Verlässliche Echtzeitsysteme – Können wir unseren Autos noch vertrauen?

Bernhard Sechser
Method Park Software AG, Erlangen
30.04.2012

Contents

- Who is Method Park?
- Why do we need Safety Standards?
- Process and Safety demands in Automotive
- Hazard Analysis and Risk Assessment
- Functional and Technical Development
- Software Process in detail
- Tool Qualification
- Summary

Contents

- **Who is Method Park?**
- Why do we need Safety Standards?
- Process and Safety demands in Automotive
- Hazard Analysis and Risk Assessment
- Functional and Technical Development
- Software Process in detail
- Tool Qualification
- Summary

Method Park - Facts and Figures

methodpark

Facts

- Founded in 2001
- Locations:
  Germany: Erlangen, Munich
  USA: Detroit, Miami

Awards

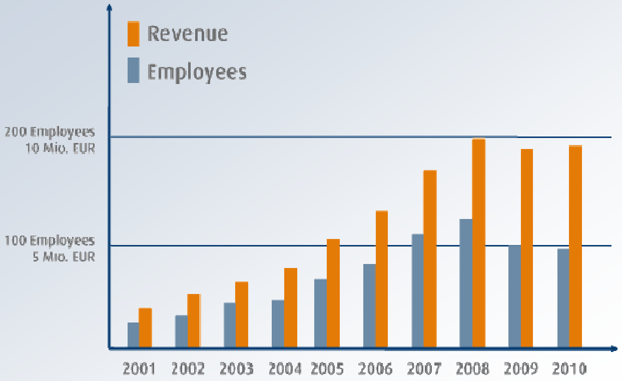
TOP JOB
Top-Arbeitgeber im deutschen Mittelstand
2004
2008
2011

IHK GRÜNDERPREIS
2005

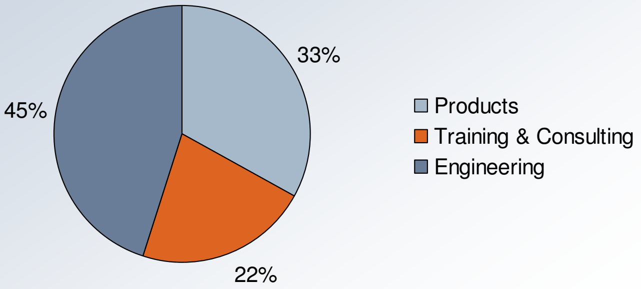
2006, 2007, 2009
BAYERNS BEST 50

GREAT PLACE TO WORK® INSTITUTE DEUTSCHLAND
Deutschlands BESTE Arbeitgeber 2009
Handelsblatt
2009

Revenue & employees



Business unit revenue



Portfolio

Product

stages

Solution for integrated process management

Engineering

Areas:

- Project Coaching
- Software Development & Support
- On Site Support
- Off Site Projects
- Fixed Price Projects

Consulting/Coaching

Topics:

- Software Process Improvement
- CMMI®, SPICE, Automotive SPICE®
- AUTOSAR, Functional Safety
- Requirements Management
- Project and Quality Management
- Software Architecture & Design
- Software Testing

Training

Wide range of seminars in the division system and software development

Accredited by the following organizations: SEI, ISTQB, ISQI, INTACS, IREP

Our Customers

methodpark

Automotive

- Audi
- Automotive Lighting
- Blaupunkt
- BMW
- Bosch
- Brose
- Continental
- Daimler
- Delphi
- ETAS
- Helbako
- IAV
- Knorr-Brakes
- Marquardt
- Peiker Acustic
- Preh
- Thales
- TRW
- Volkswagen
- Webasto
- ZF
- Zollner

Engineering/ Automation

- 7 layers
- ABB
- BDT
- Carl Schenk
- EBM Papst
- Heidelberger Druckmaschinen
- Insta
- Kratzer Automation
- Magirus
- Mettler Toledo
- Mühlbauer Group
- Rohde&Schwarz
- Siemens Industries
- Wago

Government/Public

- Bundesagentur für Arbeit
- Curiavant
- Kassenärztliche Vereinigung Baden-Württemberg

Healthcare

- Carl Zeiss
- Siemens
- Fresenius
- Agfa
- Ziehm Imaging
- NewTec
- Innovations Software
- Technology

IT/ Telecommunications

- GFT
- Intersoft
- Nash Technologies
- NEC
- Micronas
- Siemens
- Teleca

Defense

- Airbus Deutschland
- Diehl
- EADS
- Raytheon Anschütz
- KID

Further

- Bosch und Siemens Hausgeräte
- Deutsche Post
- GMC Software Technologies
- Kodak
- Landesbank Kiel
- Raab Karcher
- Giesecke & Devrient
- Thales Rail Signaling

Contents

- Who is Method Park?
- **Why do we need Safety Standards?**
- Process and Safety demands in Automotive
- Hazard Analysis and Risk Assessment
- Functional and Technical Development
- Software Process in detail
- Tool Qualification
- Summary

Examples

Source: ESA

Ariane 5 (July 4th, 1996)

Detonation shortly after takeoff because of an error in the control software

Root cause:
Insufficient tests of a reused "proven in use" software component

Source: YouTube

Examples

Application that can cause harm (a risk):

- Airbag exploding when infant is sitting in front seat

Need to assess the risk

- Infant getting injured – “not good at all”

Find a mitigation strategy, e.g. a safety function:

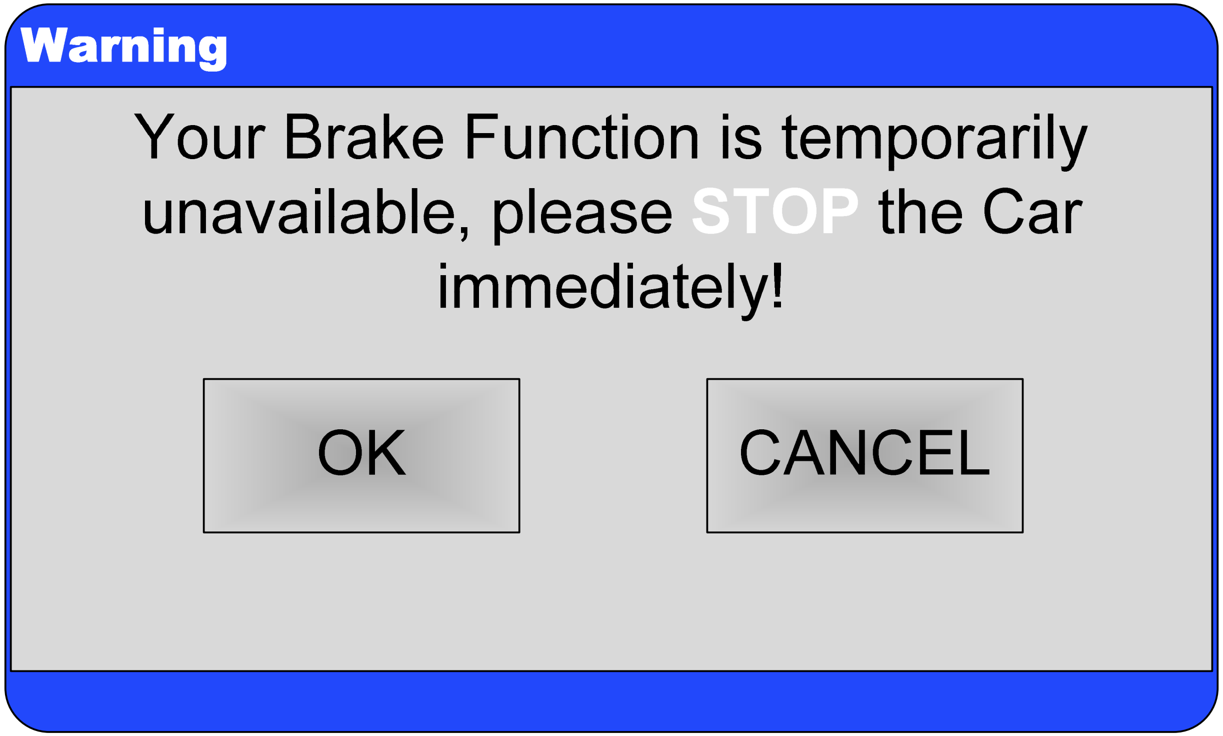
- Detecting infant in front seat and disabling airbag
  - a) sensor delivers signal to
  - b) software/hardware controlling an
  - c) actuator (disabler)

Functional Safety is then:

- An infant in front seat is not exposed to an unacceptable (unreasonable) risk

Question:
How to measure and agree on the measures?

