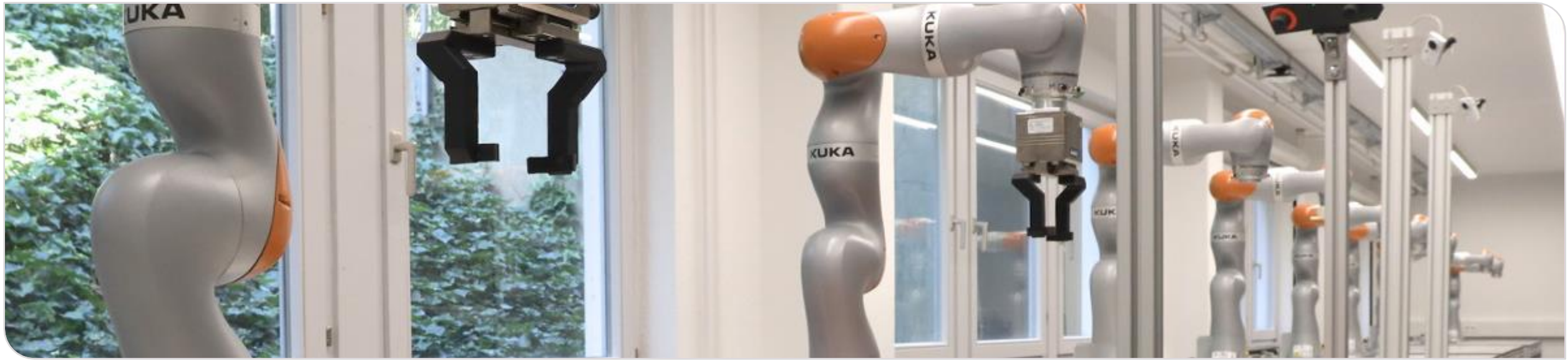
Karlsruher Institut für Technologie

Intelligente Prozess-automation und Robotik

Echtzeitsysteme

Sicherheitskritische Systeme



Wiederholung Regelungstechnik

Vorlesungsevaluation: Regelungstechnik-Vorlesung z.T. schwer verständlich

- Nächste zwei Vorlesungen (19./25.7.): Zeit für Fragen bzw. Wiederholung
- Fragen bzw. Themenwünsche bitte jeweils vorher äußern
  - Im ILIAS-Forum
  - Per Mail an tom.huck@kit.edu

Sicherheitskritische Systeme

en.wikipedia.org/wiki/Piper_Alpha#/media/File:Piper_Alpha_oil_rig_fire.jpg

Ölförderplattform “Piper Alpha”

- 167 Tote
- $3.4 Mrd. Schaden
- Ursache: Explosion an einer Pumpe aufgrund unzureichender Sicherheitsmaßnahmen während Umbau

Sicherheitskritische Systeme

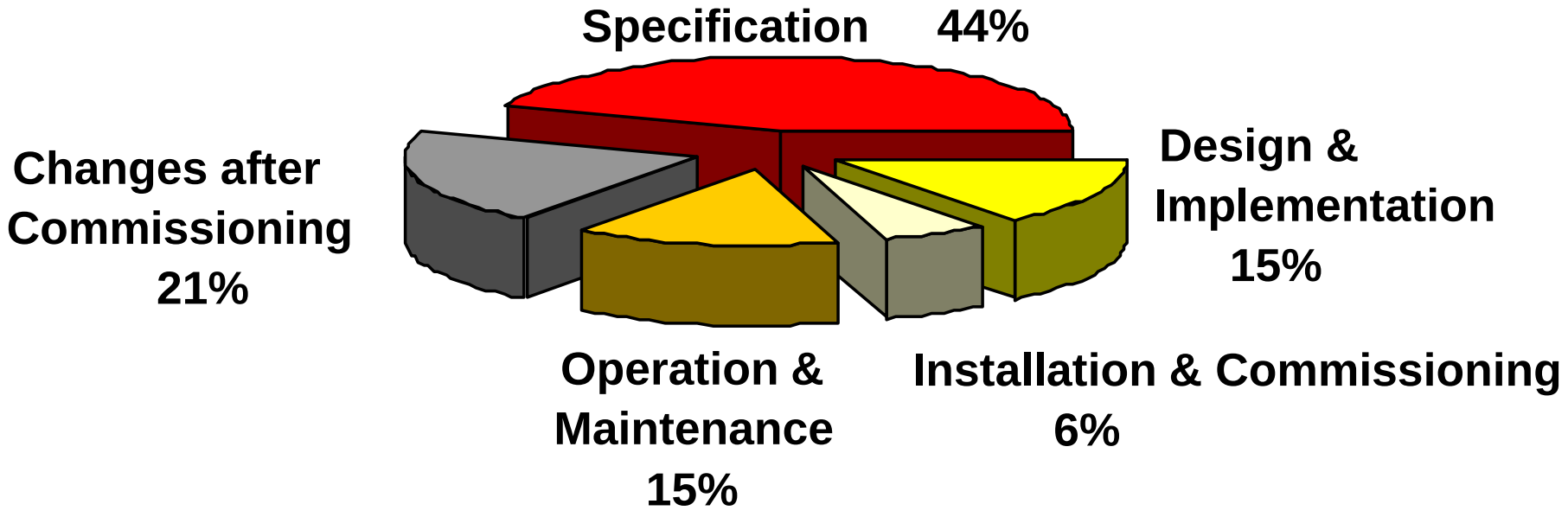
en.wikipedia.org/wiki/Piper_Alpha#/media/File:Piper_Alpha_oil_rig_fire.jpg

Ölförderplattform “Piper Alpha”

- 167 Tote
- $3.4 Mrd. Schaden
- Ursache: Explosion an einer Pumpe aufgrund unzureichender Sicherheitsmaßnahmen während Umbau

Folge: Nachbesserungsbedarf bei Entwicklung von sicherheitskritischen Systemen wurde erkannt.
Ursprung des Sicherheitsstandards „IEC 61508“

Unfallursachen bei industriellen Systemen



Source: “Out of Control: Why Control Systems go Wrong and How to Prevent Failure,” U.K.: Sheffield, Health and Safety Executive

Abgrenzung: Safety vs Security

Safety:

- „Funktionale Sicherheit“
- Schutz des Menschen vor (Fehl-)funktion der Maschine
- Betrachtung von Fehlfunktionen, Ausfällen etc.

Security:

- „IT-Sicherheit“
- Schutz technischer Systeme vor (gezielten) Angriffen
- Betrachtung von Sicherheitslücken, Schwach-stellen, möglichen Angriffspunkten, etc.

Abgrenzung: Safety vs Security

Safety:

- „Funktionale Sicherheit“
- Schutz des Menschen vor (Fehl-)funktion der Maschine
- Betrachtung von Fehlfunktionen, Ausfällen etc.

Betrachtung in dieser Vorlesung

Security:

- „IT-Sicherheit“
- Schutz technischer Systeme vor (gezielten) Angriffen
- Betrachtung von Sicherheitslücken, Schwach-stellen, möglichen Angriffspunkten, etc.

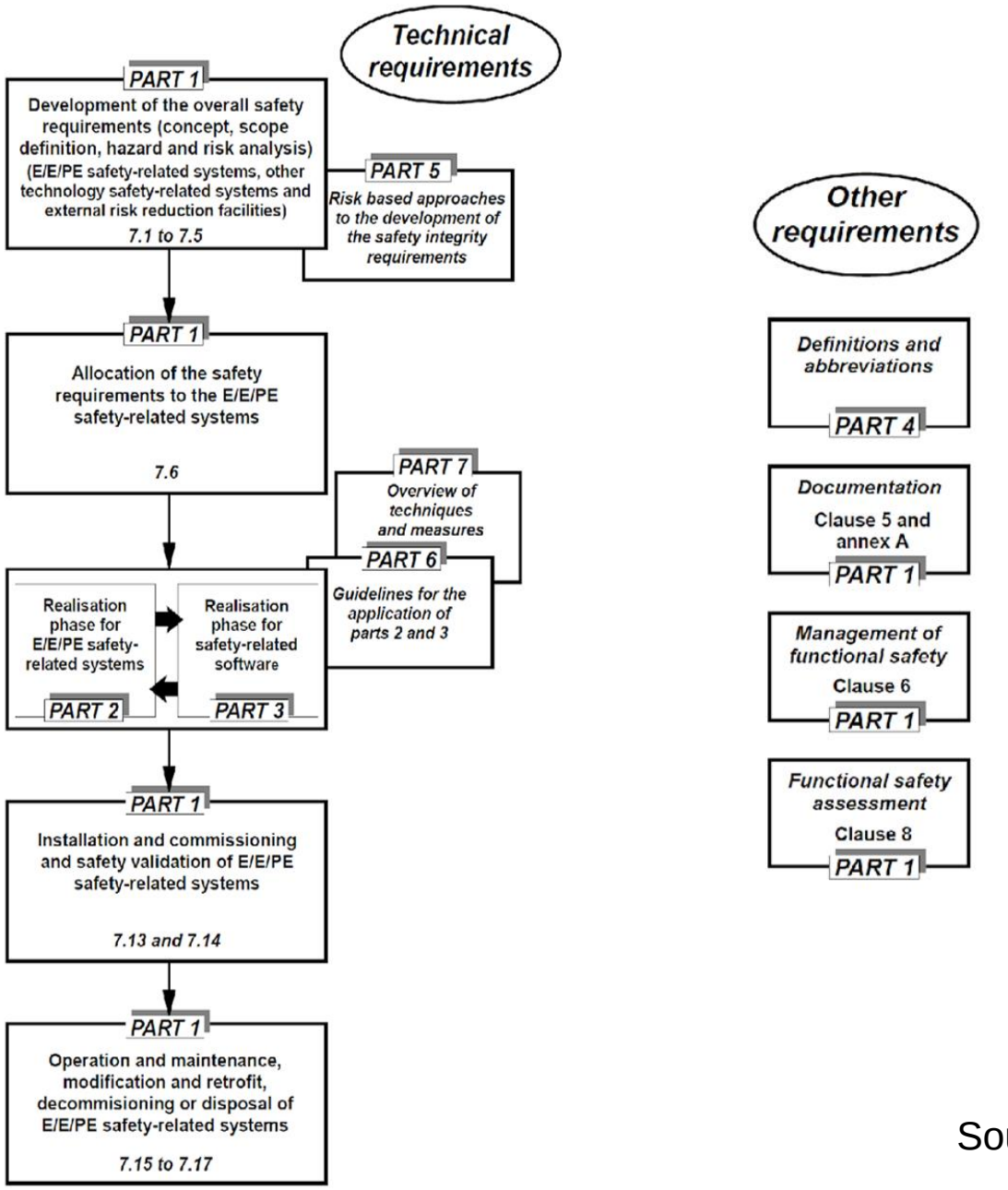
IEC 61508 als Basis für weitere Sicherheitsnormen

IEC 61508 als Basis für weitere Sicherheitsnormen

Aufbau IEC 61508

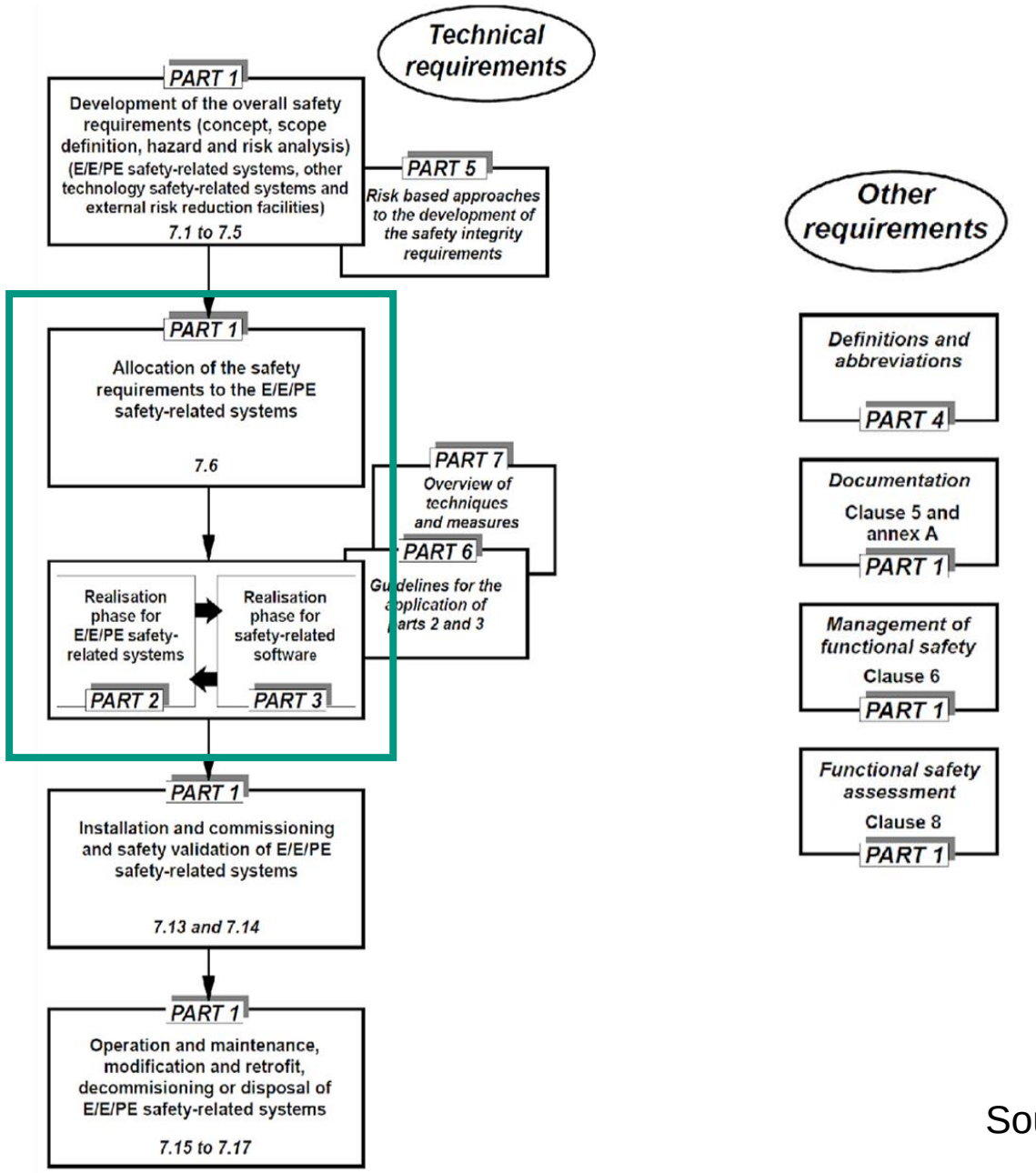
- **Part 0** Functional safety and IEC 61508
- **Part 1** General requirements
- **Part 2** Requirements for electrical/electronic/programmable electronic safety-related systems
- **Part 3** Software requirements
- **Part 4** Definitions and abbreviations
- **Part 5** Examples of methods for the determination of safety integrity levels
- **Part 6** Guidelines on the application of IEC 61508-2 and IEC 61508-3
- **Part 7** Overview of measures and techniques

