motor should be stopped in less than 1 sec

- Safety function: Interlocking
  - When the cover is lifted to 4 mm, the motor is stopped and braked in 0,8 s
- Safety integrity:
  - The probability of failure of the interlocking (safety function) shall be less than 10^{-4} (one failure in 10.000 operation)
  - Failure of interlocking is not necessarily result in an injury since the operator may be careful

Safety and dependability

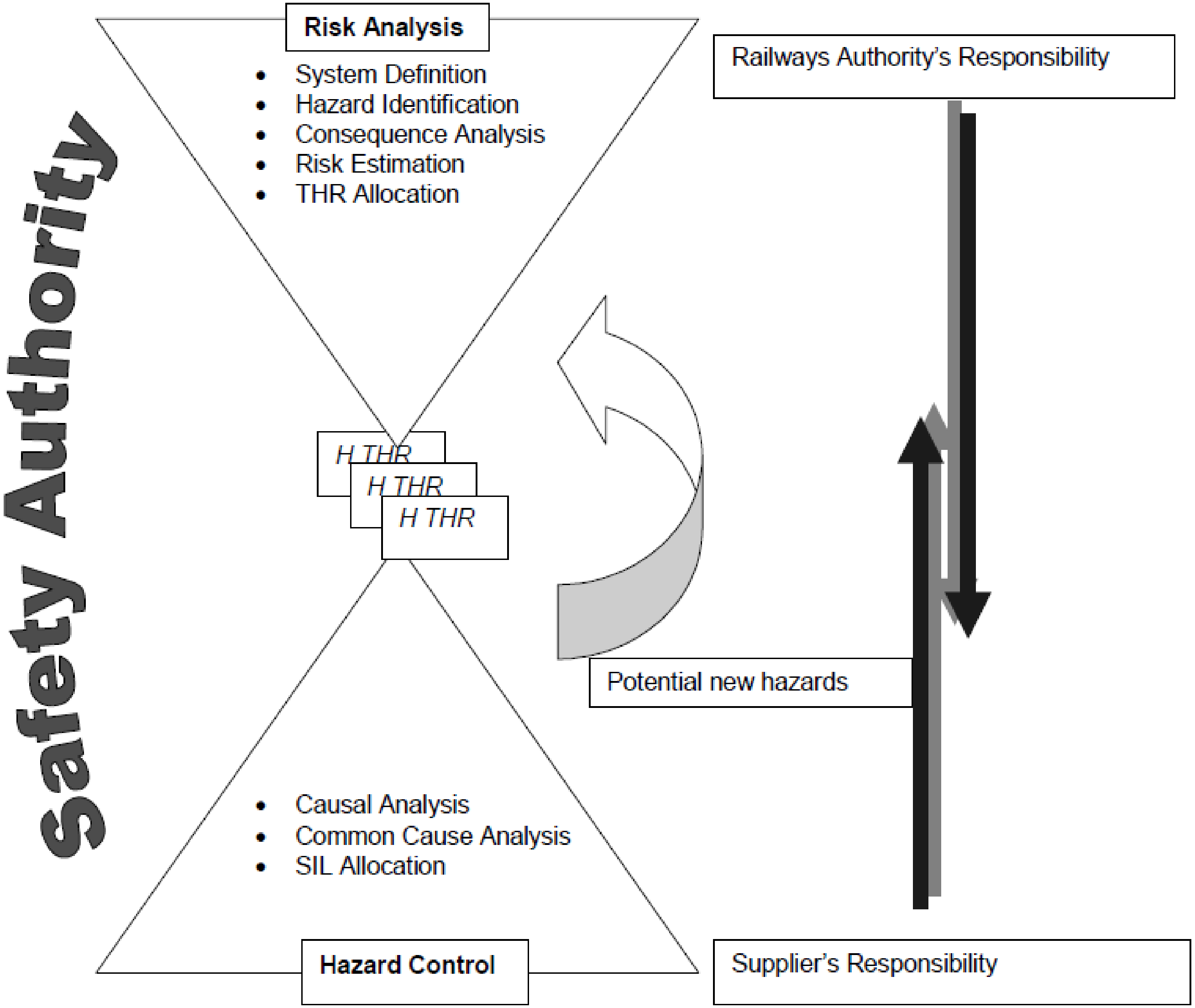
- Safety vs. reliability:
  - Fail-safe state: safe, but 0 reliability
    - Railway signaling, red state: Safety ≠ reliability
    - Airplane control: Safety = reliability
- Safety vs. availability:
  - Fail-stop state: safe, but 0 availability (and reliability)
  - High availability may result in (short) unsafe states

Safety requirements

- Requirements for a safety-related system:
  - Safety function requirements:
    - Derived from hazard identification
  - Safety integrity requirements:
    - Related to target failure measure of the safety function
    - Derived from risk estimation: Acceptable risk
- Safety standards: Risk based approach for determining target failure measure
  - Tolerable risk: Risk which is accepted in a given context based on the current values of society
  - It is the result of risk analysis
    - Performed typically by the customer
    - Considering the environment, scenarios, mode of operation, …