Examples

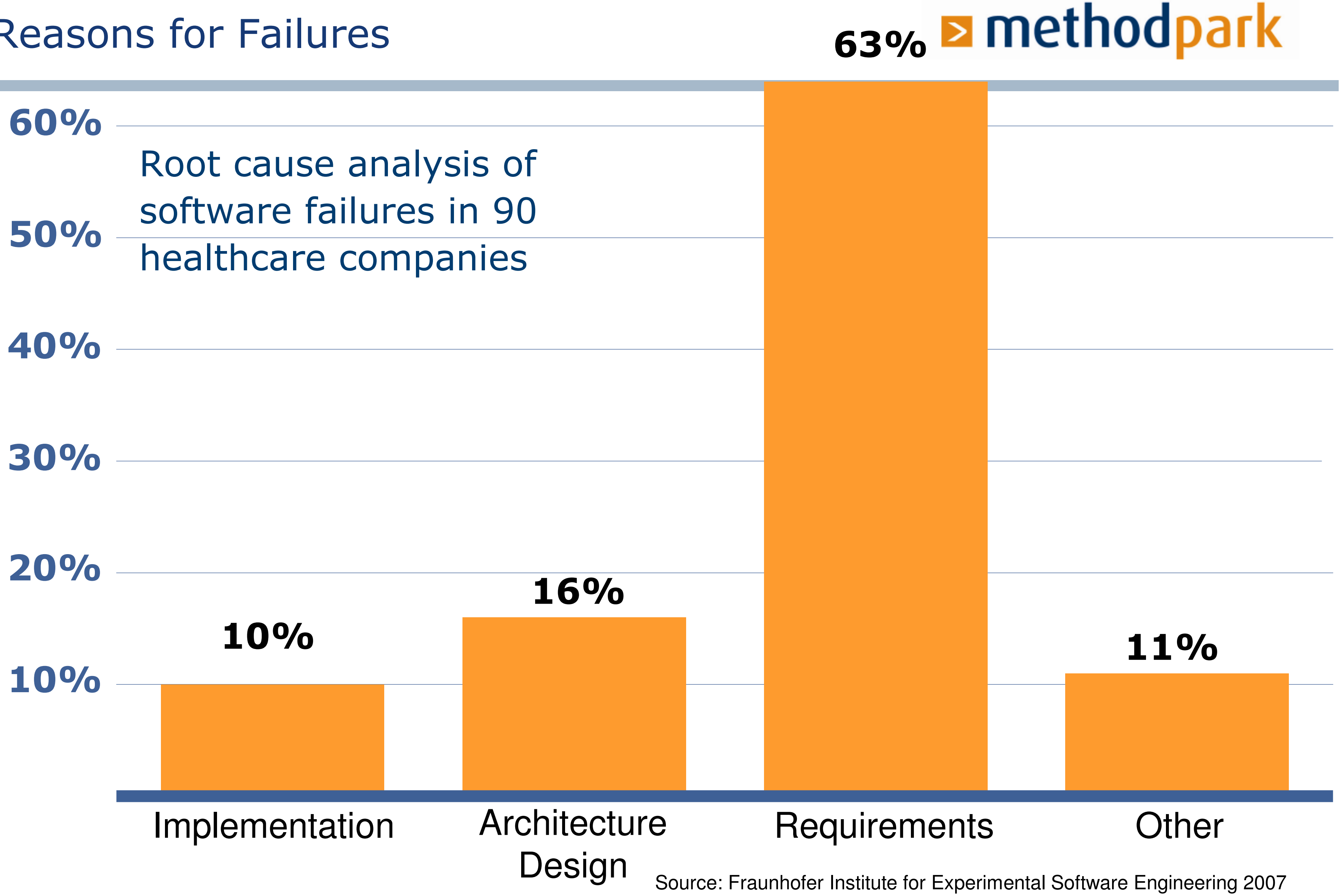


Warning

Your Brake Function is temporarily unavailable, please STOP the Car immediately!

Question:
Do we dare putting software in direct control of people’s life?

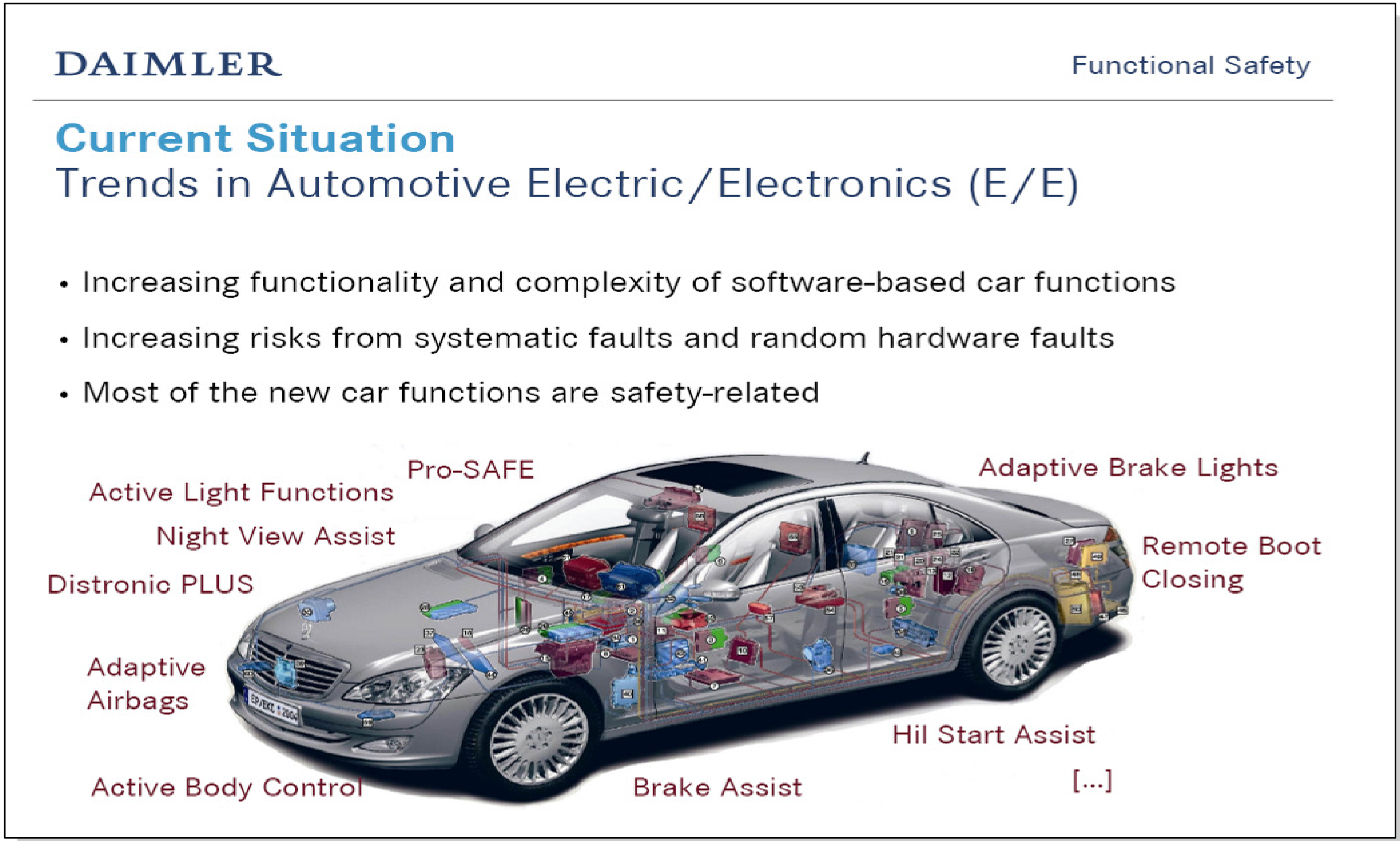
Reasons for Failures



Root cause analysis of software failures in 90 healthcare companies

Source: Fraunhofer Institute for Experimental Software Engineering 2007

Complexity



DAIMLER

Functional Safety

Current Situation

Trends in Automotive Electric/Electronics (E/E)