Aufbau IEC 61508



Source: IEC 61508

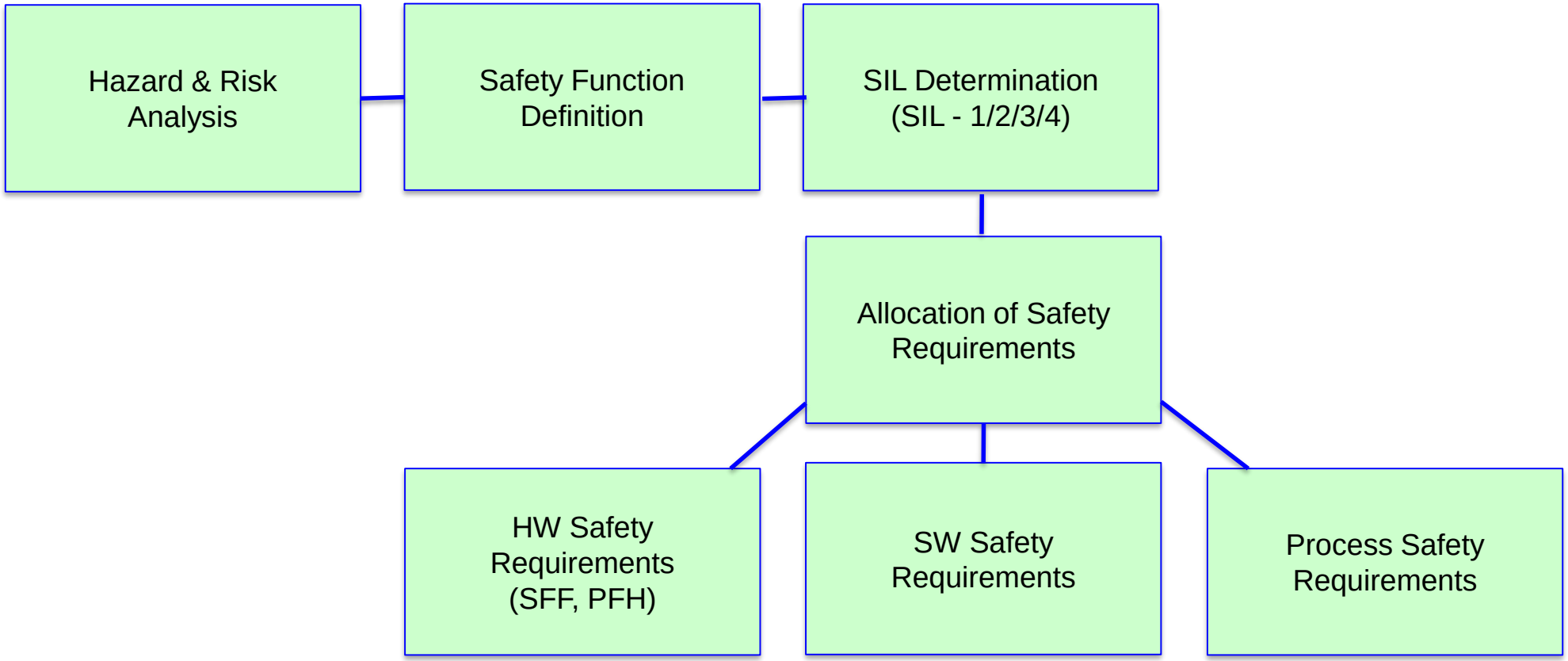
Aufbau IEC 61508



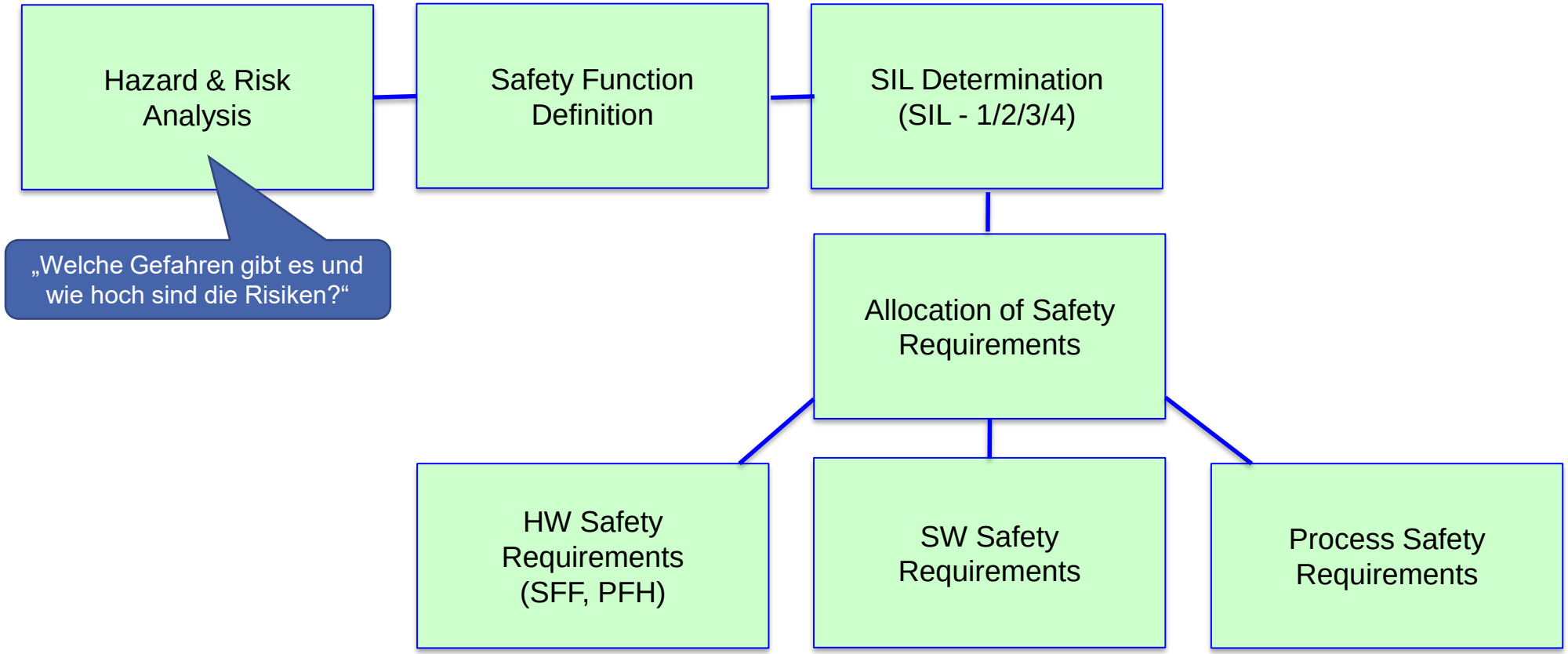
Fokus in dieser Vorlesung

Source: IEC 61508

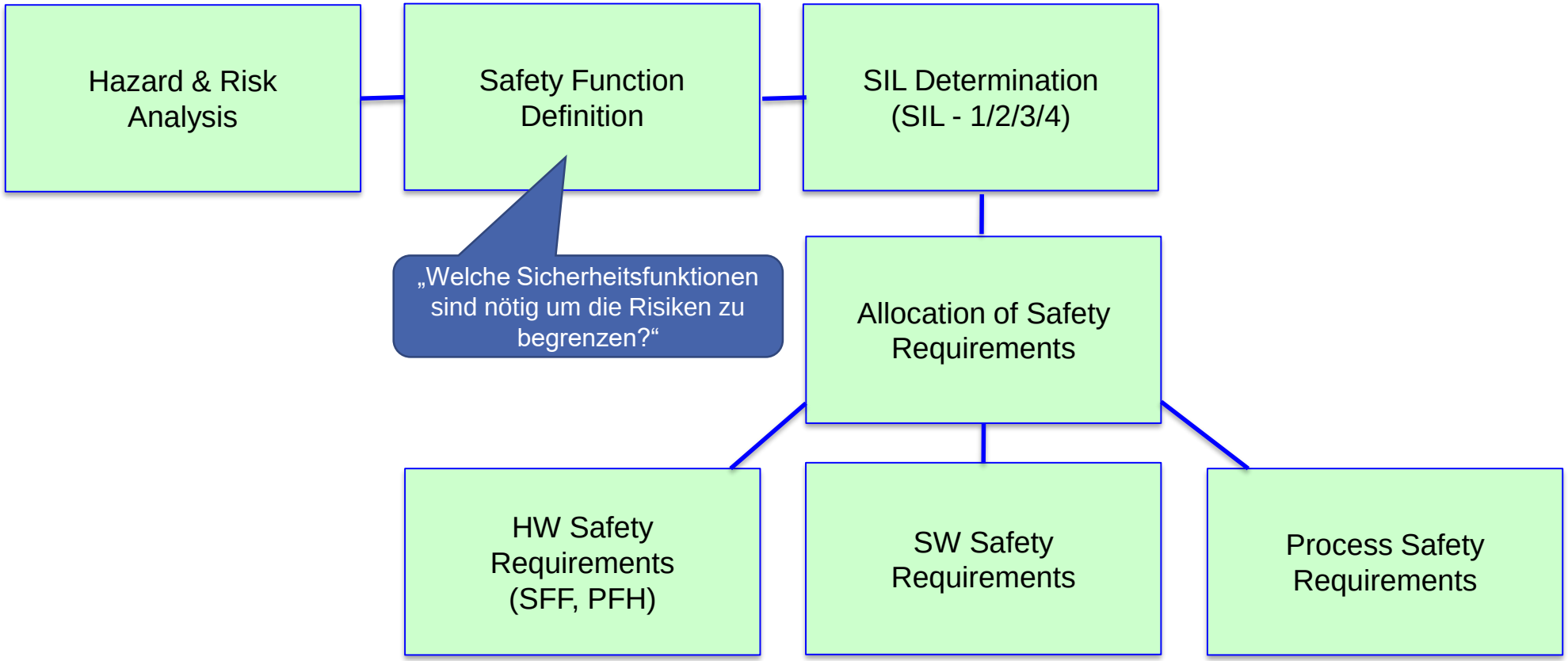
Vorgehensweise nach IEC 61508



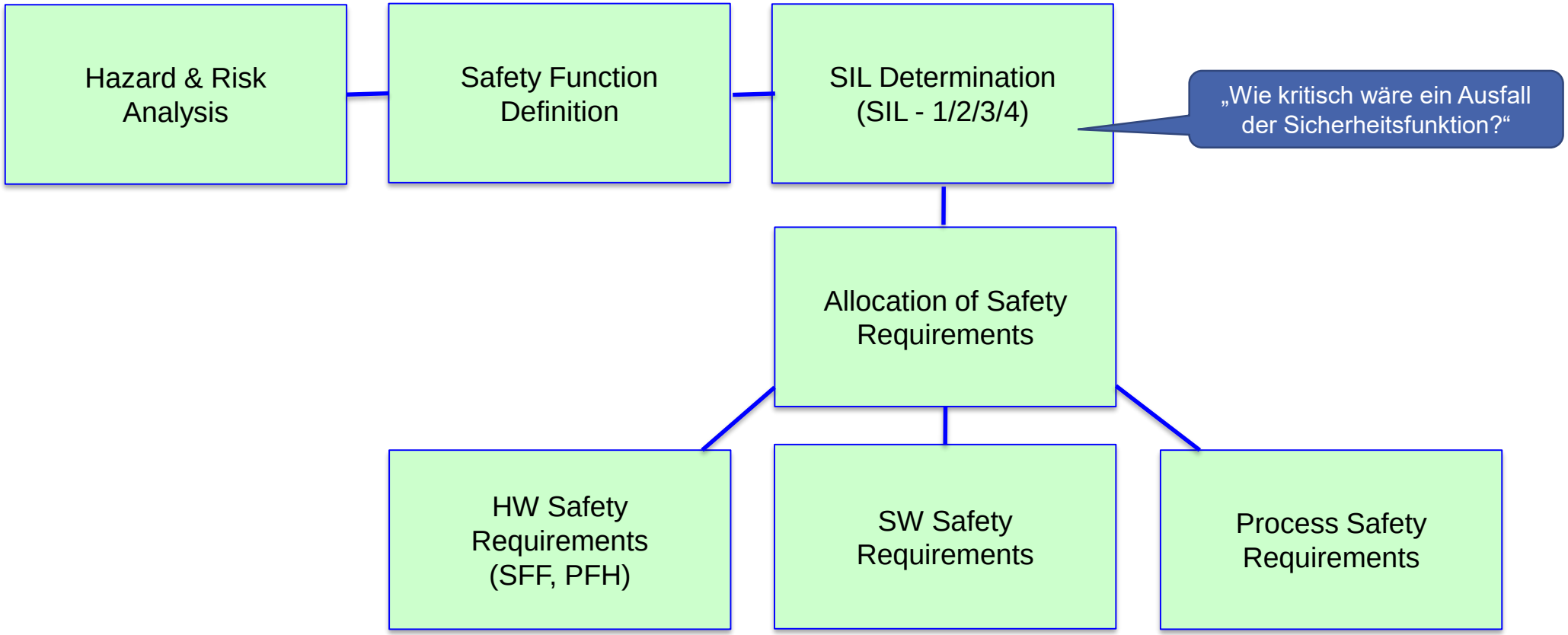
Vorgehensweise nach IEC 61508



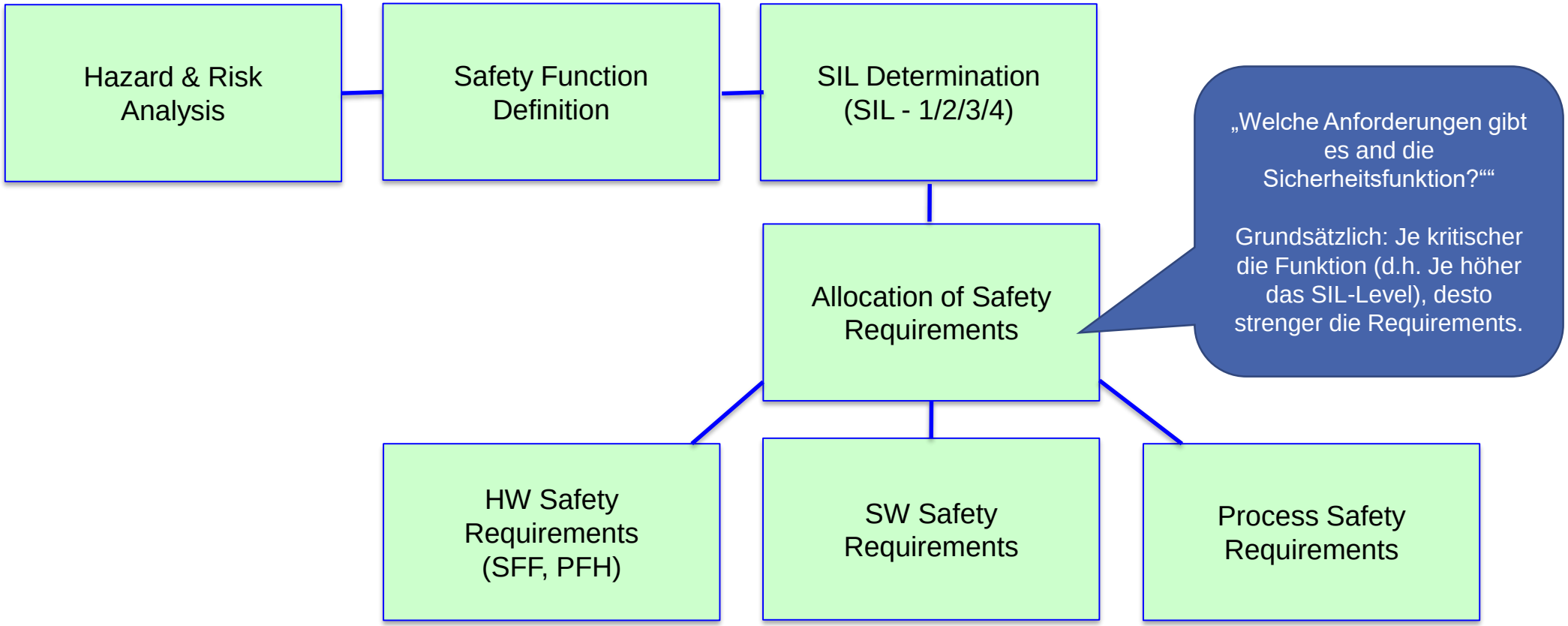
Vorgehensweise nach IEC 61508



Vorgehensweise nach IEC 61508



Vorgehensweise nach IEC 61508

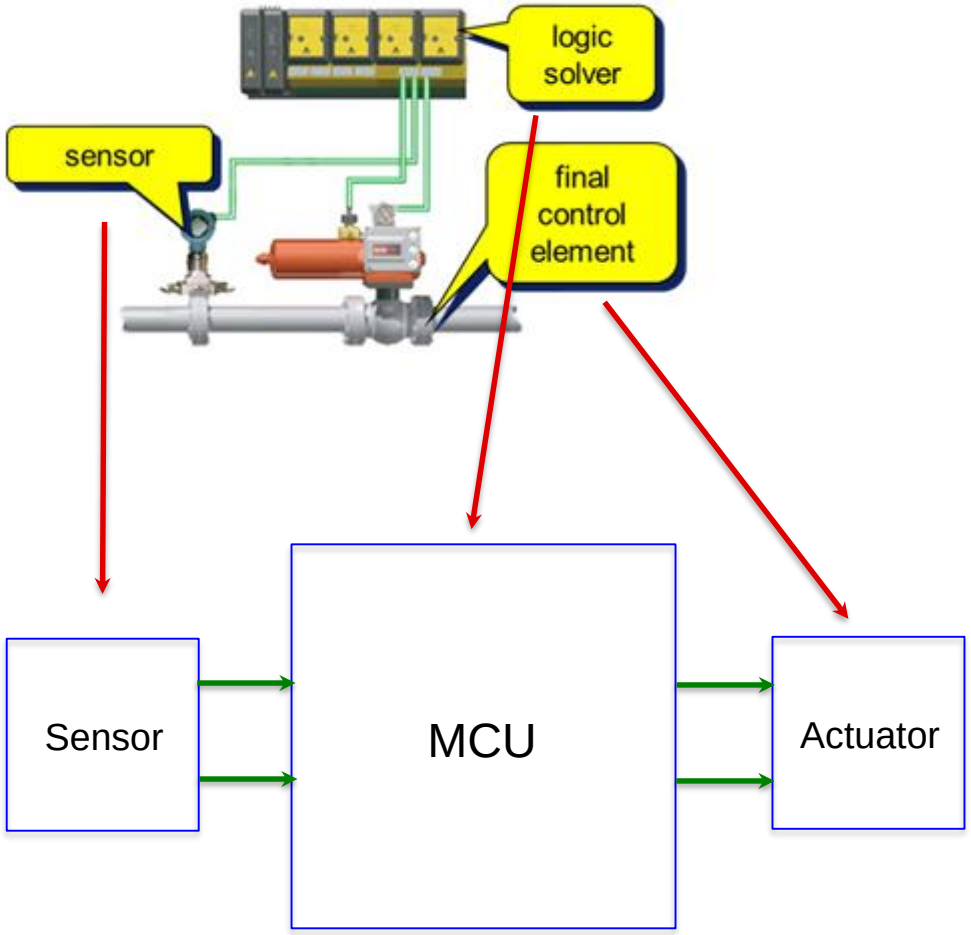


Sicherheitsfunktion – Beispiel: Ventil

- **Hazard:**
  State or event of potential danger to humans

- **Safety Function:**
  function to be implemented or other risk reduction measures, in order to achieve and/or maintain a safe state for the system