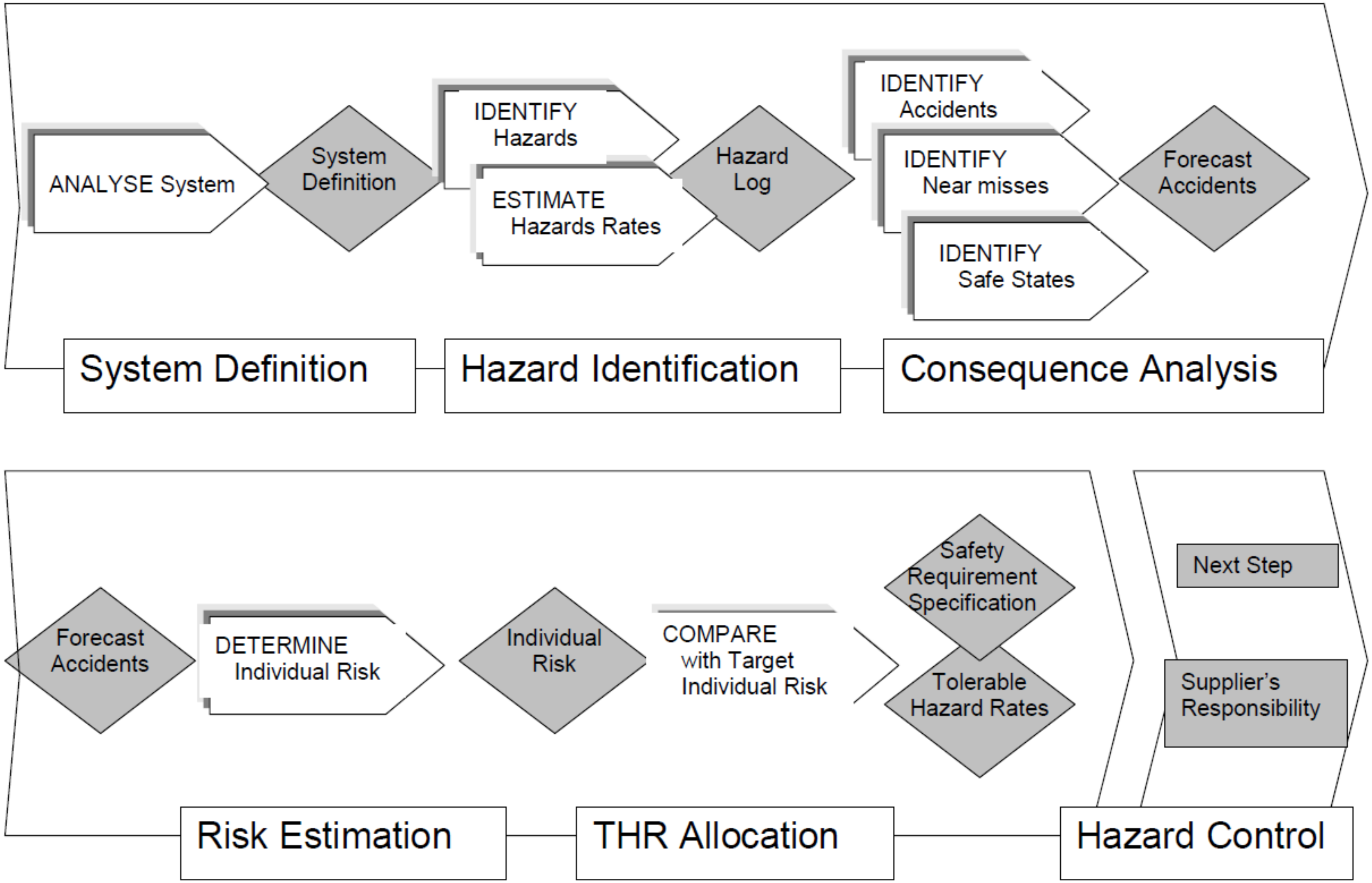
Risk based approach

- EN50129: Railway applications
- THR: Tolerable hazard rate (continuous operation)



Risk analysis

- EN50129 (railway applications)



Mode of operation

- Way in which a safety-related system is to be used:
  - Low demand mode: Frequency of demands for operation is
    - no greater than one per year and
    - no greater than twice the proof-test frequency
  - High demand (or continuous) mode: Frequency of demands for operation is
    - greater than one per year or
    - greater than twice the proof-test frequency
- Target failure measure:
  - Low demand mode: Average probability of failure to perform the desired function on demand
  - High demand mode: Probability of a dangerous failure per hour
    - Acceptable risk -> Tolerable hazard rate (THR)

Safety integrity requirements

- Low demand mode:

SIL	Average probability of failure to perform the function on demand
1	$10^{-2} \leq \text{PFD} < 10^{-1}$
2	$10^{-3} \leq \text{PFD} < 10^{-2}$
3	$10^{-4} \leq \text{PFD} < 10^{-3}$
4	$10^{-5} \leq \text{PFD} < 10^{-4}$

- High demand mode:

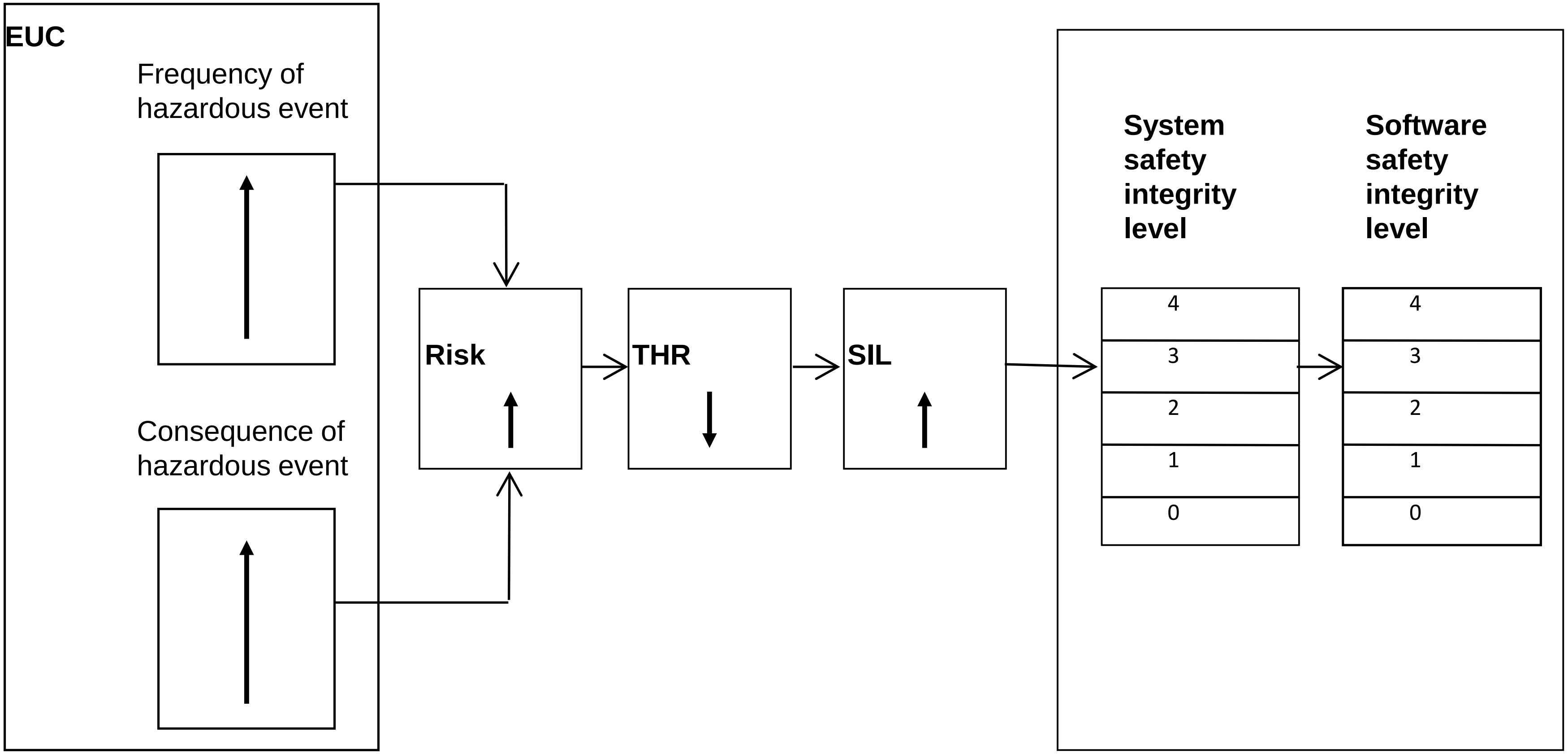
SIL	Probability of dangerous failure per hour per safety function
1	$10^{-6} \leq \text{PFH} < 10^{-5}$
2	$10^{-7} \leq \text{PFH} < 10^{-6}$
3	$10^{-8} \leq \text{PFH} < 10^{-7}$
4	$10^{-9} \leq \text{PFH} < 10^{-8}$