- Increasing functionality and complexity of software-based car functions
- Increasing risks from systematic faults and random hardware faults
- Most of the new car functions are safety-related

Pro-SAFE

Active Light Functions

Night View Assist

Distronic PLUS

Adaptive Airbags

Active Body Control

Brake Assist

Adaptive Brake Lights

Remote Boot Closing

Hil Start Assist

[...]

Source: © Courtesy of Daimler; Presentation given at Automotive Electronics and Electrical Systems Forum 2008, May 6, 2008, Stuttgart, Germany

Extract from German law

§ 823 Abs. 1 BGB:

„Anyone who injures intentionally or negligently the life, body, health, liberty, property or any other right of another person, is obliged to compensate for the resulting damages."

§ 1 Abs. 1 ProdhaftG:

„If someone is killed, his body or health injured or an item damaged by a defect in a product, the manufacturer of the product is obliged to replace the resulting damages."

Contents

- Who is Method Park?
- Why do we need Safety Standards?
- **Process & Safety demands in Automotive**
- Hazard Analysis and Risk Assessment
- Functional and Technical Development
- Software Process in detail
- Tool Qualification
- Summary

Definitions

Safety

… is the absence of unacceptable (unreasonable) risks that can cause harm achieved through a planned strategy

Functional Safety

… is part of the overall safety that depends on a system or equipment operating correctly in response to its inputs.

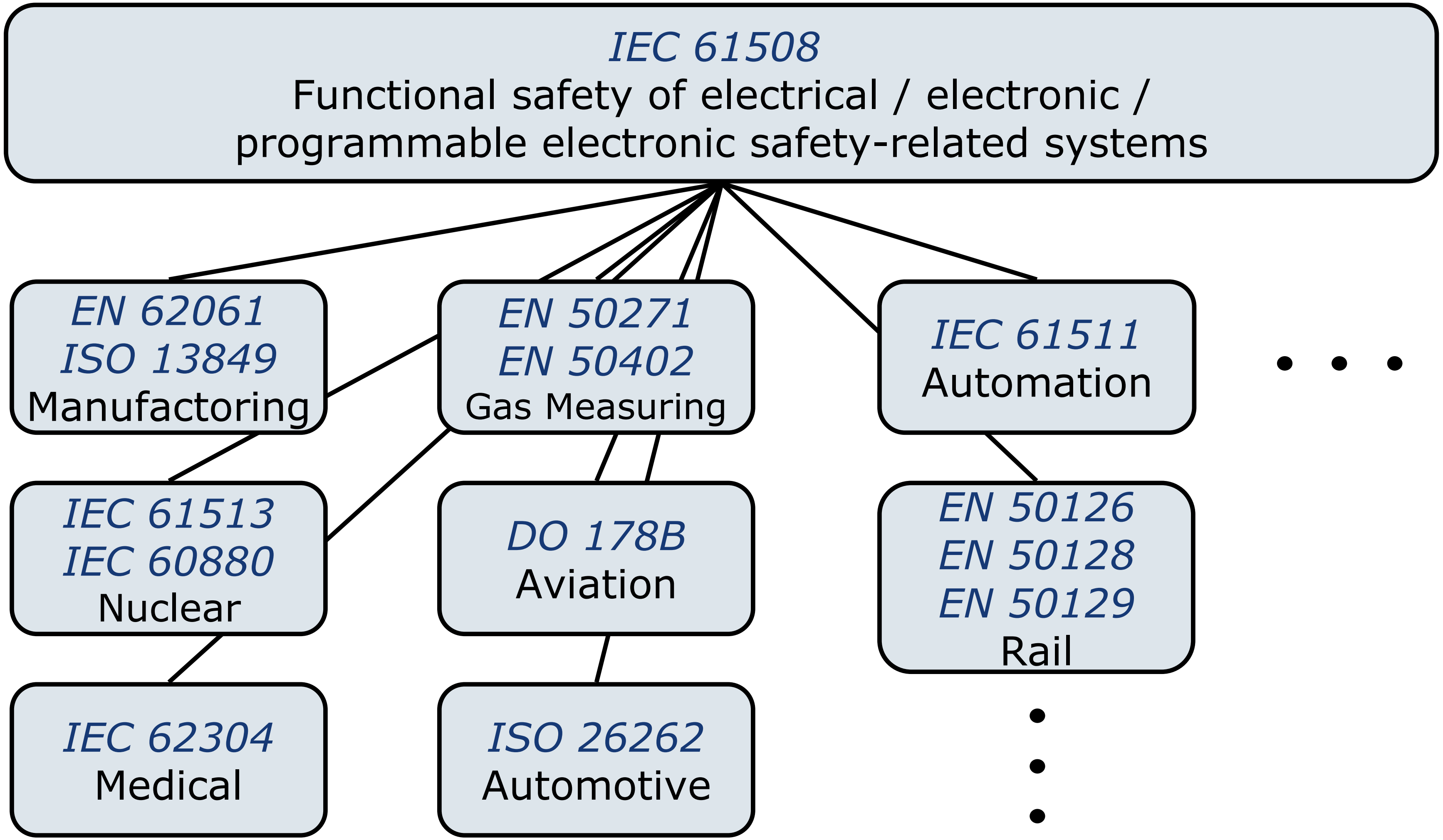
… is achieved when every specified safety function is carried out and the level of performance required of each safety function is met

… is **not** to provide the perfect car, but a safe car.

Functional Safety Management

… is the management (plan, do, act, check) of all activities necessary to reach functional safety.

Existing Standards



IEC 61508
Functional safety of electrical / electronic / programmable electronic safety-related systems

EN 62061
ISO 13849
Manufactoring

EN 50271
EN 50402
Gas Measuring

IEC 61511
Automation

. . .

IEC 61513
IEC 60880
Nuclear

DO 178B
Aviation

EN 50126
EN 50128
EN 50129
Rail

IEC 62304
Medical

ISO 26262
Automotive

Scope of ISO 26262

Why not using IEC 61508?