- **Safe State:**
  State of the system when safety is achieved and maintained

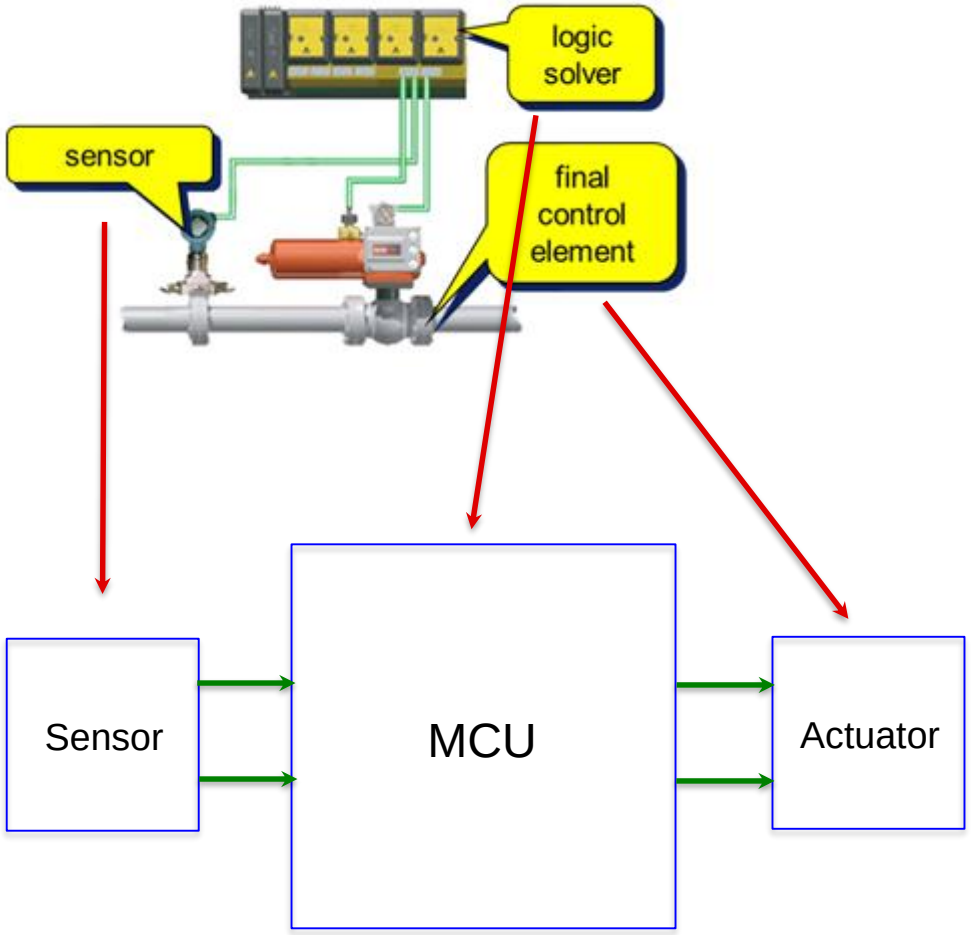


Sicherheitsfunktion – Beispiel: Ventil

- **Hazard:**
  High gas flow pressure

- **Safety Function:**
  Monitor the pressure of gas flow.
  If gas flow pressure exceeds a fixed limit, shut off the gas flow valve.

- **Safe State:**
  Gas flow is shut off

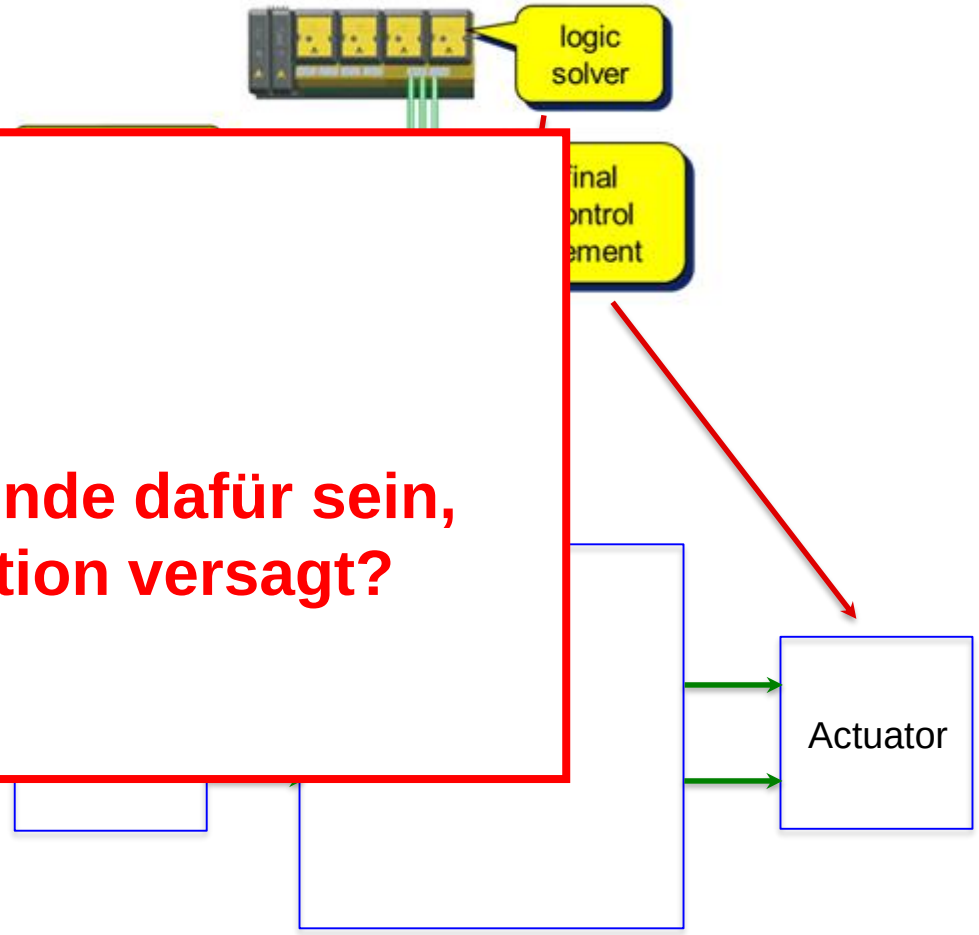


Sicherheitsfunktion – Beispiel: Ventil

- **Hazard:**
  High gas flow pressure

- **Safety Functio**
  Monitor the pre
  If gas flow pres
  shut off the gas

- **Safe State:**
  Gas flow is shu



Frage:

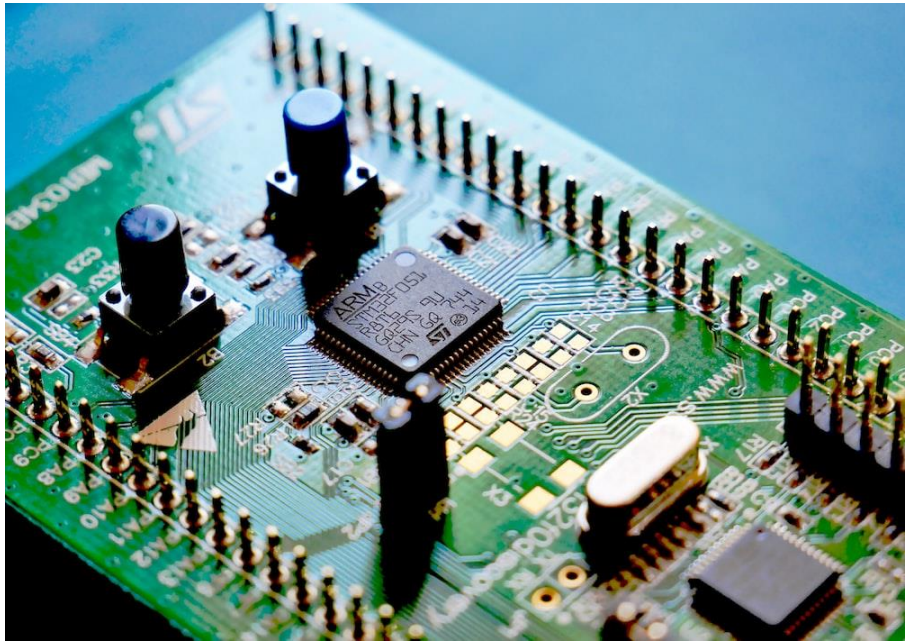
Was könnten mögliche Gründe dafür sein, dass die Sicherheitsfunktion versagt?