(PFH or THR)

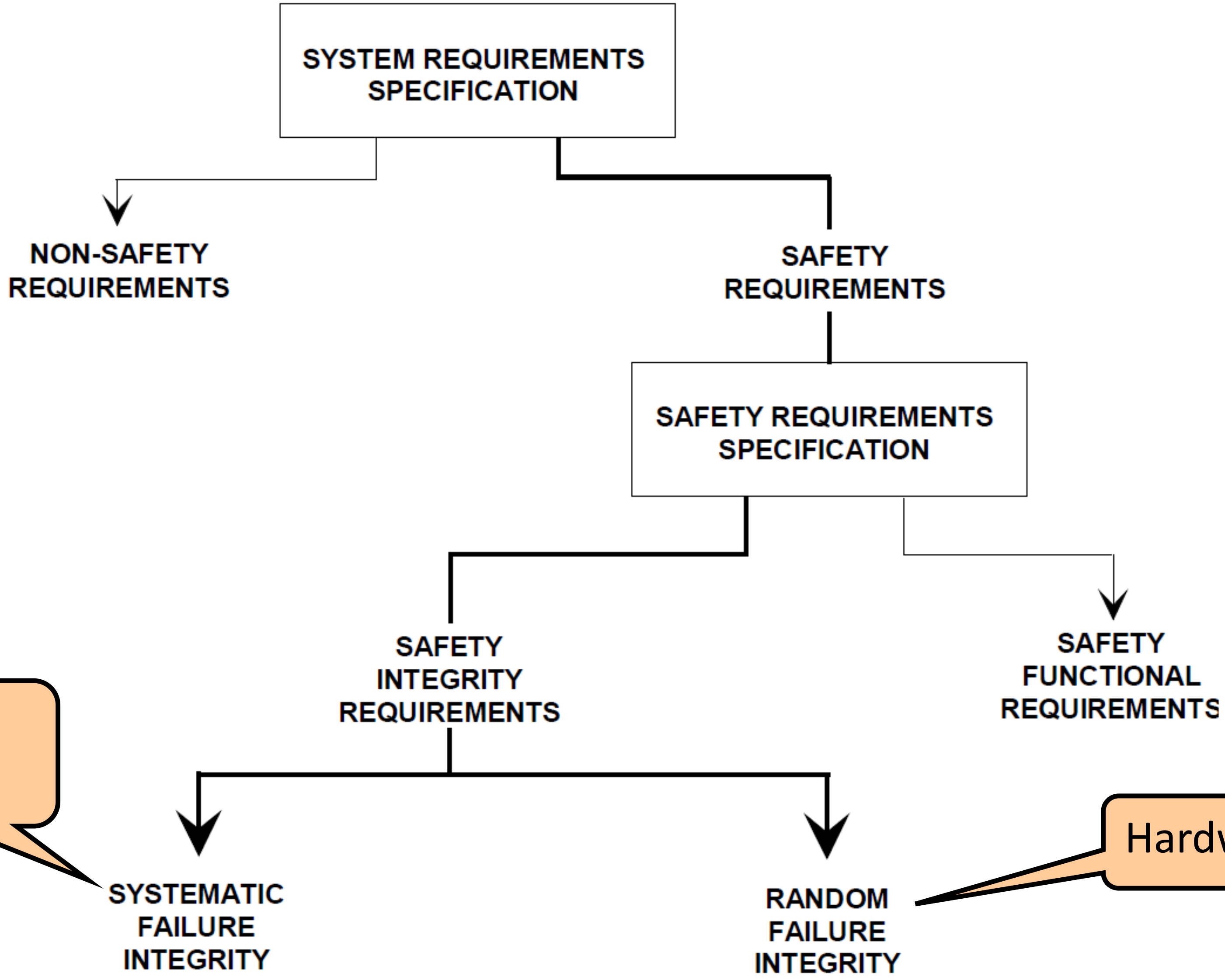
15 years lifetime:
1 failure in case of
750 equipment

Determining SIL: Overview

- Hazard identification and risk analysis -> Target failure measure


EUC
Frequency of hazardous event
Consequence of hazardous event
Risk
THR
SIL
System safety integrity level
4
3
2
1
0
Software safety integrity level
4
3
2
1
0

Structure of requirements


SYSTEM REQUIREMENTS SPECIFICATION
NON-SAFETY REQUIREMENTS
SAFETY REQUIREMENTS
SAFETY REQUIREMENTS SPECIFICATION
SAFETY INTEGRITY REQUIREMENTS
SAFETY FUNCTIONAL REQUIREMENTS
SYSTEMATIC FAILURE INTEGRITY
RANDOM FAILURE INTEGRITY

Typically software

Hardware

Challenges in achieving functional safety

- E/E/PE systems: Complexity
  - Impossible to determine every failure mode
  - Difficult to predict safety performance
- Preventing/controlling dangerous failures resulting from
  - Incorrect specification (system, HW, SW)
  - Omissions in safety requirement specification
  - Hardware failure mechanisms: Random or systematic
  - Software failure mechanisms: Systematic
  - Common cause failures
  - Human (operator) errors
  - Environmental influences (e.g., temperature, EM, mechanical)
  - Supply system disturbances (e.g., power supply)
  - …