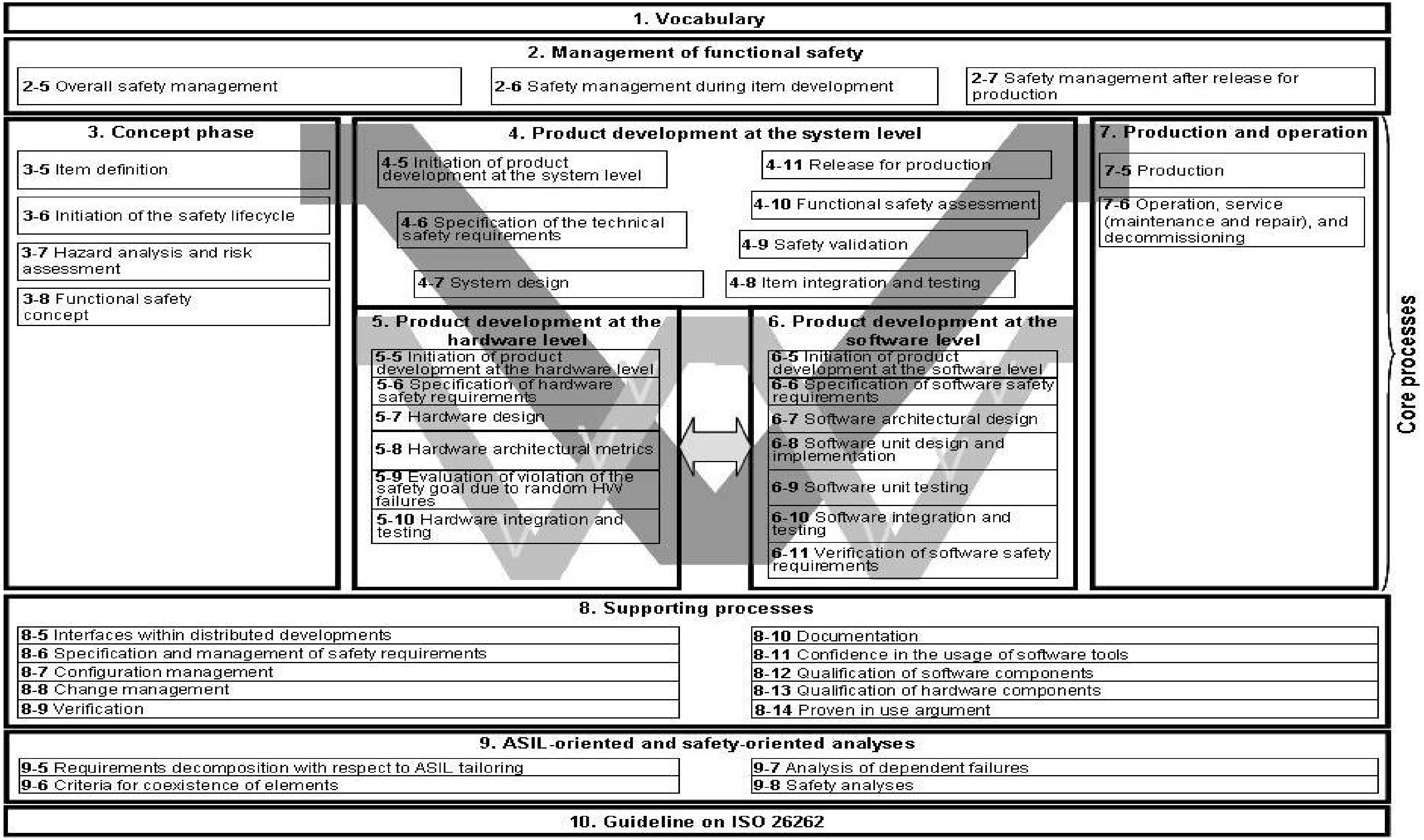
Lessons learnt from application of IEC 61508 in automotive industry:

- Not adapted to real-time and integrated embedded systems
- Not adapted to automotive development and life cycles
- No requirements for manufacturer / supplier relationship
- No ‘consumer-goods’ orientation
- …

Companies had to solve these issues themselves until introduction of

ISO 26262

Structure of ISO 26262



1. Vocabulary

2. Management of functional safety
- 2-5 Overall safety management
- 2-6 Safety management during item development
- 2-7 Safety management after release for production

3. Concept phase
- 3-5 Item definition
- 3-6 Initiation of the safety lifecycle
- 3-7 Hazard analysis and risk assessment
- 3-8 Functional safety concept

4. Product development at the system level
- 4-5 Initiation of product development at the system level
- 4-6 Specification of the technical safety requirements
- 4-7 System design
- 4-8 Item integration and testing
- 4-9 Safety validation
- 4-10 Functional safety assessment
- 4-11 Release for production

5. Product development at the hardware level
- 5-5 Initiation of product development at the hardware level
- 5-6 Specification of hardware safety requirements
- 5-7 Hardware design
- 5-8 Hardware architectural metrics
- 5-9 Evaluation of violation of the safety goal due to random HW failures
- 5-10 Hardware integration and testing

6. Product development at the software level
- 6-5 Initiation of product development at the software level
- 6-6 Specification of software safety requirements
- 6-7 Software architectural design
- 6-8 Software unit design and implementation
- 6-9 Software unit testing
- 6-10 Software integration and testing
- 6-11 Verification of software safety requirements

7. Production and operation
- 7-5 Production
- 7-6 Operation, service (maintenance and repair), and decommissioning

Core processes

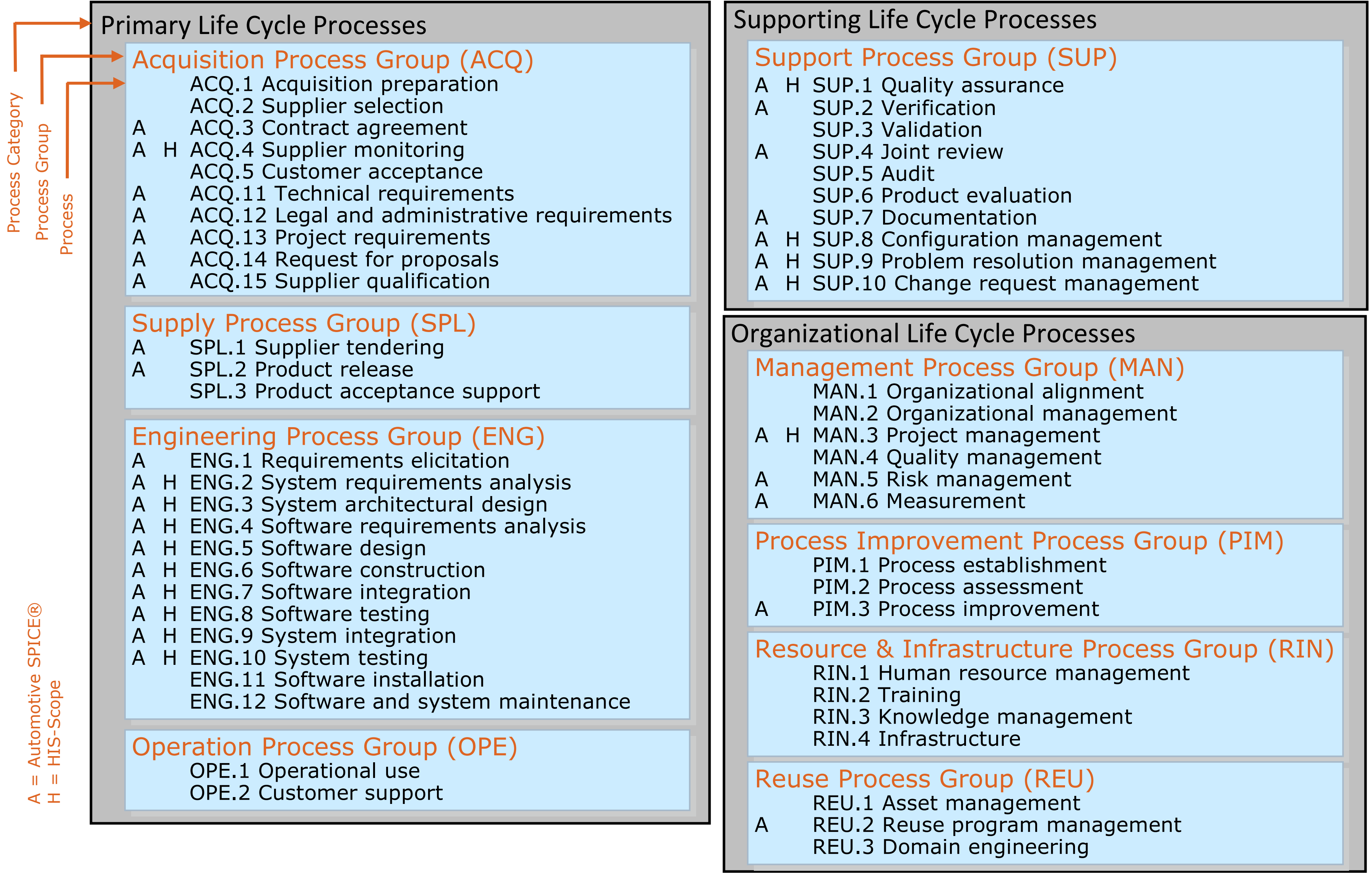
8. Supporting processes
- 8-5 Interfaces within distributed developments
- 8-6 Specification and management of safety requirements
- 8-7 Configuration management
- 8-8 Change management
- 8-9 Verification
- 8-10 Documentation
- 8-11 Confidence in the usage of software tools
- 8-12 Qualification of software components
- 8-13 Qualification of hardware components
- 8-14 Proven in use argument