Mögliche Gefahrenquellen

- Zufällige Fehler
- Systematische Fehler
- Fehlgebrauch/Missbrauch von Maschinen

Mögliche Gefahrenquellen

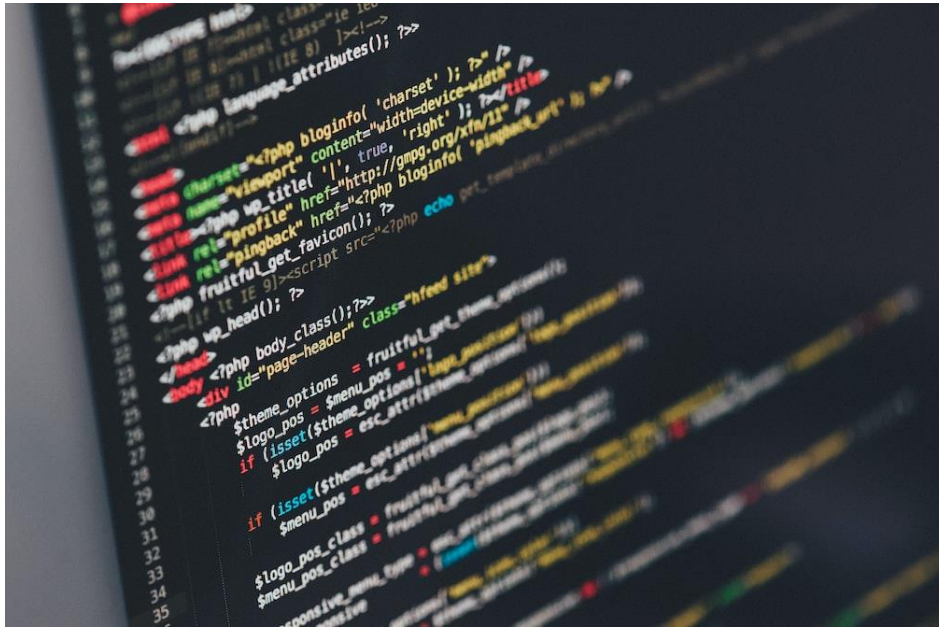
- **Zufällige Fehler**
  - Ausfall elektronischer Bauteile
  - Materialfehler
- Systematische Fehler
- Fehlgebrauch/Missbrauch von Maschinen



Ausfall von Hardwarekomponenten ist ein typischer „zufälliger“ Fehler. Man kennt zwar i.d.R. Eine statistische Auftretenswahr-scheinlichkeit, weiß aber nicht genau ob und wann ein Fehler auftreten wird.

Mögliche Gefahrenquellen

- Zufällige Fehler
- **Systematische Fehler**
  - Ungeeignete Konstruktion
  - Bugs in der Software
- Fehlgebrauch/Missbrauch von Maschinen



Merke: Software kennt keine „zufälligen“ Fehler, sondern nur systematische!

Mögliche Gefahrenquellen

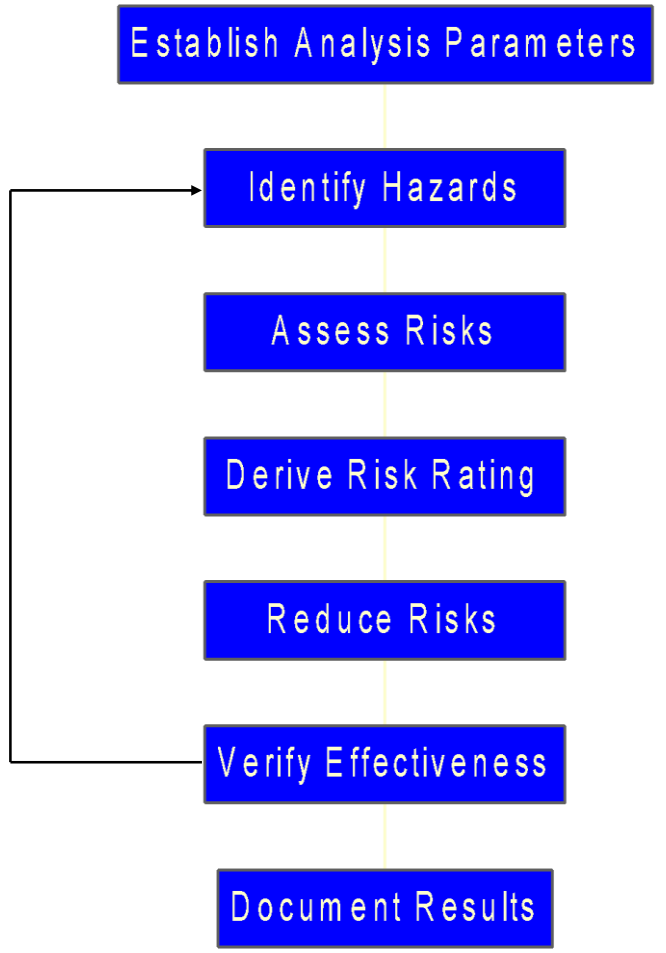
- Zufällige Fehler
- Systematische Fehler
- **Fehlgebrauch/Missbrauch von Maschinen**
  - Ungeeignete Umgebungsbedingungen
  - Bedienerfehler
  - Fehlende Wartung

Im industriellen Umfeld kann i.d.R. davon ausgegangen werden, dass Arbeiter geschult und in der Bedienung kritischer Systeme unterwiesen sind. Dennoch muss laut Norm „vorhersehbarer Fehlgebrauch“ berücksichtigt werden.

Hazard and Risk Analysis

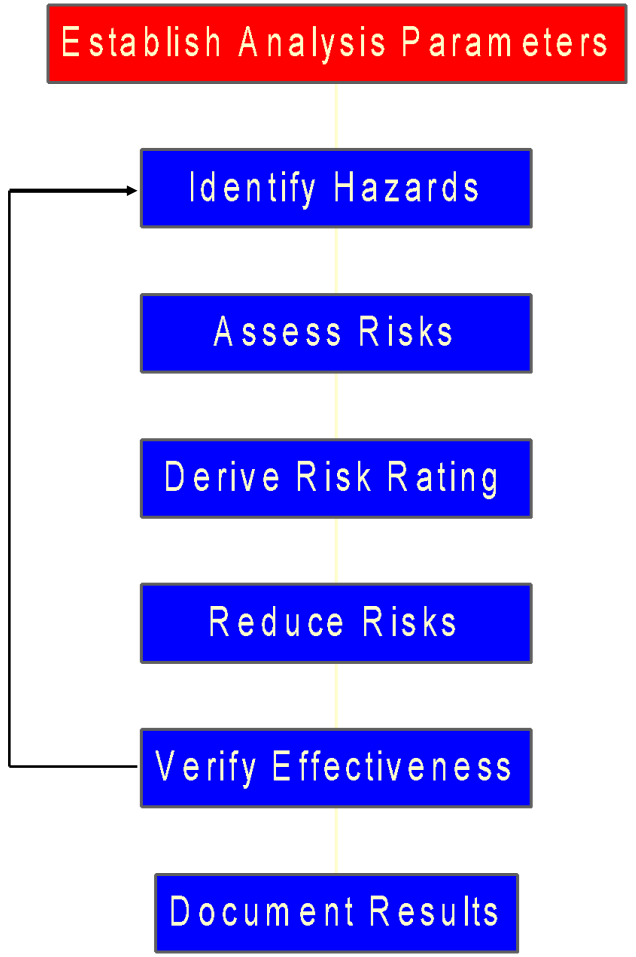
Hazard and Risk analysis
is a tool for engineers and safety practitioners to

- identify possible hazards,
- provide an evaluation of the risks, and
- prompt alternative design solutions to mitigate or control the risks to an acceptable level.



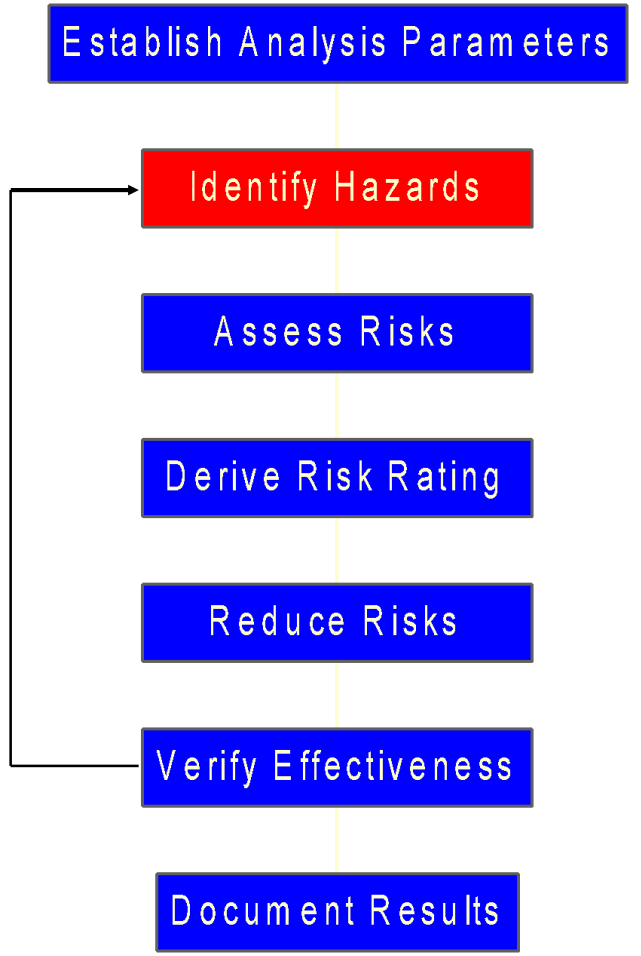
Hazard and Risk Analysis

- These parameters can be limits of the machine or design, limits on uses, limits on the scope of the analysis, or other limits.



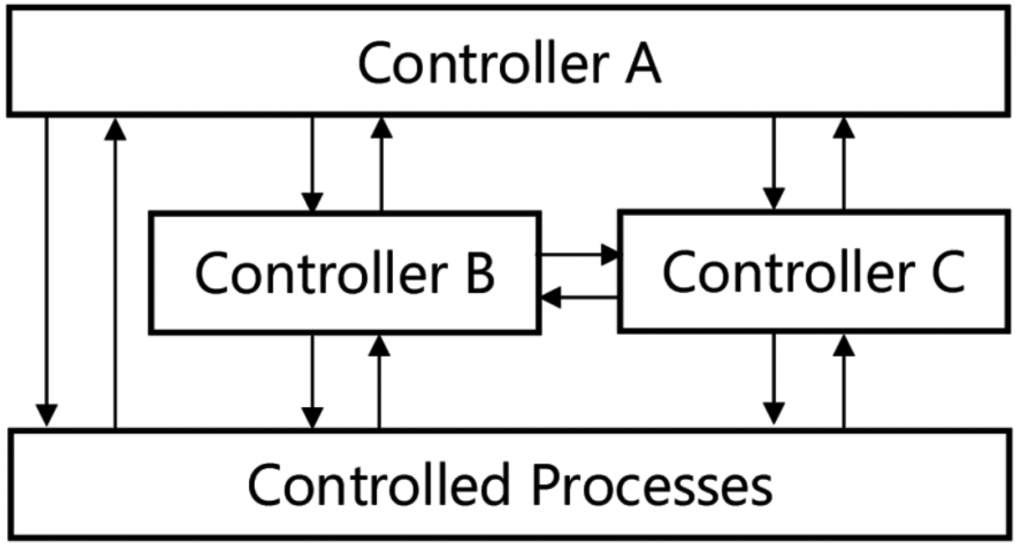
Hazard and Risk Analysis

- Identify potential hazards/hazardous situation
- This step can be supported by:
  - Expert knowledge/experience
  - Brainstorming
  - Structured analysis techniques (e.g., HAZOP, STPA)
  - Simulation
  - Formal methods



Example: Structured Hazard identification with STPA

- STPA: Systems-Theoretic Process Analysis
- Based on control structure diagram
- For each control action:
  - Analyze if a potential malfunction could be hazardous → “Hazardous Control Action” (HCA)
- For each HCA:
  - Analyze if and how the HCA can be caused.