Demonstrating SIL requirements

- Approaches:
  - Random failure integrity:
    - Quantitative approach: Based on statistics, experiments
  - Systematic failure integrity:
    - Qualitative approach: Rigor in the engineering
      - Development life cycle
      - Techniques and measures
      - Documentation
      - Independence of persons
- Safety case:
  - Documented demonstration that the product complies with the specified safety requirements
  - Systematic demonstration

Summary of the basic concepts

System safety

- emphasizes building in safety, not adding it to a completed design
- deals with systems as a whole rather than with subsystems or components
- takes a larger view of hazards than just failures
- emphasizes analysis rather than past experience and standards
- emphasizes qualitative rather than quantitative approaches

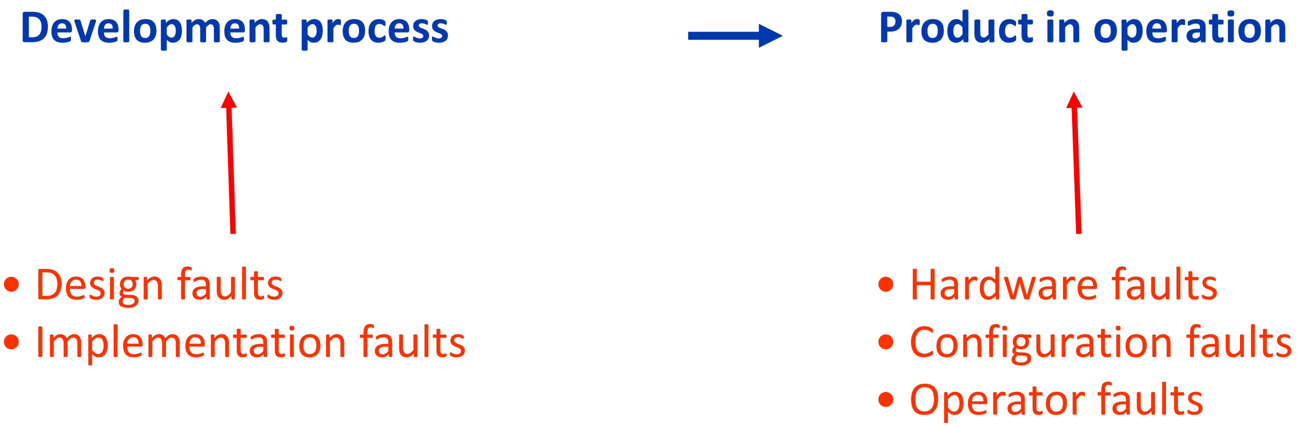
Dependability related requirements

(Safety is not enough)

Characterizing the system services

- Typical characteristics of services:
  - Reliability, availability, integrity, ...
  - These depend on the failures during the use of the services (the good quality of the production process is not enough)
- Composite characteristic: Dependability
  - Definition: Ability to provide service in which reliance can justifiably be placed
    - Justifiably: based on analysis, evaluation, measurements
    - Reliance: the service satisfies the needs
  - Basic question: How to avoid or handle the faults affecting the services?