9. ASIL-oriented and safety-oriented analyses
- 9-5 Requirements decomposition with respect to ASIL tailoring
- 9-6 Criteria for coexistence of elements
- 9-7 Analysis of dependent failures
- 9-8 Safety analyses

10. Guideline on ISO 26262

Source: ISO/FDIS 26262 - BL18

ISO 15504 & Automotive SPICE®



Primary Life Cycle Processes

Acquisition Process Group (ACQ)

ACQ.1 Acquisition preparation
ACQ.2 Supplier selection
A ACQ.3 Contract agreement
A H ACQ.4 Supplier monitoring
ACQ.5 Customer acceptance
A ACQ.11 Technical requirements
A ACQ.12 Legal and administrative requirements
A ACQ.13 Project requirements
A ACQ.14 Request for proposals
A ACQ.15 Supplier qualification

Supply Process Group (SPL)

A SPL.1 Supplier tendering
A SPL.2 Product release
SPL.3 Product acceptance support

Engineering Process Group (ENG)

A ENG.1 Requirements elicitation
A H ENG.2 System requirements analysis
A H ENG.3 System architectural design
A H ENG.4 Software requirements analysis
A H ENG.5 Software design
A H ENG.6 Software construction
A H ENG.7 Software integration
A H ENG.8 Software testing
A H ENG.9 System integration
A H ENG.10 System testing
ENG.11 Software installation
ENG.12 Software and system maintenance

Operation Process Group (OPE)

OPE.1 Operational use
OPE.2 Customer support

Supporting Life Cycle Processes

Support Process Group (SUP)

A H SUP.1 Quality assurance
A SUP.2 Verification
SUP.3 Validation
A SUP.4 Joint review
SUP.5 Audit
SUP.6 Product evaluation
A SUP.7 Documentation
A H SUP.8 Configuration management
A H SUP.9 Problem resolution management
A H SUP.10 Change request management

Organizational Life Cycle Processes

Management Process Group (MAN)

MAN.1 Organizational alignment
MAN.2 Organizational management
A H MAN.3 Project management
MAN.4 Quality management
A MAN.5 Risk management
A MAN.6 Measurement

Process Improvement Process Group (PIM)

PIM.1 Process establishment
PIM.2 Process assessment
A PIM.3 Process improvement

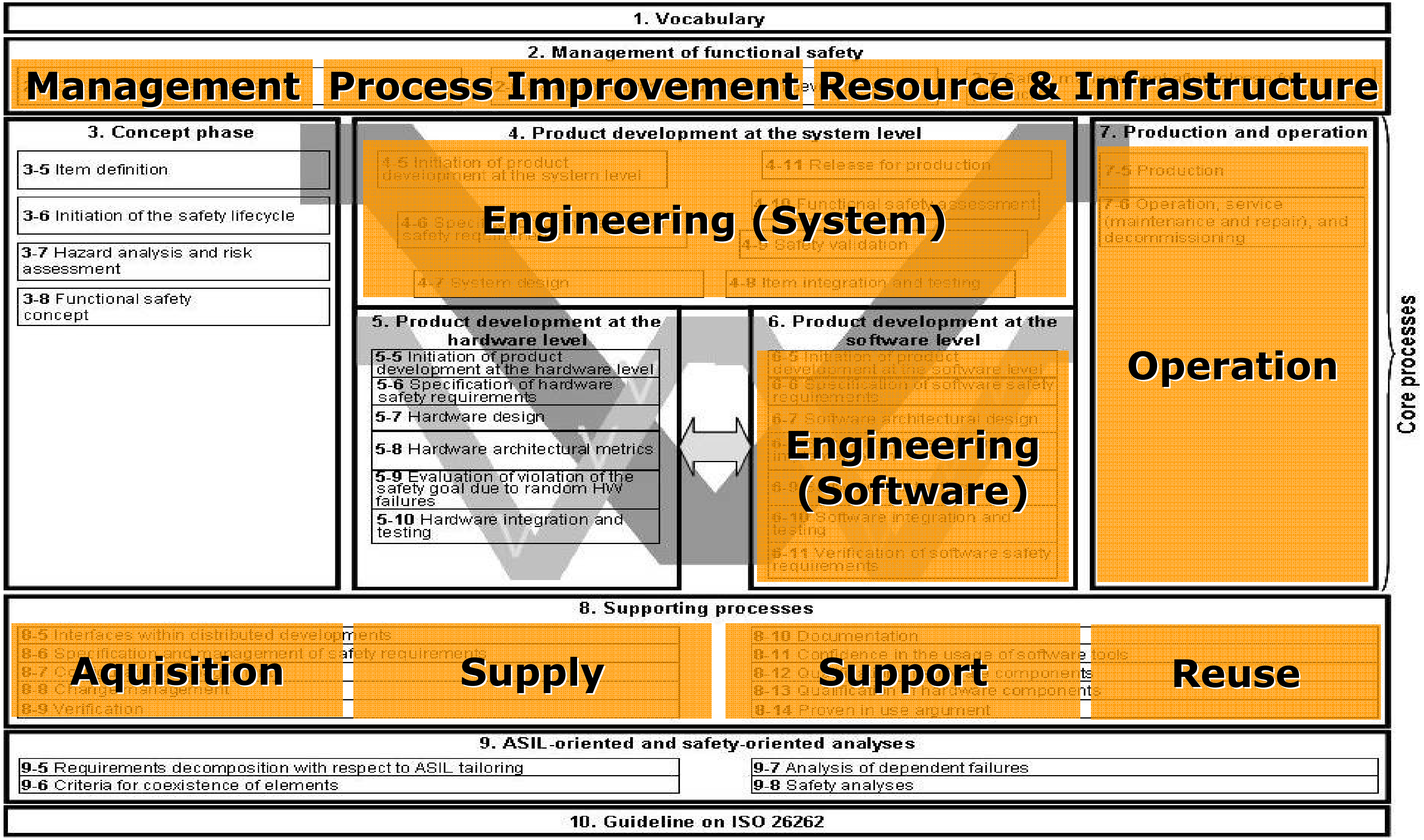
Resource & Infrastructure Process Group (RIN)

RIN.1 Human resource management
RIN.2 Training
RIN.3 Knowledge management
RIN.4 Infrastructure

Reuse Process Group (REU)

REU.1 Asset management
A REU.2 Reuse program management
REU.3 Domain engineering

Structure of ISO 26262



1. Vocabulary

2. Management of functional safety

Management | Process Improvement | Resource & Infrastructure

3. Concept phase

3-5 Item definition

3-6 Initiation of the safety lifecycle

3-7 Hazard analysis and risk assessment

3-8 Functional safety concept

4. Product development at the system level

Engineering (System)

4-5 Initiation of product development at the system level

4-6 Spec... safety requ...