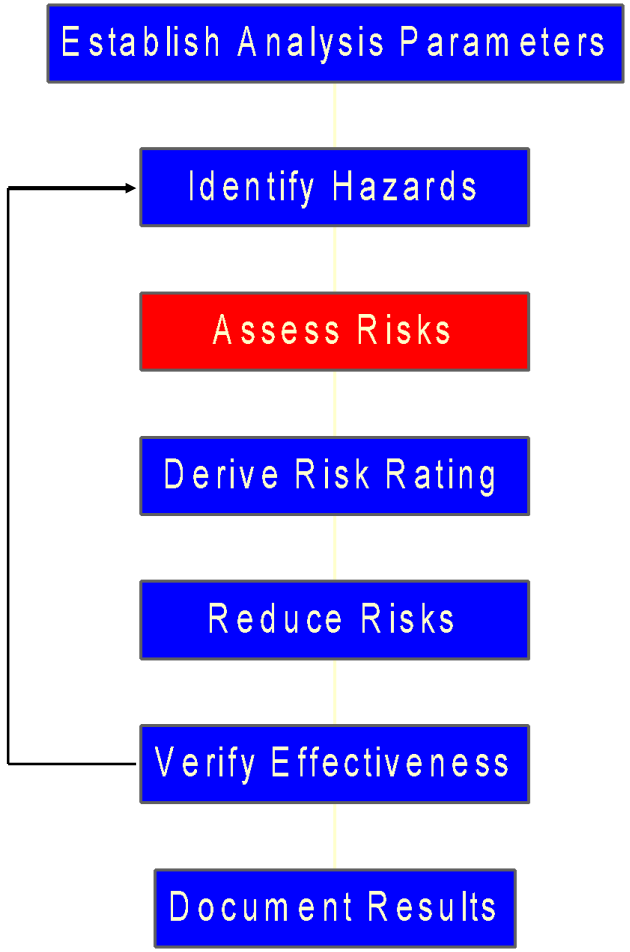
https://www.omron.com/global/en/assets/file/technology/omrontechnics/vol53/OMT_Vol53_006EN.pdf

For more details see: N. Leveson, „Engineering a Safer World: Systems Thinking Applied to Safety“, MIT Press, 2012

Hazard and Risk Analysis

Two risk factors are used:

- severity of injury
- probability of occurrence



Hazard and Risk Analysis

- **Hazard:**
  Potentially dangerous condition,
  which is triggered by an event, called the cause of the hazard.

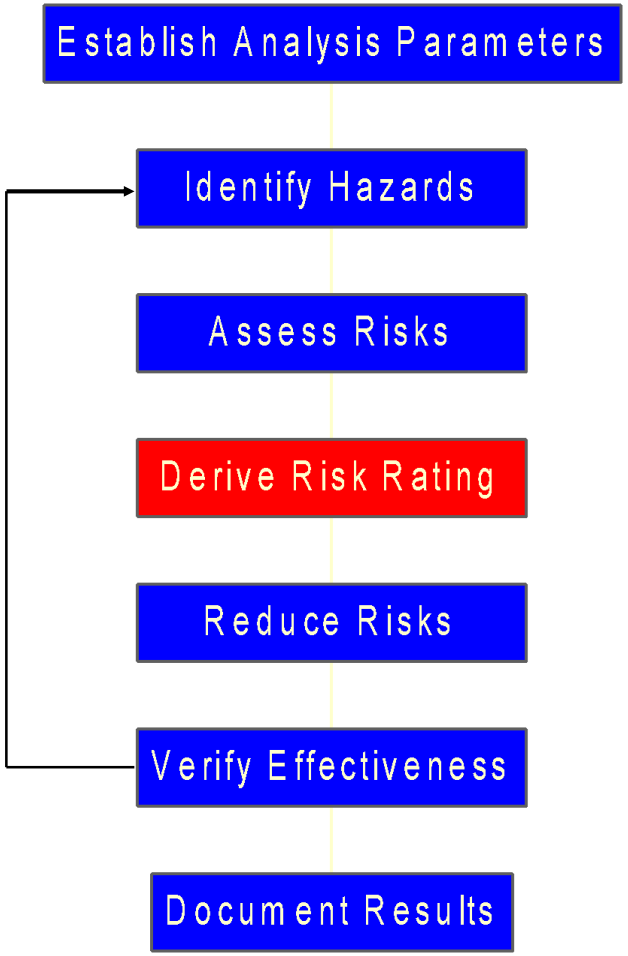
- **Risk:**
  hazard that is associated
  with a severity and a probability of occurrence.

Hazard and Risk Analysis

Risk matrix:

0 – 5 = Low Risk 6 – 10 = Moderate Risk 11 – 15 = High Risk 16 – 25 = extremely high unacceptable risk		Severity of the potential injury/damage				
		Insignificant damage to Property, Equipment or Minor Injury 1	Non-Reportable Injury, minor loss of Process or slight damage to Property 2	Reportable Injury moderate loss of Process or limited damage to Property 3	Major Injury, Single Fatality critical loss of Process/damage to Property 4	Multiple Fatalities Catastrophic Loss of Business 5
Likelihood of the hazard happening	Almost Certain 5	5	10	15	20	25
	Will probably occur 4	4	8	12	16	20
	Possible occur 3	3	6	9	12	15
	Remote possibility 2	2	4	6	8	10
	Extremely Unlikely 1	1	2	3	4	5

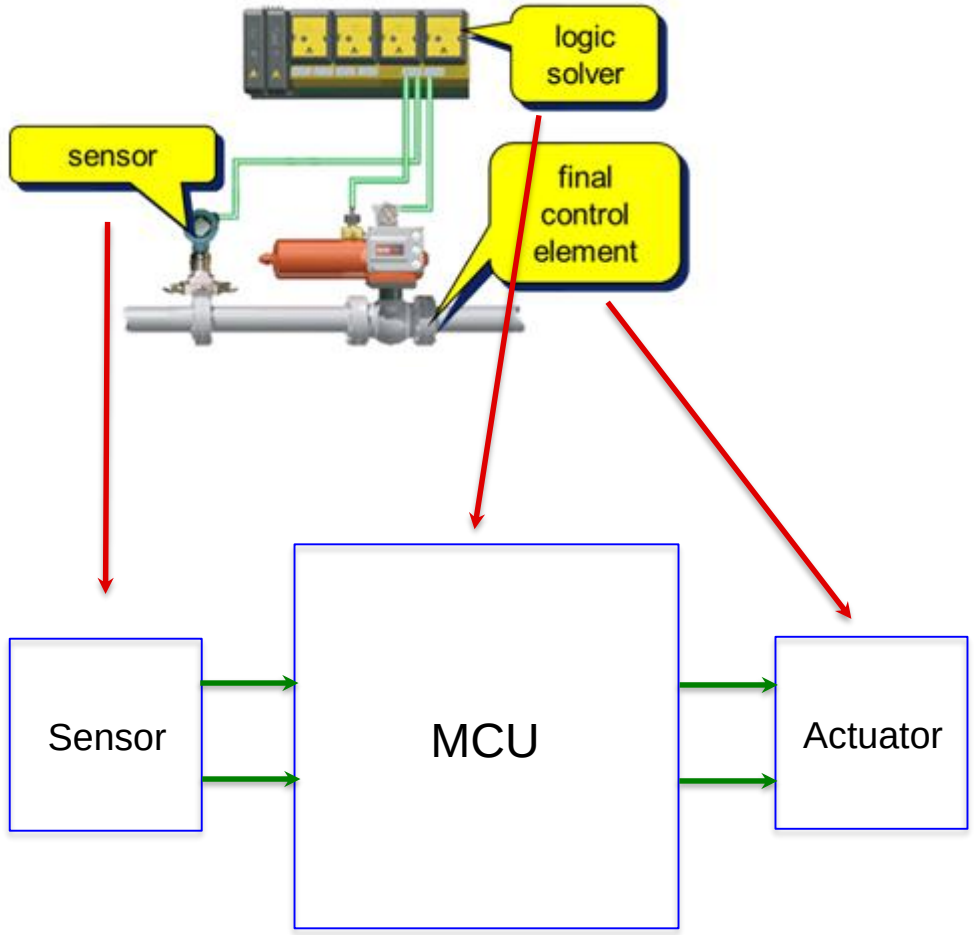
If the risk is determined to not be acceptable, it is necessary to reduce that risk by implementing protective measures.



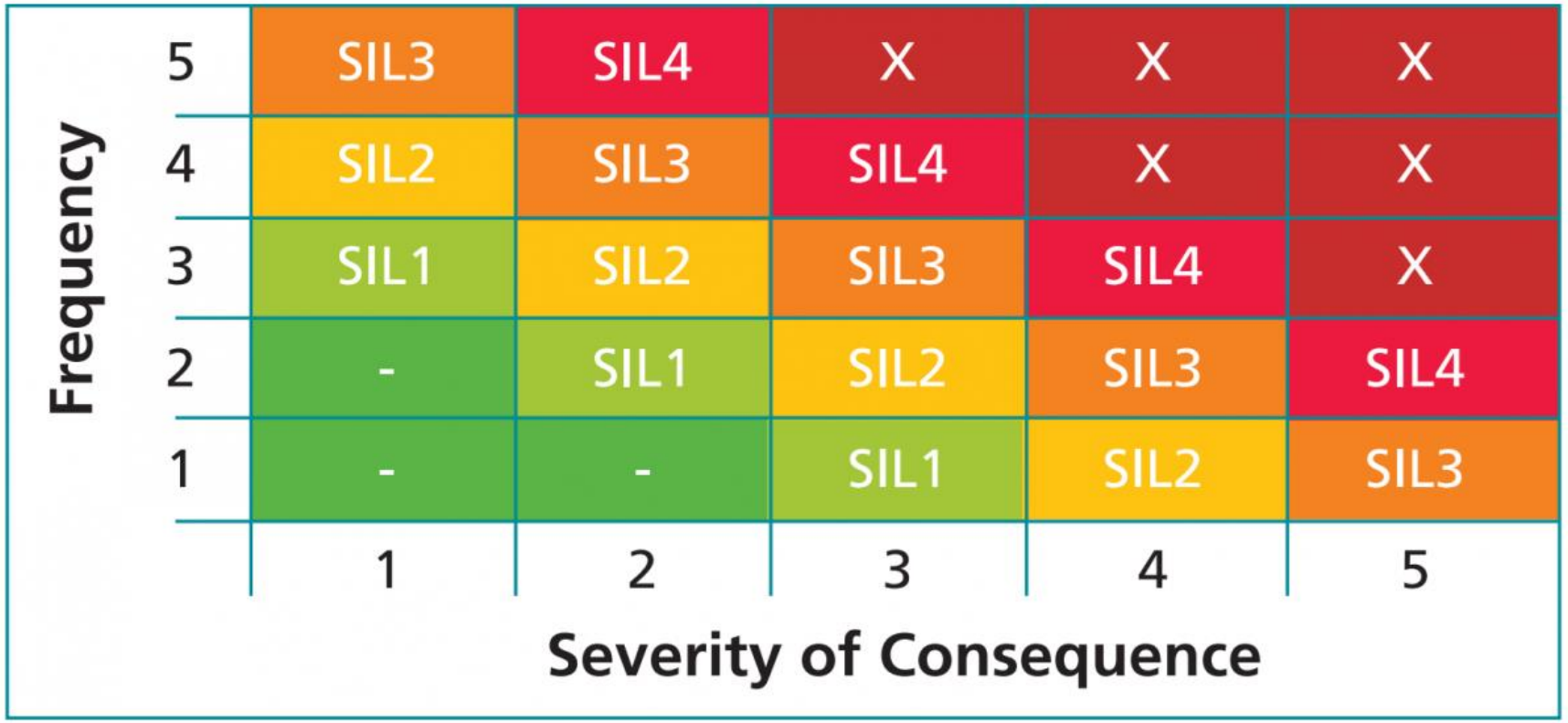
Risk Analysis / Safety Integrity Level

- **Risk Analysis**
  determines the performance requirement of the safety function, i.e. **SIL level**,

- **Safety Integrity Level** (SIL 1/2/3/4)
  is determined by the consequence and the frequency of hazardous event. The higher the SIL level, the higher the risk reduction requirements.



Safety Integrity Level



Safety Integrity Level

SIL Level	Probability of Failure	Consequence	Application Example
4	1 failure in 110,000 years	Potential for fatalities in the community	Nuclear Power Plant Control
3	1 failure in 11,100 years	Potential for multiple on-site fatalities	Hazardous area laser curtain sensors
2	1 failure in 1,100 years	Potential for major on-site injuries or fatalities	Hazardous liquid flow meter
1	1 failure in 110 years	Potential for minor on-site injuries	Thermal Meter

Safety Integrity Level

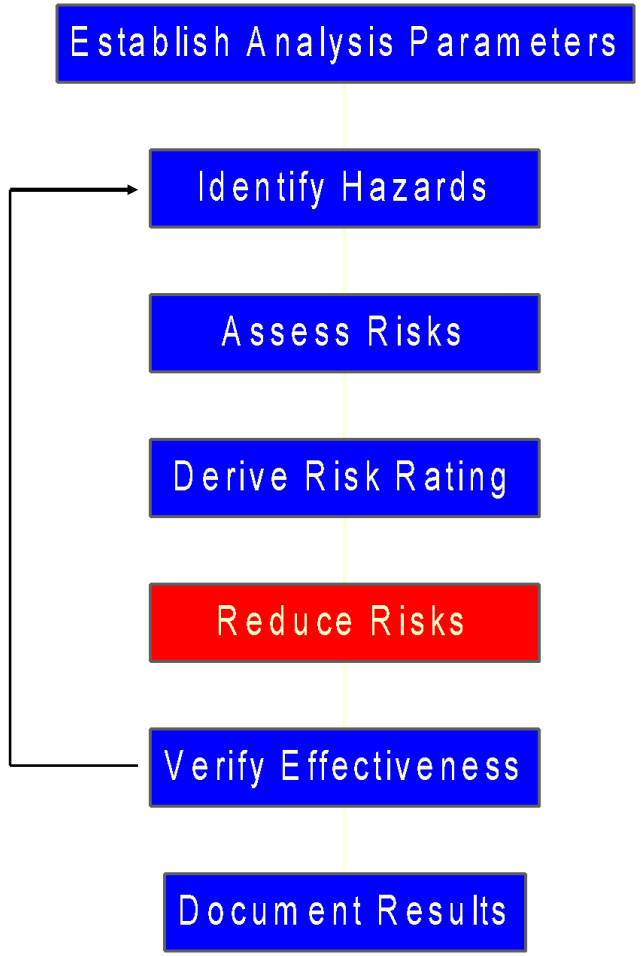
Safety Integrity Level	Safety	Probability of Failure on Demand	Risk Reduction Factor
SIL 4	> 99.99%	0.001% to 0.01%	100,000 to 10,000
SIL 3	99.9% to 99.99%	0.01% to 0.1%	10,000 to 1,000
SIL 2	99% to 99.9%	0.1% to 1%	1,000 to 100
SIL 1	90% to 99%	1% to 10%	100 to 10

Probability of Failure on Demand (PFD): Wahrscheinlichkeit eines Versagens bei Anforderung
(z.B. einer Sicherheitsfunktion bei Dauerbetrieb einer Maschine)

Hazard and Risk Analysis

Remedy actions are taken to reduce risks following the hazard hierarchy:

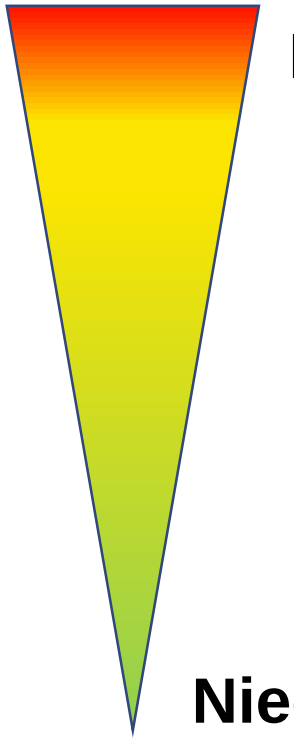
- Eliminate hazards through design
- Protect
- Warn the user
- Train the user(s)
- Personal protective equipment



Klassifikation von Schutzmaßnahmen nach ISO12100

Sicherheit von Maschinen

- **Konstruktiv:**
  Beseitigt Gefährdungen oder mindert Risiken, indem Konstruktions- oder Betriebseigenschaften der Maschine verändert werden (ohne Anwendung von trennenden / nicht-trennenden Schutzeinrichtungen)
- **Technisch:**
  Einsatz von Schutzeinrichtungen, wenn Risiken nicht durch konstruktive Maßnahmen beseitigt werden können.
- **Benutzerinformation:**
  Schutzmaßnahme durch Kommunikation/Weitergabe von Informationen an Benutzer.