Fault effects

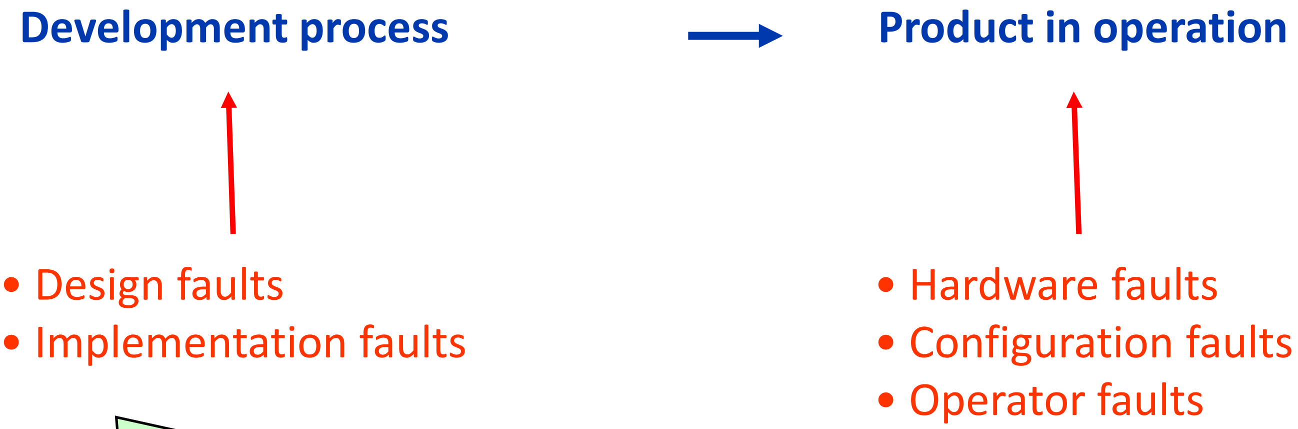


Development process → **Product in operation**

- Design faults
- Implementation faults

- Hardware faults
- Configuration faults
- Operator faults

Fault effects



Development process → **Product in operation**

- Design faults
- Implementation faults

- Hardware faults
- Configuration faults
- Operator faults

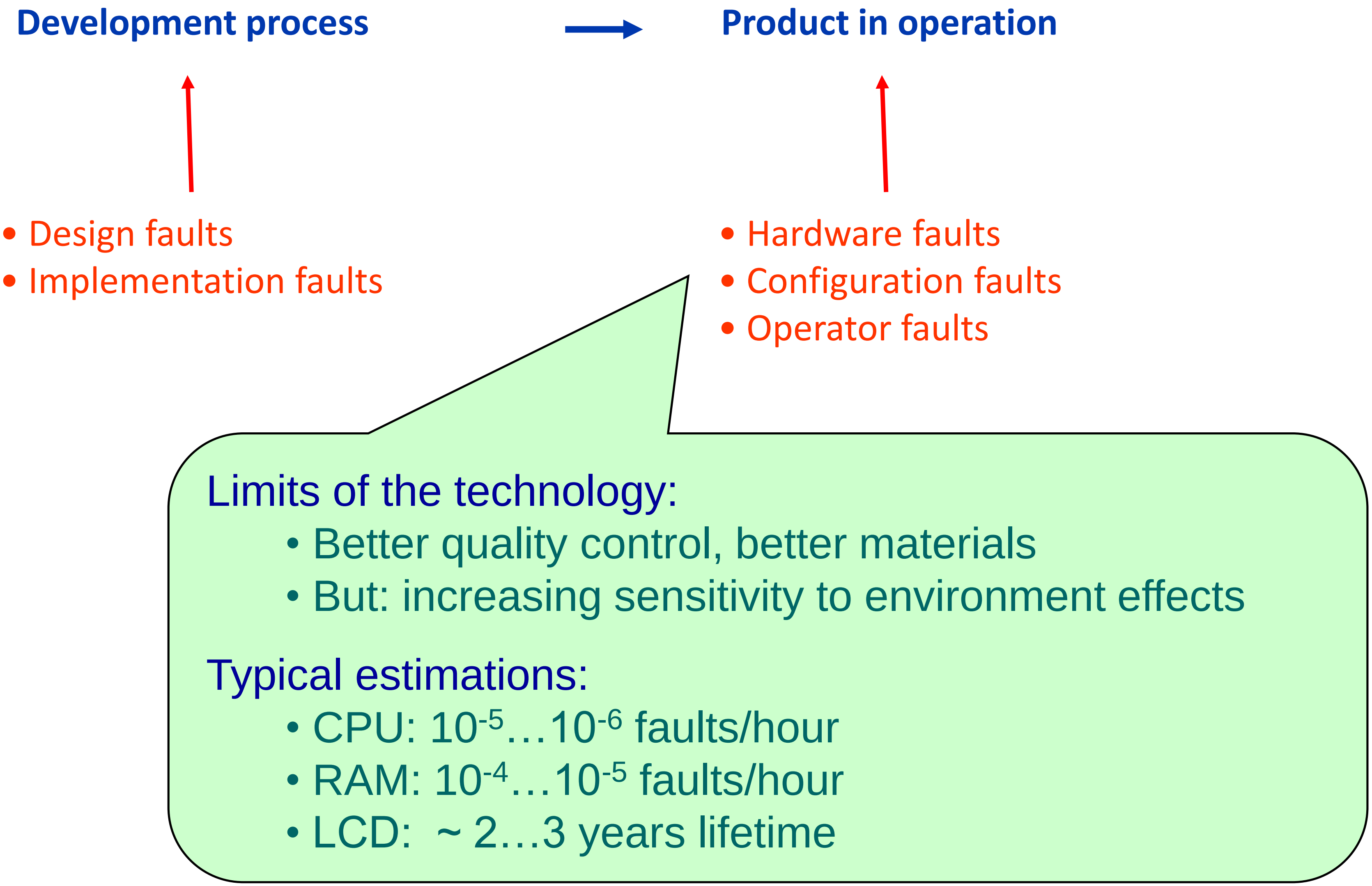
Development process:

- Better quality management, better methodology
- But: Increasing complexity, difficulty in verification

Typical estimations for 1000 lines of code:

- Good development “by hand” : <10 faults
- Tool-supported development: ~1-2 faults
- Application of formal methods: <1 faults

Fault effects



Development process → **Product in operation**

- Design faults
- Implementation faults

- Hardware faults
- Configuration faults
- Operator faults

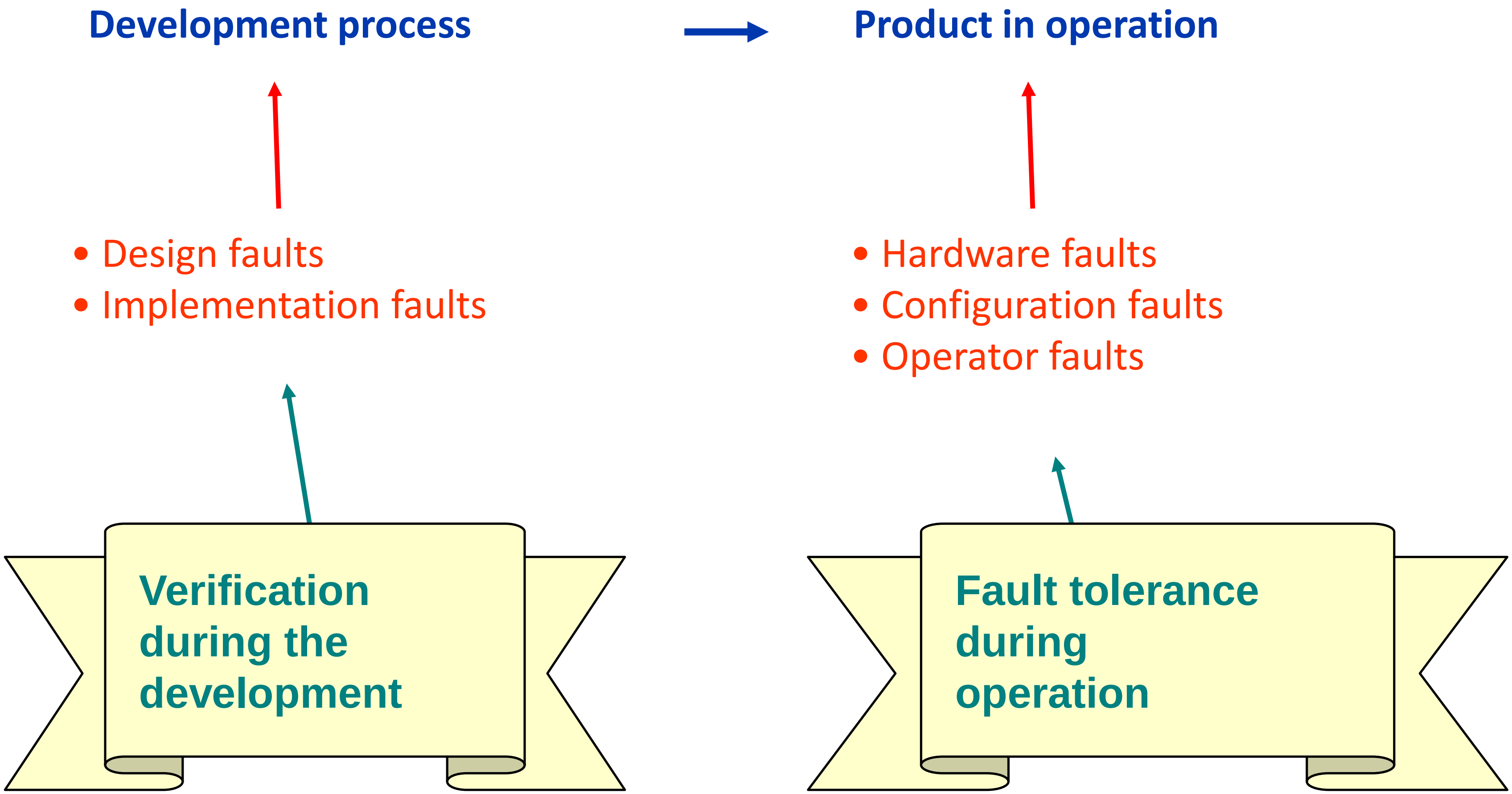
Limits of the technology:

- Better quality control, better materials
- But: increasing sensitivity to environment effects

Typical estimations:

- CPU: 10^{-5}…10^{-6} faults/hour
- RAM: 10^{-4}…10^{-5} faults/hour
- LCD: ~ 2…3 years lifetime

Fault effects



Development process → **Product in operation**

- Design faults
- Implementation faults

- Hardware faults
- Configuration faults
- Operator faults

Verification during the development

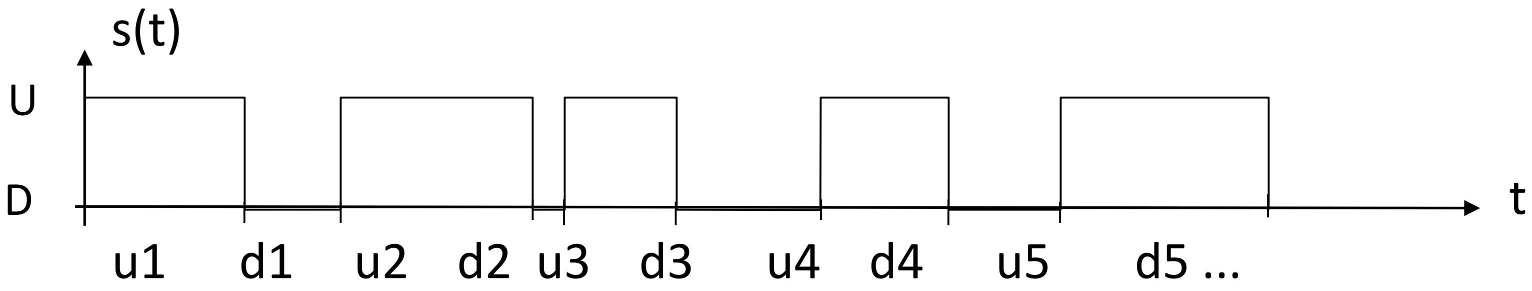
Fault tolerance during operation

Dependability and security

- Basic attributes of dependability:
  - Availability: Probability of correct service (considering repairs and maintenance)
  - Reliability: Probability of continuous correct service (until the first failure)
  - Safety: Freedom from unacceptable risk of harm
  - Integrity: Avoidance of erroneous changes or alterations
  - Maintainability: Possibility of repairs and improvements
- (Attributes of security:)
  - Availability
  - Integrity
  - Confidentiality: absence of unauthorized disclosure of information

Dependability metrics: Mean values

- Partitioning the state of the system: s(t)
  - Correct (U, up) and incorrect (D, down) state partitions



- Mean values:
  - Mean Time to First Failure: MTFF = E{u1}
  - Mean Up Time: MUT = MTTF = E{ui}

    (Mean Time To Failure)
  - Mean Down Time: MDT = MTTR = E{di}

    (Mean Time To Repair)
  - Mean Time Between Failures: MTBF = MUT + MDT

Dependability metrics: Probability functions

- Availability:

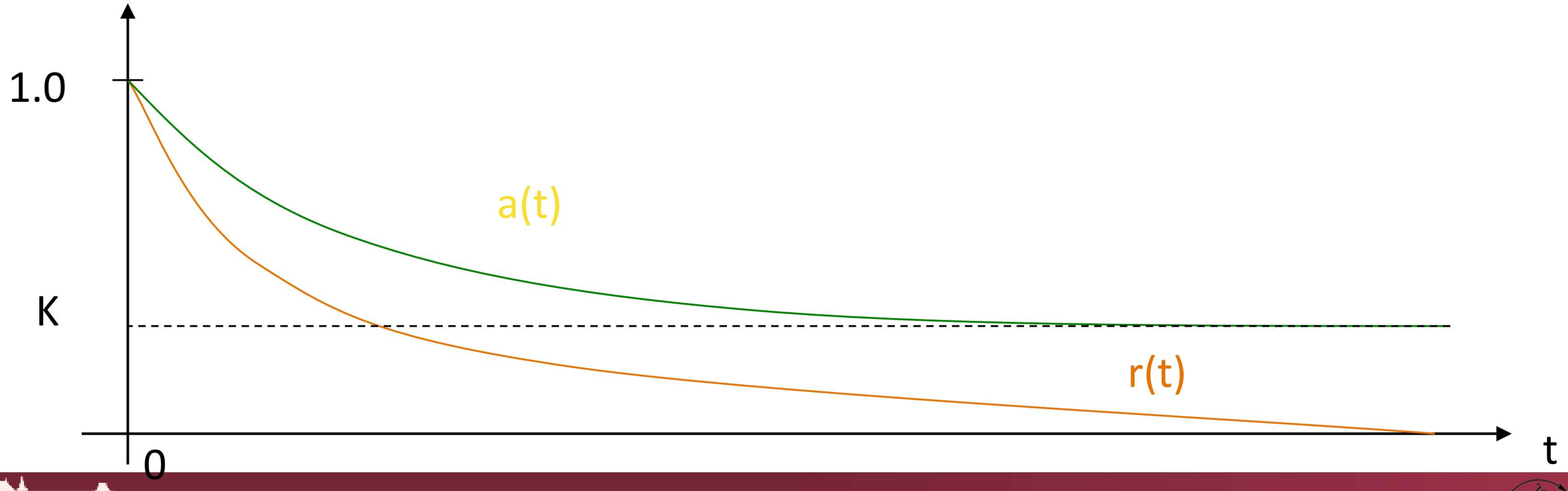
$$a(t) = P\{s(t) \in U\}$$ (failures may occur)