4-7 System design

4-8 Item integration and testing

4-9 Safety validation

4-10 Functional safety assessment

4-11 Release for production

5. Product development at the hardware level

5-5 Initiation of product development at the hardware level

5-6 Specification of hardware safety requirements

5-7 Hardware design

5-8 Hardware architectural metrics

5-9 Evaluation of violation of the safety goal due to random HW failures

5-10 Hardware integration and testing

6. Product development at the software level

Engineering (Software)

6-5 Initiation of product development at the software level

6-6 Specification of software safety requirements

6-7 Software architectural design

6-10 Software integration and testing

6-11 Verification of software safety requirements

7. Production and operation

Operation

7-5 Production

7-6 Operation, service (maintenance and repair), and decommissioning

Core processes

8. Supporting processes

Aquisition | Supply | Support | Reuse

8-5 Interfaces within distributed developments

8-6 Specification and management of safety requirements

8-7 C...

8-8 Change management

8-9 Verification

8-10 Documentation

8-11 Confidence in the usage of software tools

8-12 Q... components

8-13 Qualificat... hardware components

8-14 Proven in use argument

9. ASIL-oriented and safety-oriented analyses

9-5 Requirements decomposition with respect to ASIL tailoring

9-6 Criteria for coexistence of elements

9-7 Analysis of dependent failures

9-8 Safety analyses

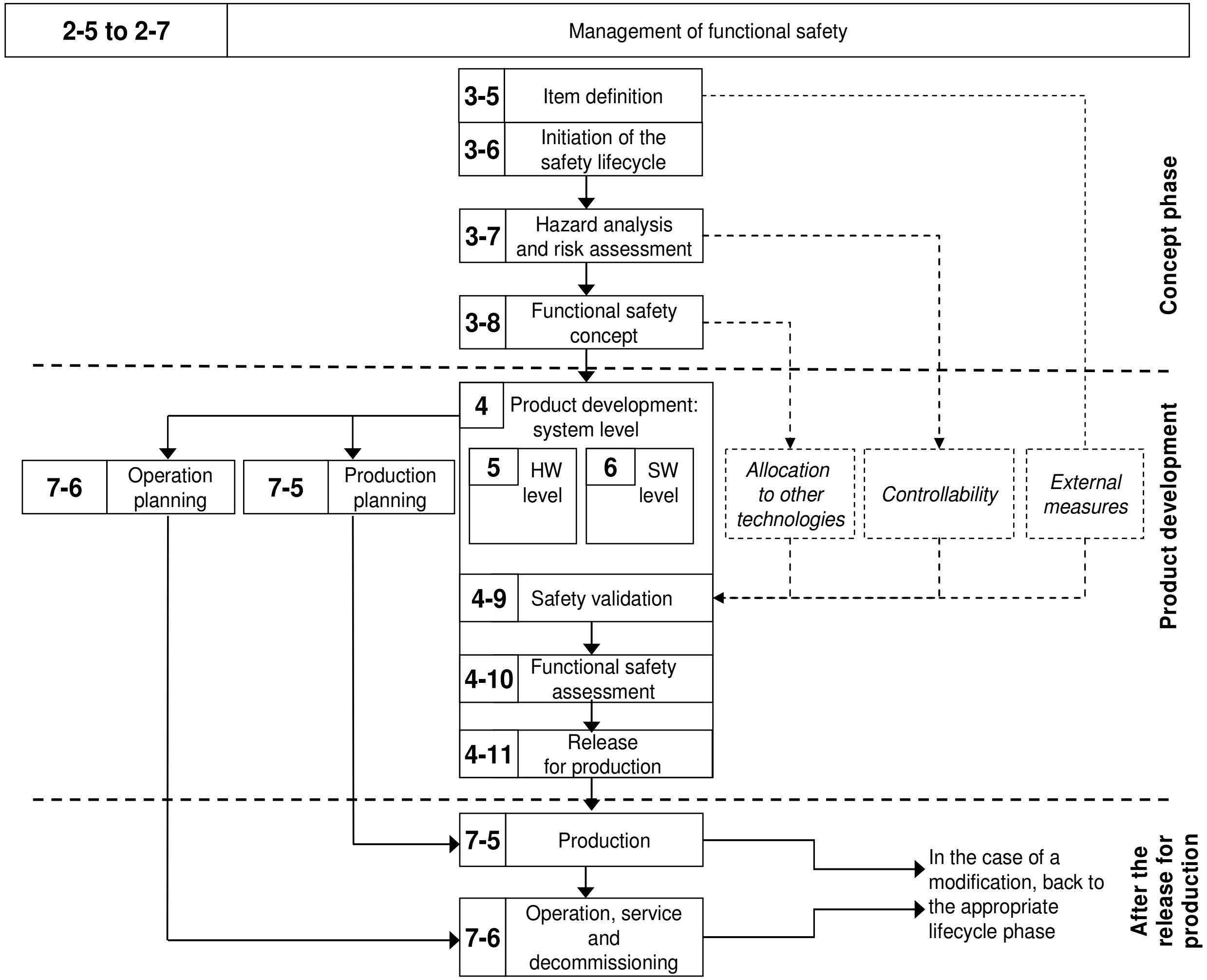
10. Guideline on ISO 26262

ISO 15504 Process Groups

Source: ISO/FDIS 26262 - BL18

Safety Lifecycle Overview

- Concept