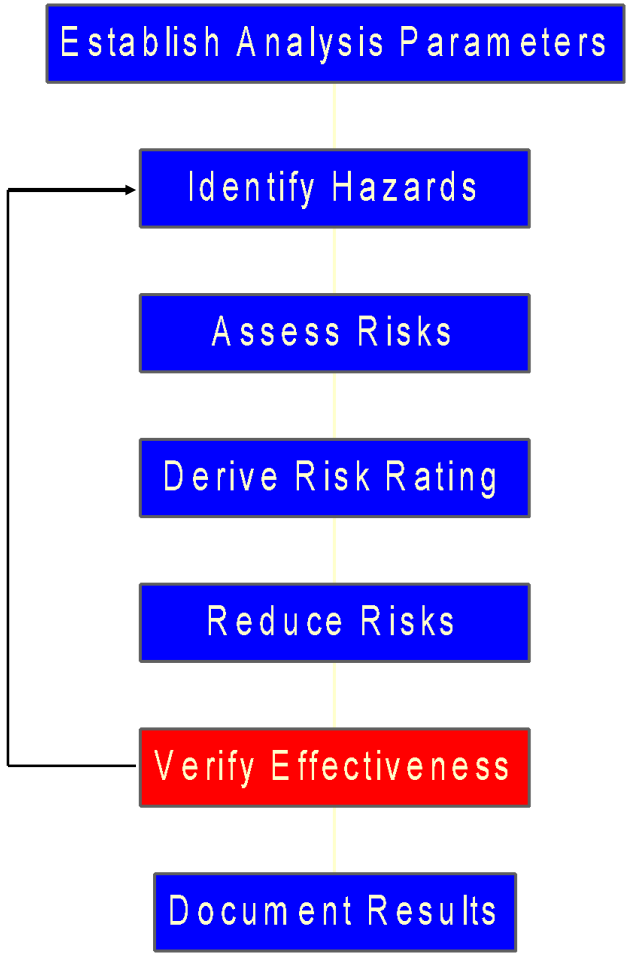


Hohe Priorität

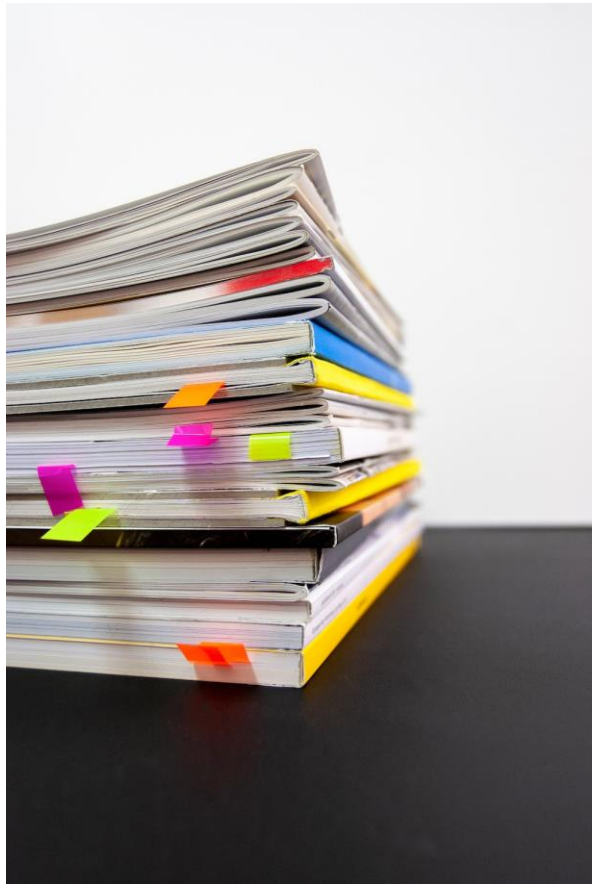
Niedrige Priorität

Hazard and Risk Analysis

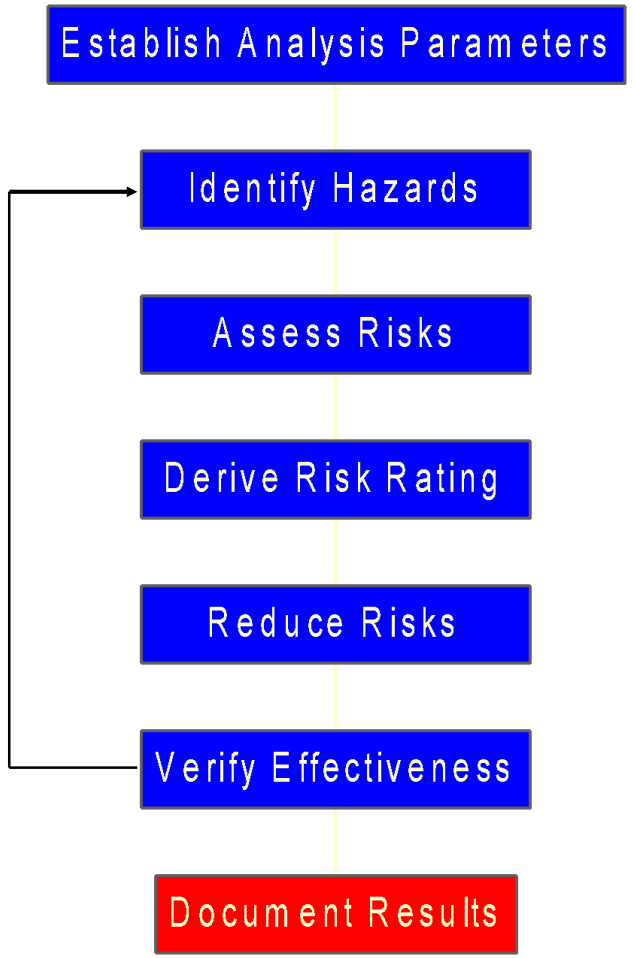
This assessment verifies that the remedy actions have reduced the risks to an acceptable level.



Hazard and Risk Analysis



Documentation of the development Process is extremely important (also from a legal perspective!)



Hazard and Risk Analysis

When to stop:

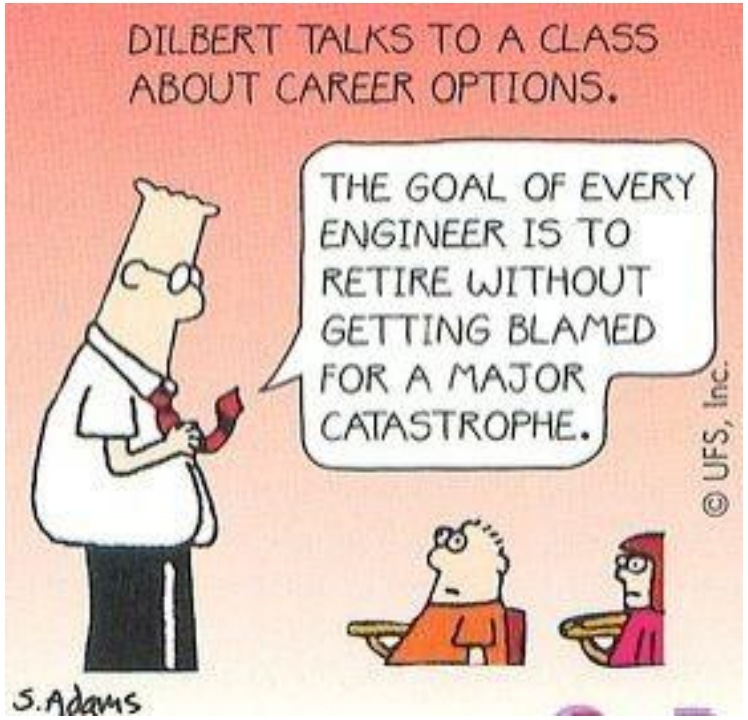
- There is no zero risk,
  always some residual risk remains.
- If the residual risk is acceptable,
  then the risk assessment process
  is completed.

Hazard and Risk Analysis

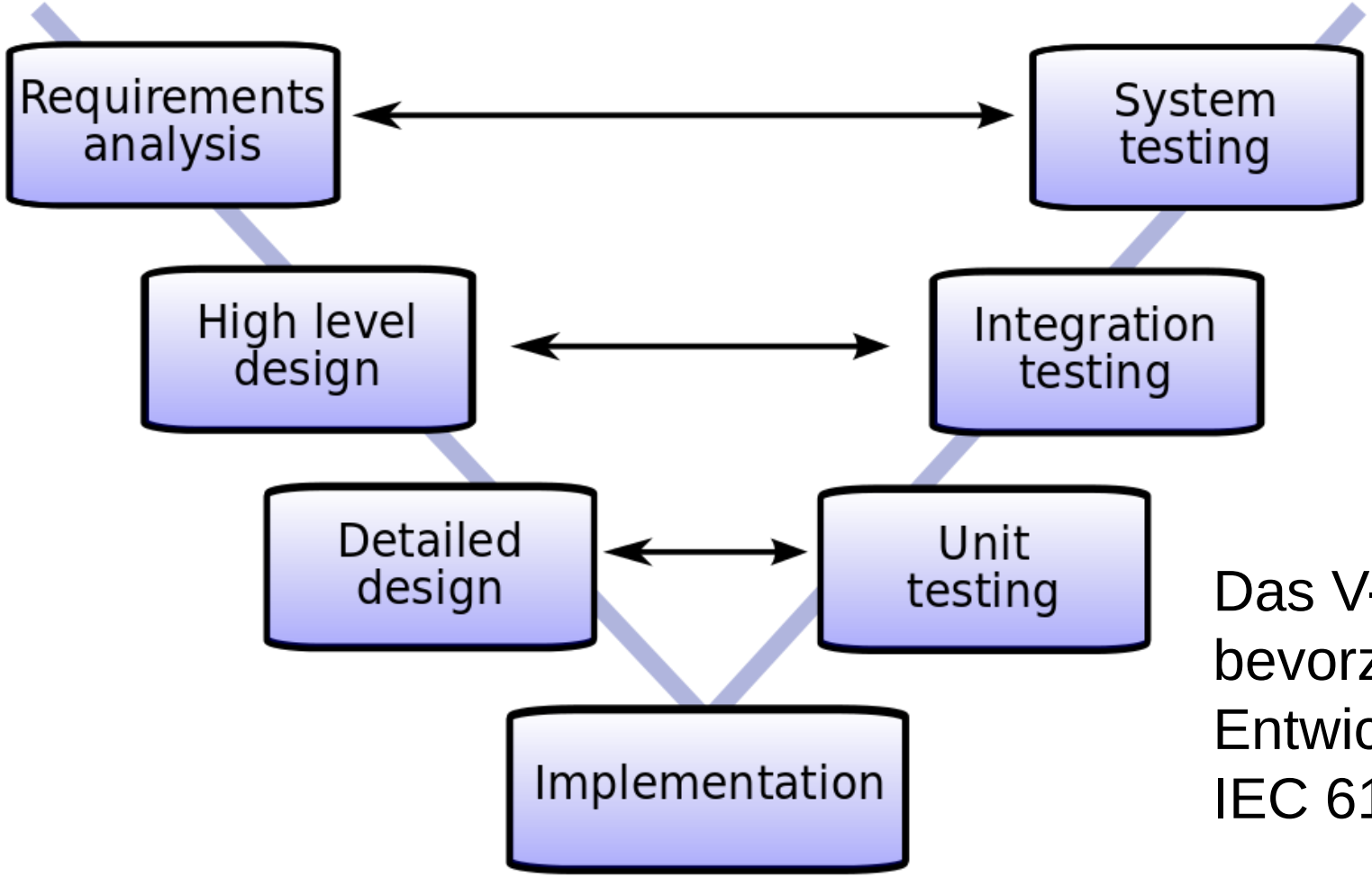
Caution!

- Any hazard, which is not identified,
  will not be addressed by safety measures and
  will not be detected during testing.

- Hazard not identified during this analysis
  can create **substantial risk to users**
  of the system.