- Reliability:

$$r(t) = P\{s(t') \in U, \forall t' < t\}$$ (no failure until t)

- Asymptotic availability: $K = \lim_{t \to \infty} a(t)$ (regular repairs)

In other way: $K = A = \mathrm{MTTF} / (\mathrm{MTTF} + \mathrm{MTTR})$



Availability related requirements

Availability	Failure period per year
99%	~ 3,5 days
99,9%	~ 9 hours
99,99% („4 nines”)	~ 1 hour
99,999% („5 nines”)	~ 5 minutes
99,9999% („6 nines”)	~ 32 sec
99,99999%	~ 3 sec

Availability of a system built up from components, where the availability of a component is 95%:

- Availability of a system built from 2 components: 90%
- Availability of a system built from 5 components : 77%
- Availability of a system built from 10 components : 60%

Attributes of components

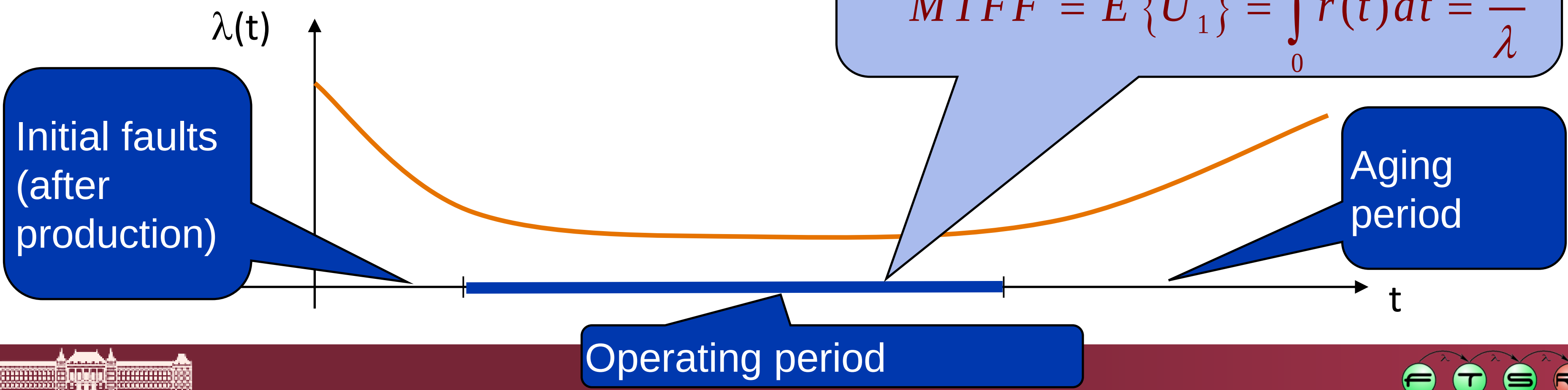
- Fault rate: $\lambda(t)$
  - Probability density that the component will fail at time point t given that it has been correct until t

$$\lambda(t)\Delta t = P\left\{s(t+\Delta t)\in D \mid s(t)\in U\right\} \text{ while } \Delta t \to 0$$

  - In other way (on the basis of the definition of reliability):

$$\lambda(t) = -\frac{1}{r(t)}\frac{dr(t)}{dt}, \quad \text{thus } \; r(t) = e^{-\int_0^t \lambda(t)dt}$$

  - For electronic components:



Here $r(t) = e^{-\lambda t}$

$$MTFF = E\left\{U_1\right\} = \int_0^\infty r(t)dt = \frac{1}{\lambda}$$

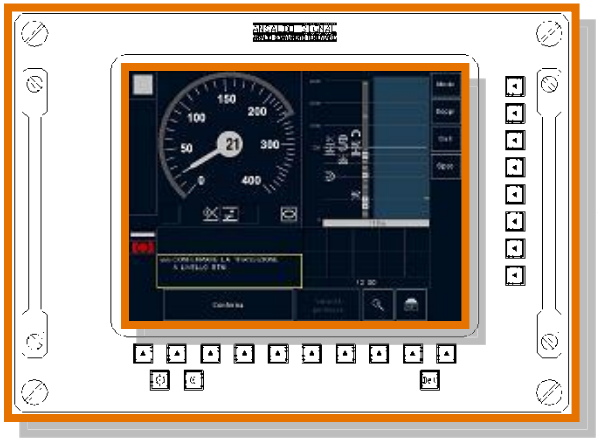
Initial faults (after production)

Operating period

Aging period

Case study: development of a DMI

Driver



DMI

EVC: European Vital Computer (on board)

EVC

Maintenance centre

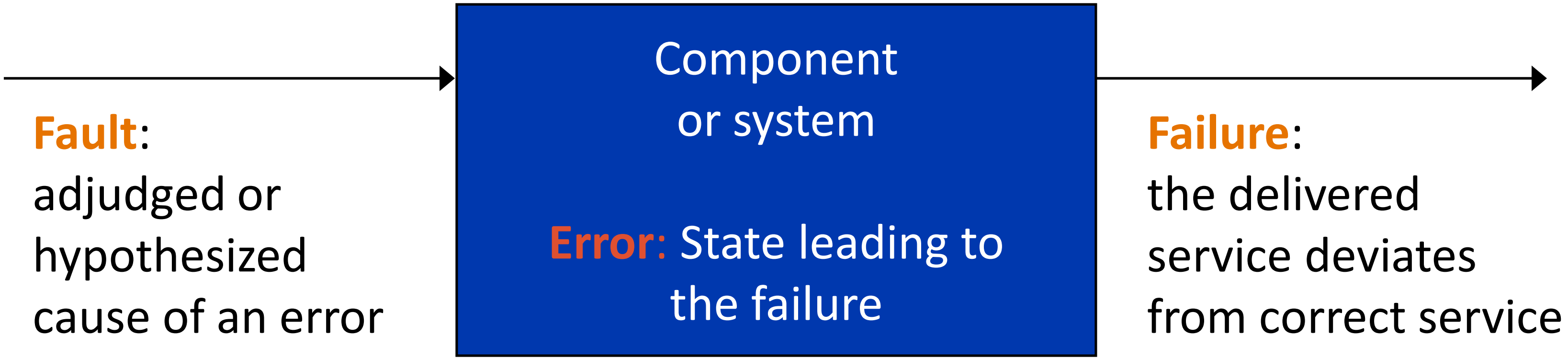
Characteristic:

- Safety-critical functions
  - Information visualization
  - Processing driver commands
  - Data transfer to EVC
- Safe wireless communication
  - System configuration
  - Diagnostics
  - Software update

Case study: DMI requirements

- Safety:
  - Safety Integrity Level: SIL 2
  - Tolerable Hazard Rate: $10^{-7} <= \text{THR} < 10^{-6}$ hazardous failures per hours
  - CENELEC standards: EN 50129 and EN 50128
- Reliability:
  - Mean Time To Failure: MTTF > 5000 hours (5000 hours: ~ 7 months)
- Availability:
  - A = MTTF / (MTTF+MTTR), A > 0.9952
    Faulty state: shall be less than 42 hours per year
    MTTR < 24 hours if MTTF=5000 hours

Threats to dependability



Component or system

Fault: adjudged or hypothesized cause of an error

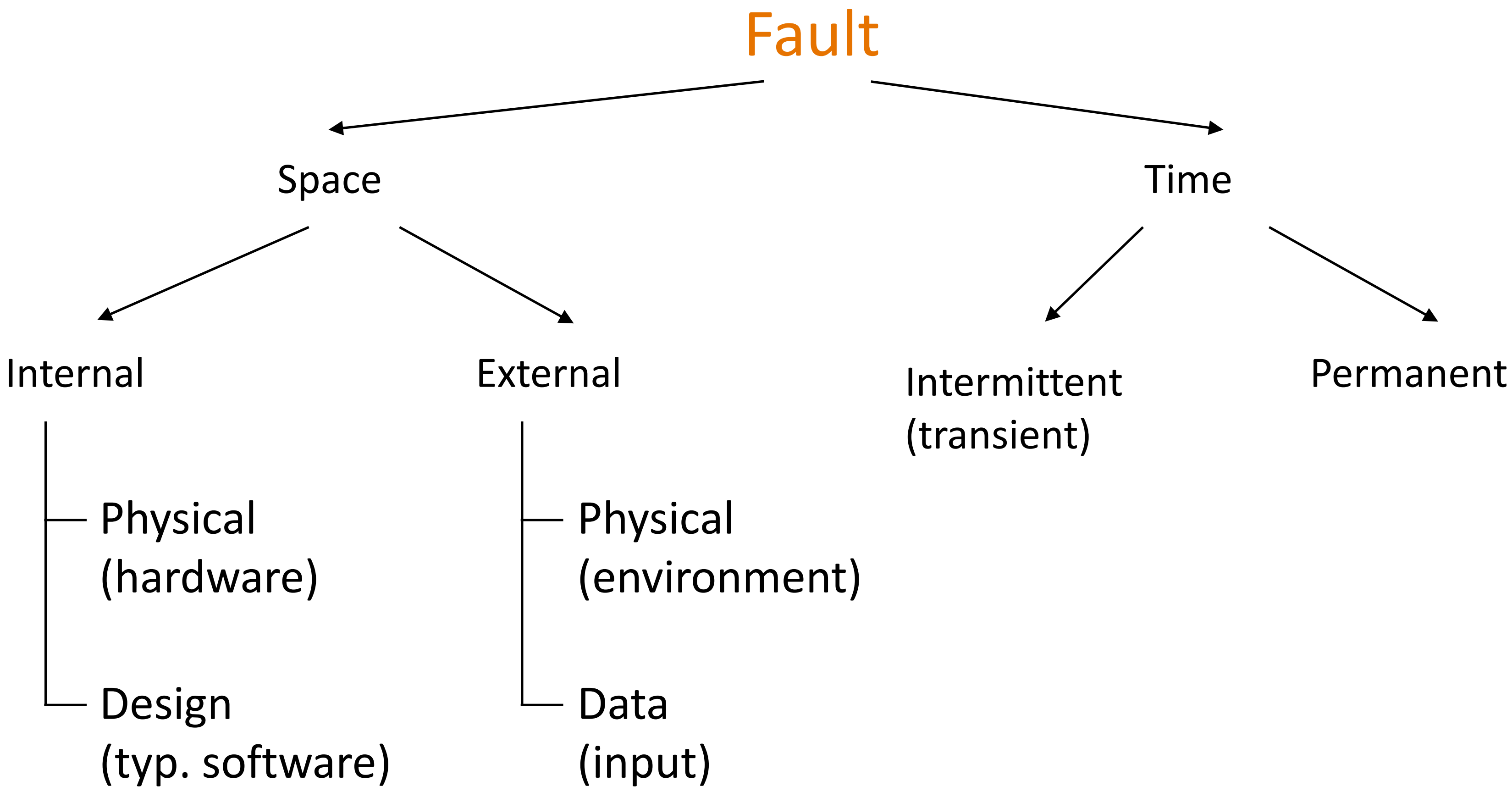
Error: State leading to the failure

Failure: the delivered service deviates from correct service

Fault → Error → Failure examples:

Fault	Error	Failure
Bit flip in the memory due to a cosmic particle →	Reading the faulty memory cell will result in incorrect value →	The robot arm collides with the wall
The programmer increases a variable instead of decreasing →	The faulty statement is executed and the value of the variable will be incorrect →	The final result of the computation will be incorrect

The characteristics of faults



Fault

Space

Time

Internal

External

Intermittent (transient)

Permanent

Physical (hardware)

Design (typ. software)

Physical (environment)

Data (input)

Software fault:

- Permanent design fault (systematic)
- Activation of the fault depends on the operational profile (inputs)

Means to improve dependability

- Fault prevention:
  - Physical faults: Good components, shielding, ...
  - Design faults: Good design methodology
- Fault removal:
  - Design phase: Verification and corrections
  - Prototype phase: Testing, diagnostics, repair
- Fault tolerance: avoiding service failures
  - Operational phase: Fault handling, reconfiguration
- Fault forecasting: estimating faults and their effects
  - Measurements and prediction

    E.g., Self-Monitoring, Analysis and Reporting Technology (SMART)

Summary

- Safety-critical systems
  - Hazard, risk
  - THR and Safety Integrity Level
- Dependability
  - Attributes of dependability
  - Fault -> Error -> Failure chain
  - Means to improve dependability