- Development
- Production

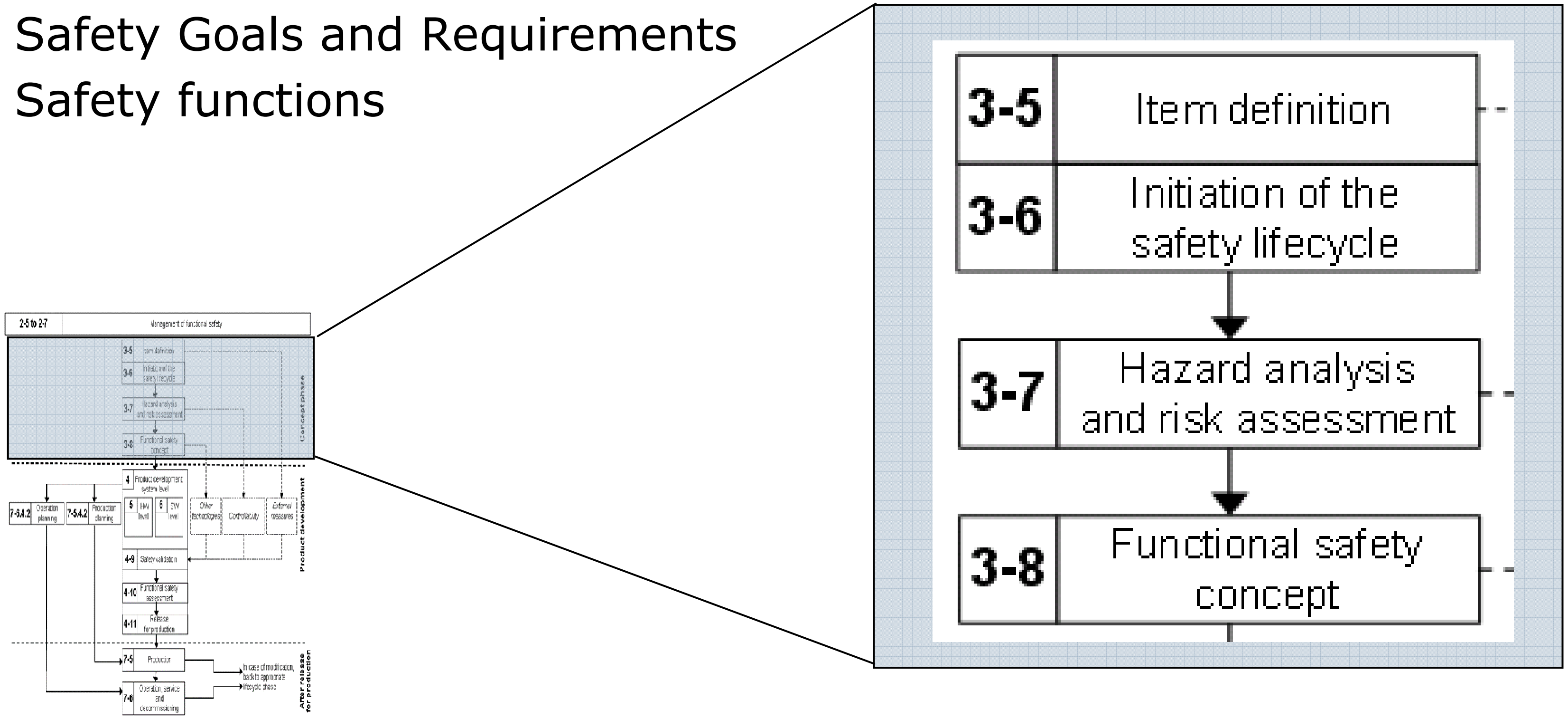

2-5 to 2-7
Management of functional safety
3-5
Item definition
3-6
Initiation of the safety lifecycle
3-7
Hazard analysis and risk assessment
3-8
Functional safety concept
Concept phase
4
Product development: system level
5
HW level
6
SW level
7-6
Operation planning
7-5
Production planning
Allocation to other technologies
Controllability
External measures
4-9
Safety validation
4-10
Functional safety assessment
4-11
Release for production
Product development
7-5
Production
7-6
Operation, service and decommissioning
In the case of a modification, back to the appropriate lifecycle phase
After the release for production

Source: ISO/FDIS 26262-2 – BL18

Safety Lifecycle Overview

Concept Phase

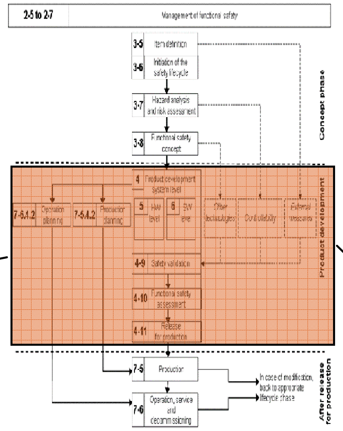
- Focus on entire system
- Risks
- Safety Goals and Requirements
- Safety functions

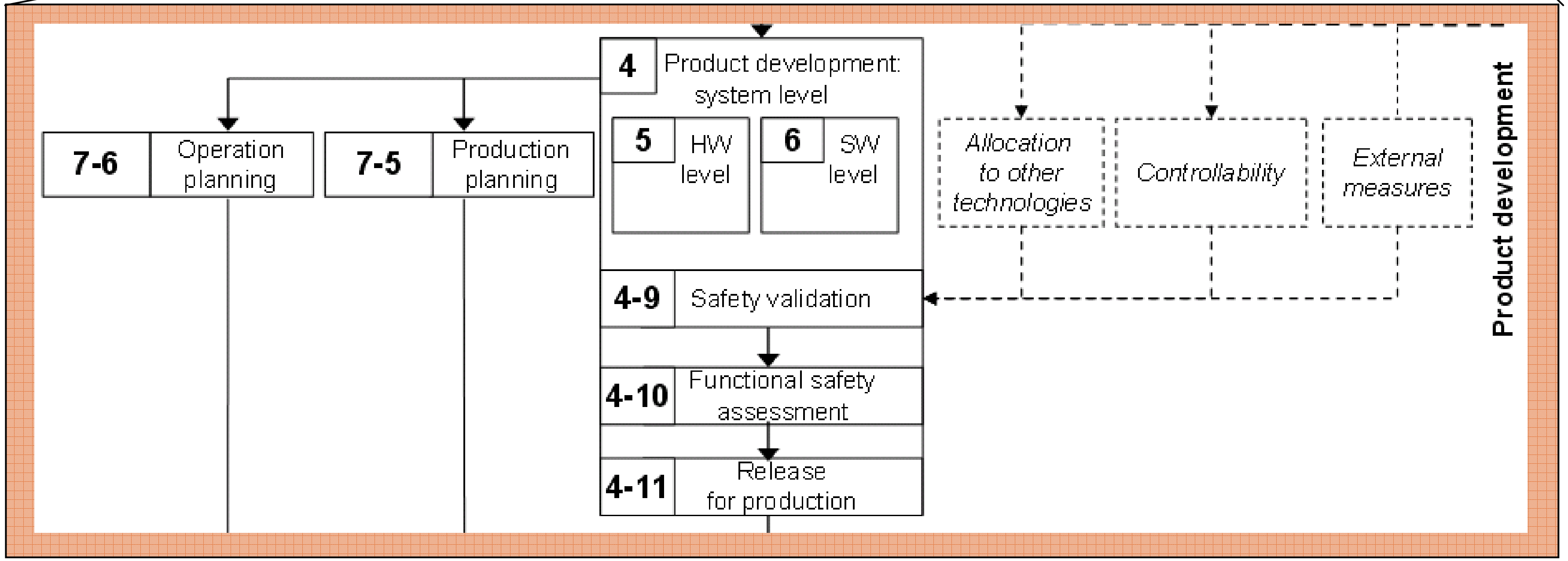

2-5 to 2-7
Management of functional safety
3-5 Item definition
3-6 Initiation of the safety lifecycle
3-7 Hazard analysis and risk assessment
3-8 Functional safety concept
Concept phase
4 Product development system level
5 HW level
6 SW level
Other technologies
Controllability
External measures
7-6.4.2 Operation planning
7-5.4.2 Production planning
4-9 Safety validation
4-10 Functional safety assessment
4-11 Release for production
Product development
7-5 Production
7-6 Operation, service and decommissioning
In case of modification, back to approprate lifecycle phase
After release for production
3-5 Item definition
3-6 Initiation of the safety lifecycle
3-7 Hazard analysis and risk assessment
3-8 Functional safety concept