Entwicklungsprozess



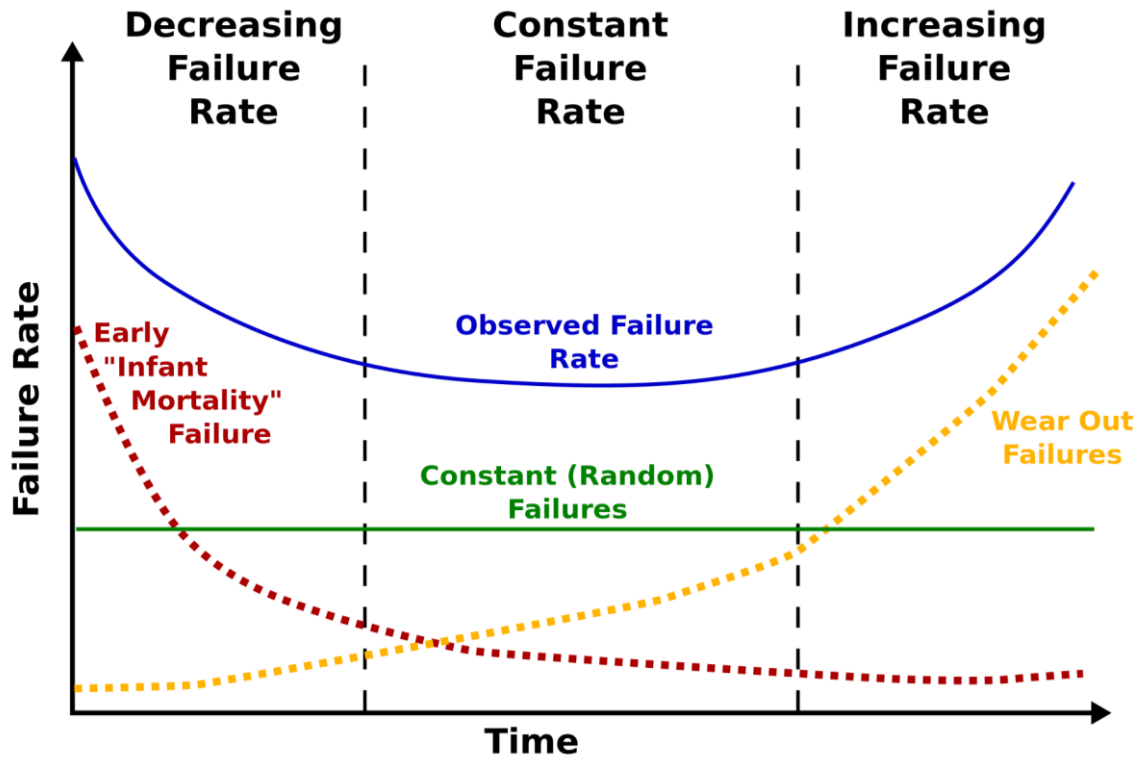
Das V-Modell ist der bevorzugte Entwicklungsprozess nach IEC 61508

Entwicklungsprozess

Phase	Process Steps in Phase
Safety Requirements	Create and Inspect Product Safety Requirements
Safety Validation Test Planning	Create and Inspect Safety Validation Test Plan
System Architecture Design	Create and Inspect System Architecture Design Perform System FMEA Create and Inspect Derived Safety Requirements Create and Inspect Integration Test Plan
Hardware Design	Perform Detailed Hardware Design Perform Hardware FMEDA Perform Fault Injection Testing
Software Design	Create and Inspect Software Architecture Perform Software Criticality Analysis and HAZOP Create and Inspect Detailed Software Design
Implementation	Create and Inspect Code Perform Static Analysis Unit Test Code
Integration and Safety Validation Test Execution	Perform Integration Testing Perform Validation Testing

Hardware

- Systematic and random failures
- Permanent and transient failures



Hardware - Permanent Failures

- Examples: Short, Open, Stuck At, Drift

- Sources of permanent component failure rate data:
  - MILHDBK 217F
  - SN29500
  - IEC/TR 62380
  - Supplier reliability data

Easy for simple components!
Estimation for complex components (based on # of transistors, # of memory bits, temperature, package effect)

- Failure rate is commonly expressed in FIT (Failure In Time)
  → 1 FIT = 1 failure in 1E9 hours.

- E.g. perform FMEDA to calculate SFF and PFH

Safety Integrity Level

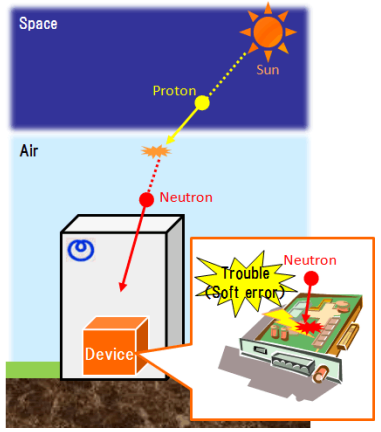
- Safety Integrity Level is characterized by SFF and PFD_{AVG} or PFH
  - Safe Failure Fraction (SFF) → relative metric
  - Probability of Fail on Demand Average (PFD_{AVG}) → absolute metric
  - Probability of Fail per Hour (PFH) → absolute metric

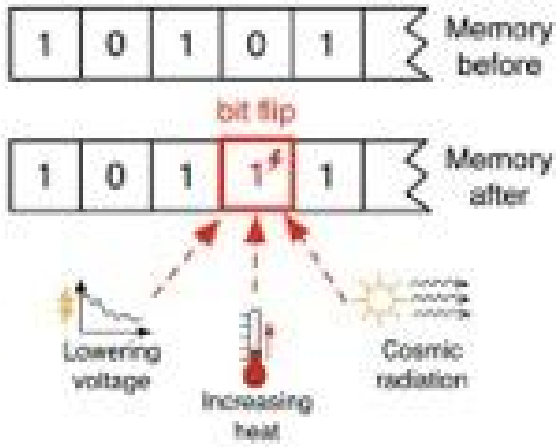
- Failure rate: λ
  - λ_{SAFE}: Safe / non-hazardous failure of a component
  - $\lambda_{DANGEROUS\text{-}DETECTED}$: Hazardous failure that is detected
  - $\lambda_{DANGEROUS\text{-}UNDETECTED}$: Hazardous failure that is not detected

- $PFH \approx 1 / \lambda_{DANGEROUS\text{-}UNDETECTED}$
- $SFF = (\lambda_{SAFE} + \lambda_{DANGEROUS\text{-}DETECTED}) / (\lambda_{SAFE} + \lambda_{DANGEROUS\text{-}DETECTED} + \lambda_{DANGEROUS\text{-}UNDETECTED})$

Hardware – Transient Failures

- Failures of only temporary time („Bitkipper“)
- Caused by:
  - Cosmic Rays
  - EMC





- Failure rate data source is TI experiments in Los Alamos lab and TI lab
- Handle e.g. through permanent self-tests implemented in SW, sufficient Hamming-distance

Software

- No random failures
- Only systematic failures

- Steve McConnel:
  „Industry average: about 15-50 errors per 1000 lines of delivered code“

- How to prevent bugs (in delivered software)?
  - Testing
  - Process of developing software

Also applies to prevention of systematic failures in Hardware

IEC 61508-3

Edition 2.0 2010-04

INTERNATIONAL STANDARD

NORME INTERNATIONALE

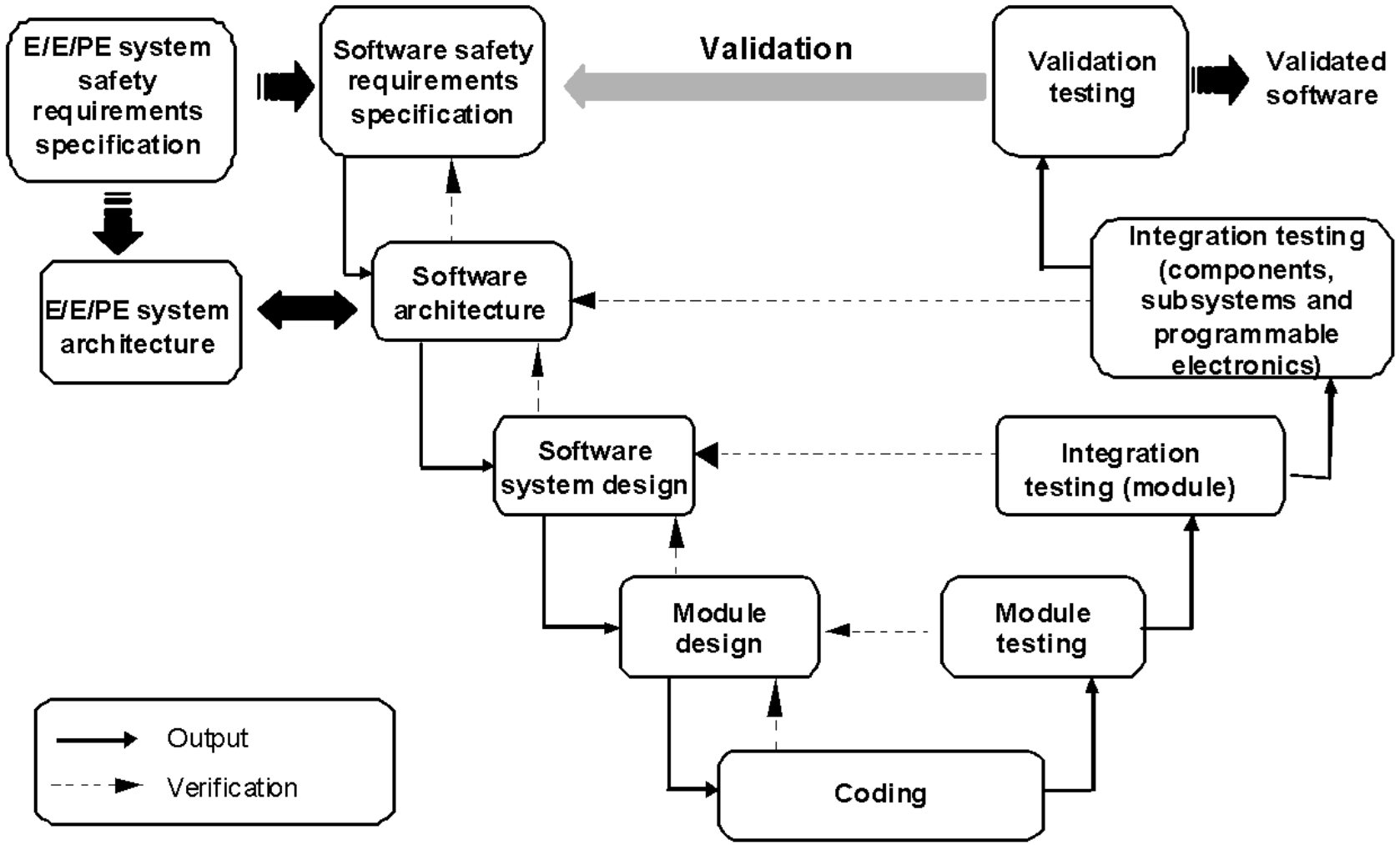
BASIC SAFETY PUBLICATION
PUBLICATION FONDAMENTALE DE SÉCURITÉ

Functional safety of electrical/electronic/programmable electronic safety-related systems –
Part 3: Software requirements

Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité –
Partie 3: Exigences concernant les logiciels

IEC 61508-3:2010

Software Development V-model



Software Development: Recommended Techniques

IEC 61508 makes recommendations for development techniques depending on the respective SIL Level

- HR: Highly recommended
- R: Recommended
- NR: Not recommended

HR	the technique or measure is highly recommended for this safety integrity level. If this technique or measure is not used then the rationale behind not using it should be detailed with reference to Annex C during the safety planning and agreed with the assessor.
R	the technique or measure is recommended for this safety integrity level as a lower recommendation to a HR recommendation.
---	the technique or measure has no recommendation for or against being used.
NR	the technique or measure is positively not recommended for this safety integrity level. If this technique or measure is used then the rationale behind using it should be detailed with reference to Annex C during the safety planning and agreed with the assessor.

Source: IEC 61508, Part 3

Software Safety Requirements Specification

Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1a	Semi-formal methods	Table B.7	R	R	HR	HR
1b	Formal methods	B.2.2, C.2.4	---	R	R	HR
2	Forward traceability between the system safety requirements and the software safety requirements	C.2.11	R	R	HR	HR
3	Backward traceability between the safety requirements and the perceived safety needs	C.2.11	R	R	HR	HR
4	Computer-aided specification tools to support appropriate techniques/measures above	B.2.4	R	R	HR	HR

NOTE 1 The software safety requirements specification will always require a description of the problem in natural language and any necessary mathematical notation that reflects the application.

NOTE 2 The table reflects additional requirements for specifying the software safety requirements clearly and precisely.

NOTE 3 See Table C.1.

NOTE 4 The references (which are informative, not normative) "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

Source: IEC 61508, Part 3

Software Design and Development: Software Architecture Design

Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
	Architecture and design feature					
1	Fault detection	C.3.1	---	R	HR	HR
2	Error detecting codes	C.3.2	R	R	R	HR
3a	Failure assertion programming	C.3.3	R	R	R	HR
3b	Diverse monitor techniques (with independence between the monitor and the monitored function in the same computer)	C.3.4	---	R	R	----
3c	Diverse monitor techniques (with separation between the monitor computer and the monitored computer)	C.3.4	---	R	R	HR
3d	Diverse redundancy, implementing the same software safety requirements specification	C.3.5	---	---	---	R
3e	Functionally diverse redundancy, implementing different software safety requirements specification	C.3.5	---	---	R	HR
3f	Backward recovery	C.3.6	R	R	---	NR
3g	Stateless software design (or limited state design)	C.2.12	---	---	R	HR
4a	Re-try fault recovery mechanisms	C.3.7	R	R	---	---
4b	Graceful degradation	C.3.8	R	R	HR	HR
5	Artificial intelligence - fault correction	C.3.9	---	NR	NR	NR
6	Dynamic reconfiguration	C.3.10	---	NR	NR	NR
7	Modular approach	Table B.9	HR	HR	HR	HR

Source: IEC 61508, Part 3

Software Design and Development: Software Architecture Design

Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
	Architecture and design feature					
8	Use of trusted/verified software elements (if available)	C.2.10	R	HR	HR	HR
9	Forward traceability between the software safety requirements specification and software architecture	C.2.11	R	R	HR	HR
10	Backward traceability between the software safety requirements specification and software architecture	C.2.11	R	R	HR	HR
11a	Structured diagrammatic methods **	C.2.1	HR	HR	HR	HR
11b	Semi-formal methods **	Table B.7	R	R	HR	HR
11c	Formal design and refinement methods **	B.2.2, C.2.4	---	R	R	HR
11d	Automatic software generation	C.4.6	R	R	R	R
12	Computer-aided specification and design tools	B.2.4	R	R	HR	HR
13a	Cyclic behaviour, with guaranteed maximum cycle time	C.3.11	R	HR	HR	HR
13b	Time-triggered architecture	C.3.11	R	HR	HR	HR
13c	Event-driven, with guaranteed maximum response time	C.3.11	R	HR	HR	-
14	Static resource allocation	C.2.6.3	-	R	HR	HR
15	Static synchronisation of access to shared resources	C.2.6.3	-	-	R	HR

Software Design and Development: Detailed Design

Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1a	Structured methods **	C.2.1	HR	HR	HR	HR
1b	Semi-formal methods **	Table B.7	R	HR	HR	HR
1c	Formal design and refinement methods **	B.2.2, C.2.4	---	R	R	HR
2	Computer-aided design tools	B.3.5	R	R	HR	HR
3	Defensive programming	C.2.5	---	R	HR	HR
4	Modular approach	Table B.9	HR	HR	HR	HR
5	Design and coding standards	C.2.6 Table B.1	R	HR	HR	HR
6	Structured programming	C.2.7	HR	HR	HR	HR
7	Use of trusted/verified software elements (if available)	C.2.10	R	HR	HR	HR
8	Forward traceability between the software safety requirements specification and software design	C.2.11	R	R	HR	HR

NOTE 1 See Table C.4.

NOTE 2 There is still debate about the suitability of OO software development for safety-related systems. See Annex G of IEC 61508-7 for guidance on object oriented architecture and design.

NOTE 3 The references (which are informative, not normative) "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

** Group 1, "Structured methods". Use measure 1a only if 1b is not suited to the domain for SIL 3+4.

Software Design and Development: Support Tools and Programming Language

Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	Suitable programming language	C.4.5	HR	HR	HR	HR
2	Strongly typed programming language	C.4.1	HR	HR	HR	HR
3	Language subset	C.4.2	---	---	HR	HR
4a	Certified tools and certified translators	C.4.3	R	HR	HR	HR
4b	Tools and translators: increased confidence from use	C.4.4	HR	HR	HR	HR
NOTE 1 See Table C.3. NOTE 2 The references (which are informative, not normative) "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.						
* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.						

Widely used: C / C++

BUT: not every function can be used! (see next slide)

Not only language, but also tools relevant → e.g. Compiler!

Design and Coding Standards

Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	Use of coding standard to reduce likelihood of errors	C.2.6.2	HR	HR	HR	HR
2	No dynamic objects	C.2.6.3	R	HR	HR	HR
3a	No dynamic variables	C.2.6.3	---	R	HR	HR
3b	Online checking of the installation of dynamic variables	C.2.6.4	---	R	HR	HR
4	Limited use of interrupts	C.2.6.5	R	R	HR	HR
5	Limited use of pointers	C.2.6.6	---	R	HR	HR
6	Limited use of recursion	C.2.6.7	---	R	HR	HR
7	No unstructured control flow in programs in higher level languages	C.2.6.2	R	HR	HR	HR
8	No automatic type conversion	C.2.6.2	R	HR	HR	HR

NOTE 1 Measures 2, 3a and 5. The use of dynamic objects (for example on the execution stack or on a heap) may impose requirements on both available memory and also execution time. Measures 2, 3a and 5 do not need to be applied if a compiler is used which ensures a) that sufficient memory for all dynamic variables and objects will be allocated before runtime, or which guarantees that in case of memory allocation error, a safe state is achieved; b) that response times meet the requirements.

NOTE 2 See Table C.11.

NOTE 3 The references (which are informative, not normative) "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

Source: IEC 61508, Part 3

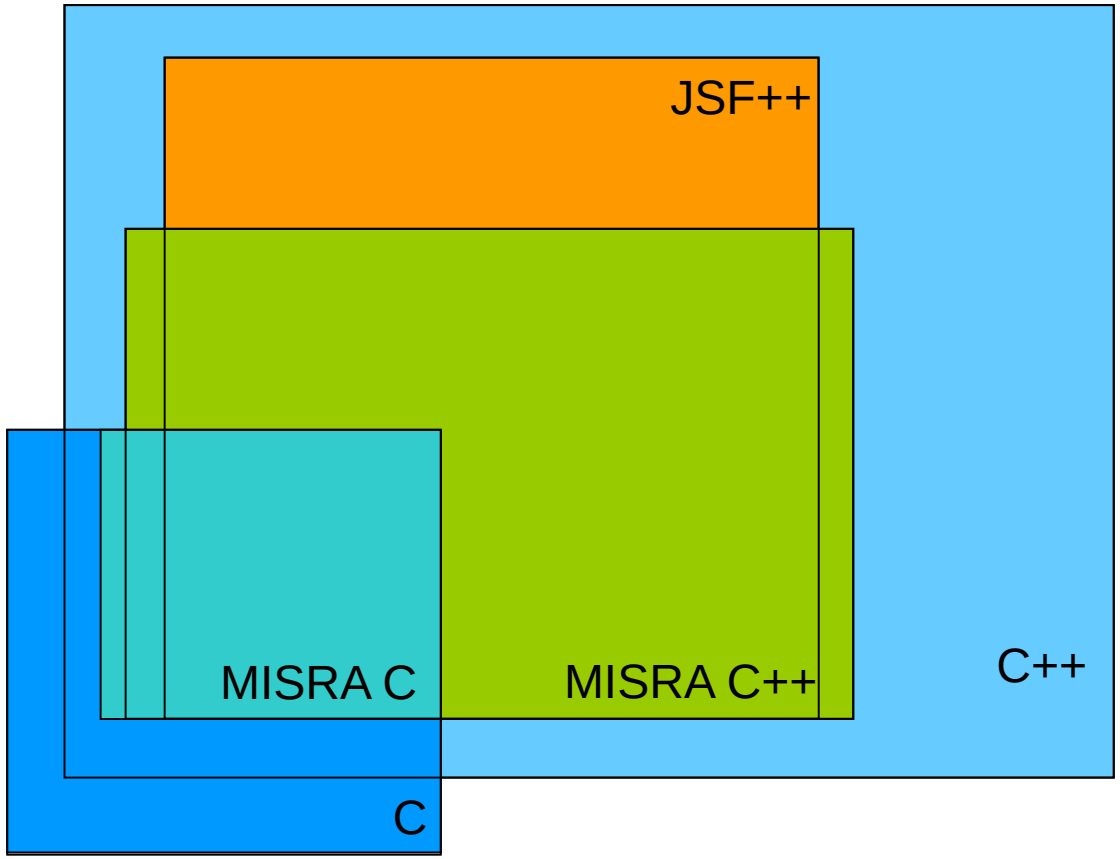
Safety-critical Coding Standards

- Misra C/C++ (Motor Industry Software Reliability Association)
  - Rules first introduced in 1998
  - Revised in 2004: 141 rules for C
  - Revised to cover C++ in 2008 (mostly derived from JSF rules): 228 rules
  - Widely used in motor vehicle industry
  - Some support in popular embedded compilers
  - Closed standard
- JSF
  - Joint Strike Fighter Air Vehicle standards, introduced in 2005
  - 232 rules for C and C++, some based on Misra C
  - Not widely used yet due to low uptake of C++ for safety-critical
  - Open standard
- EC
  - A subset of ISO C, introduced in 2003 by Les Hatton
  - Designed to be “measurement based”
  - Open standard

References

- Coding rules
  - Misra C: http://www.misra.org.uk/
  - JSF: http://www.research.att.com/~bs/JSF-AV-rules.pdf
  - EC: http://www.leshatton.org/ISOC_subset1103.html

Relationship Between C, C++, MISRA C/C++, JSF++



Rationale for Coding Rules

- Clarity
  - Reduce programmer confusion
  - A construct may be perfectly unambiguous and well-defined, but may make code difficult to read
  - *“Do not use goto statements”* (JPL 1.1)
- Predictability
  - Eliminate sources of ambiguity
  - Helps with portability
  - *“Do not use union types”* (Misra C++ 9-5-1)
- Simplicity
  - Keep the program simple
  - May help reduce the cost of testing
  - Keeps programs amenable to analysis
  - *“No recursion”* (JPL 1.2)

Rationale for Coding Rules

- Defense
  - Encourage defensive programming
  - Fosters maintainability
  - *“Switch statement shall be a well-formed switch statement”* (Misra C++ 6-4-3)
- Compliance
  - Standards compliance
  - Aids portability
  - *“Use IEEE floating point formats”* (Misra C++ 0-4-3)
- Process
  - How code is developed, not about the code itself
  - *“Compile with all warnings enabled, and use source code analyzers”* (JPL 10)
- Performance
  - Nothing to do with safety
  - *“Trivial forwarding functions should be inlined”* (JSF 124)

Software Design and Development: Software Module Testing and Integration

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	Probabilistic testing	C.5.1	---	R	R	R
2	Dynamic analysis and testing	B.6.5 Table B.2	R	HR	HR	HR
3	Data recording and analysis	C.5.2	HR	HR	HR	HR
4	Functional and black box testing	B.5.1 B.5.2 Table B.3	HR	HR	HR	HR
5	Performance testing	Table B.6	R	R	HR	HR
6	Model based testing	C.5.27	R	R	HR	HR
7	Interface testing	C.5.3	R	R	HR	HR
8	Test management and automation tools	C.4.7	R	HR	HR	HR
9	Forward traceability between the software design specification and the module and integration test specifications	C.2.11	R	R	HR	HR
10	Formal verification	C.5.12	---	---	R	R

NOTE 1 Software module and integration testing are verification activities (see Table B.9).

NOTE 2 See Table C.5.

NOTE 3 Technique 9. Formal verification may reduce the amount and extent of module and integration testing required.

NOTE 4 The references (which are informative, not normative) "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

Programmable Electronics Integration (Hardware and Software)

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	Functional and black box testing	B.5.1 B.5.2 Table B.3	HR	HR	HR	HR
2	Performance testing	Table B.6	R	R	HR	HR
3	Forward traceability between the system and software design requirements for hardware/software integration and the hardware/software integration test specifications	C.2.11	R	R	HR	HR
NOTE 1 Programmable electronics integration is a verification activity (see Table A.9). NOTE 2 See Table C.6. NOTE 3 The references (which are informative, not normative) "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.						
* Appropriate techniques/measures shall be selected according to the safety integrity level.						

Source: IEC 61508, Part 3

Software Aspects of System Safety Validation

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	Probabilistic testing	C.5.1	---	R	R	HR
2	Process simulation	C.5.18	R	R	HR	HR
3	Modelling	Table B.5	R	R	HR	HR
4	Functional and black-box testing	B.5.1 B.5.2 Table B.3	HR	HR	HR	HR
5	Forward traceability between the software safety requirements specification and the software safety validation plan	C.2.11	R	R	HR	HR
6	Backward traceability between the software safety validation plan and the software safety requirements specification	C.2.11	R	R	HR	HR

NOTE 1 See Table C.7.

NOTE 2 The references (which are informative, not normative) "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

Source: IEC 61508, Part 3

Failure Analysis

Technique/Measure *		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1a	Cause consequence diagrams	B.6.6.2	R	R	R	R
1b	Event tree analysis	B.6.6.3	R	R	R	R
2	Fault tree analysis	B.6.6.5	R	R	R	R
3	Software functional failure analysis	B.6.6.4	R	R	R	R

NOTE 1 Preliminary hazard analysis should have already taken place in order to categorize the software into the most appropriate safety integrity level.

NOTE 2 See Table C.14.

NOTE 3 The references (which are informative, not normative) "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

Source: IEC 61508, Part 3

Modelling

	Technique/Measure *	Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	Data flow diagrams	C.2.2	R	R	R	R
2a	Finite state machines	B.2.3.2	---	R	HR	HR
2b	Formal methods	B.2.2, C.2.4	---	R	R	HR
2c	Time Petri nets	B.2.3.3	---	R	HR	HR
3	Performance modelling	C.5.20	R	HR	HR	HR
4	Prototyping/animation	C.5.17	R	R	R	R
5	Structure diagrams	C.2.3	R	R	R	HR

NOTE 1 If a specific technique is not listed in the table, it should not be assumed that it is excluded from consideration. It should conform to this standard.

NOTE 2 Quantification of probabilities is not required.

NOTE 3 See Table C.15.

NOTE 4 The references (which are informative, not normative) "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

Source: IEC 61508, Part 3

Static Analysis

Technique/Measure *		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	Boundary value analysis	C.5.4	R	R	HR	HR
2	Checklists	B.2.5	R	R	R	R
3	Control flow analysis	C.5.9	R	HR	HR	HR
4	Data flow analysis	C.5.10	R	HR	HR	HR
5	Error guessing	C.5.5	R	R	R	R
6a	Formal inspections, including specific criteria	C.5.14	R	R	HR	HR
6b	Walk-through (software)	C.5.15	R	R	R	R
7	Symbolic execution	C.5.11	---	---	R	R
8	Design review	C.5.16	HR	HR	HR	HR
9	Static analysis of run time error behaviour	B.2.2, C.2.4	R	R	R	HR
10	Worst-case execution time analysis	C.5.20	R	R	R	R

NOTE 1 See Table C.18.

NOTE 2 The references "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

Source: IEC 61508, Part 3

Reliability vs. Safety

Reliability (Zuverlässigkeit):

- The probability an item will perform its required function in the specified manner over a given time period and under specified or assumed conditions.
- Most accidents result from errors in specified requirements or functions and deviations from assumed conditions (i.e., not necessarily from unreliable components!)

How to reduce failures and failure rate:

- Redundancy
- Safety factors and margins
- Derating
- Screening self-tests
- Timed replacements

Reliability ≠ Safety

Reliability vs. Safety

- Safety and reliability are NOT the same
  - Sometimes increasing one can even decrease the other.
  - Making all the components highly reliable will have no impact on systematic faults.
- For relatively simple, electro-mechanical systems with primarily component failure accidents, reliability engineering can increase safety.
- But accidents in high-tech systems are changing their nature, and we must change our approaches to safety accordingly.