Safety Lifecycle Overview

Product Development

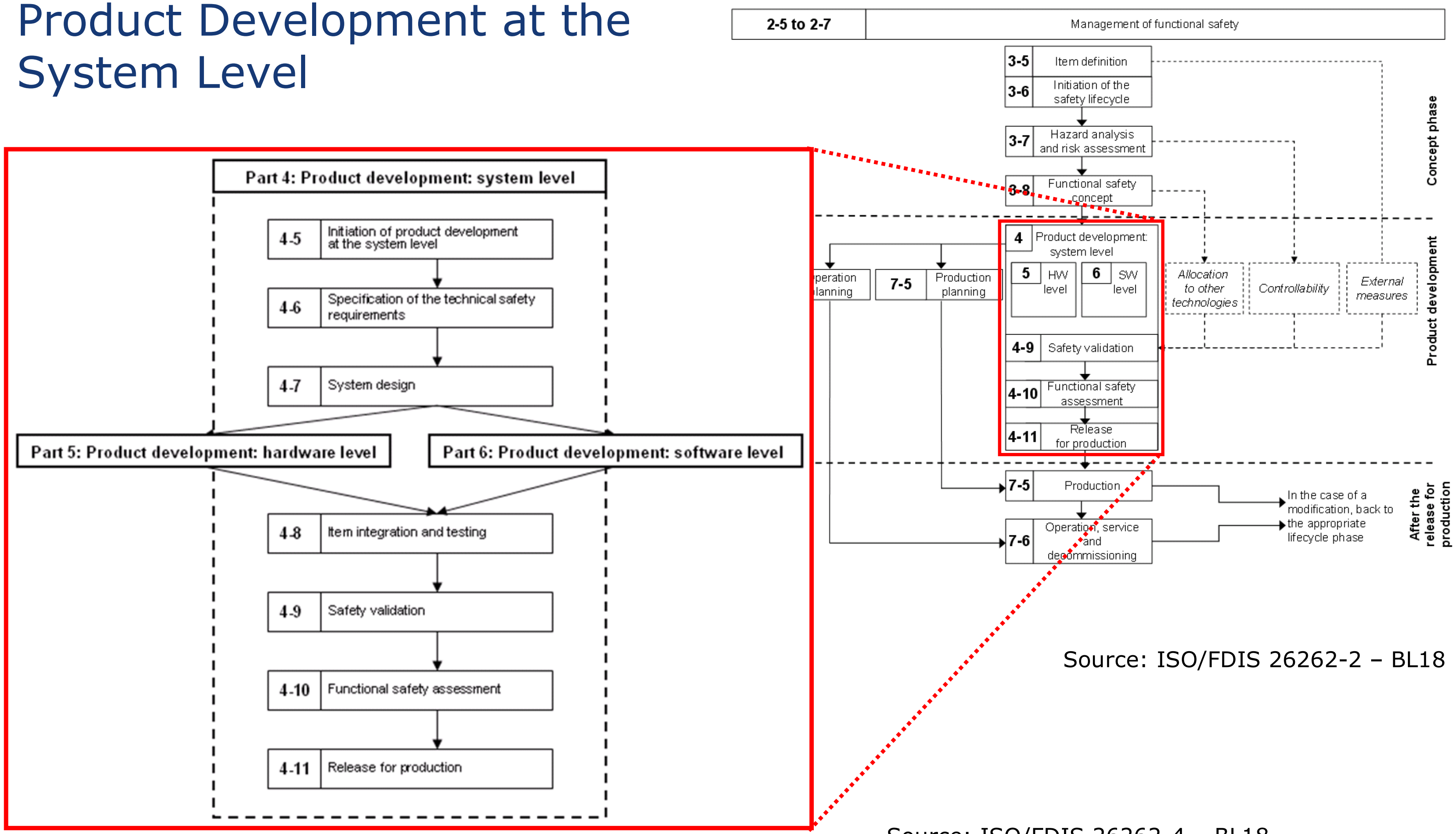
- System, Hardware and Software
- Safety validation and assessment
- Production and Operation (Planning)


2-5 to 2-7
Management of functional safety
3-5
Item definition
3-6
Initiation of the safety lifecycle
3-7
Hazard analysis and risk assessment
3-8
Functional safety concept
Concept phase
4
Product development system level
7-6.4.2
Operation planning
7-5.4.2
Production planning
5
HW level
6
SW level
Other technologies
Controllability
External measures
4-9
Safety validation
4-10
Functional safety assessment
4-11
Release for production
Product development
7-5
Production
7-6
Operation, service and decommissioning
In case of modification, back to appropriate lifecycle phase
After release for production



Product Development

Product Development at the System Level



Part 4: Product development: system level

4-5 Initiation of product development at the system level

4-6 Specification of the technical safety requirements

4-7 System design

Part 5: Product development: hardware level

Part 6: Product development: software level

4-8 Item integration and testing

4-9 Safety validation

4-10 Functional safety assessment

4-11 Release for production

2-5 to 2-7 Management of functional safety

3-5 Item definition

3-6 Initiation of the safety lifecycle

3-7 Hazard analysis and risk assessment

3-8 Functional safety concept

4 Product development: system level

5 HW level

6 SW level

4-9 Safety validation

4-10 Functional safety assessment

4-11 Release for production

peration lanning

7-5 Production planning

Allocation to other technologies

Controllability

External measures

7-5 Production

7-6 Operation, service and decommissioning

In the case of a modification, back to the appropriate lifecycle phase

Concept phase

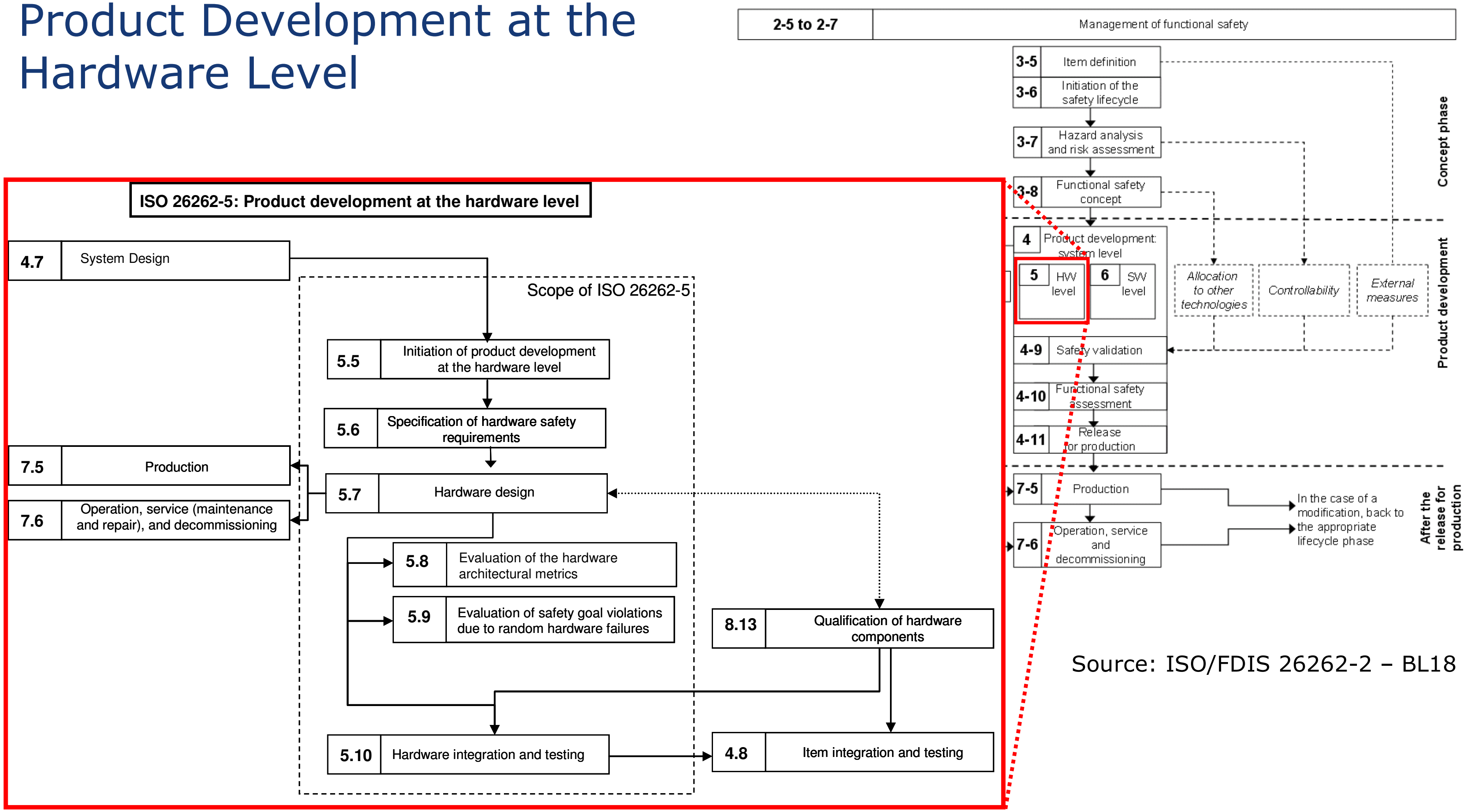
Product development

After the release for production

Source: ISO/FDIS 26262-2 – BL18

Source: ISO/FDIS 26262-4 – BL18

Product Development at the Hardware Level

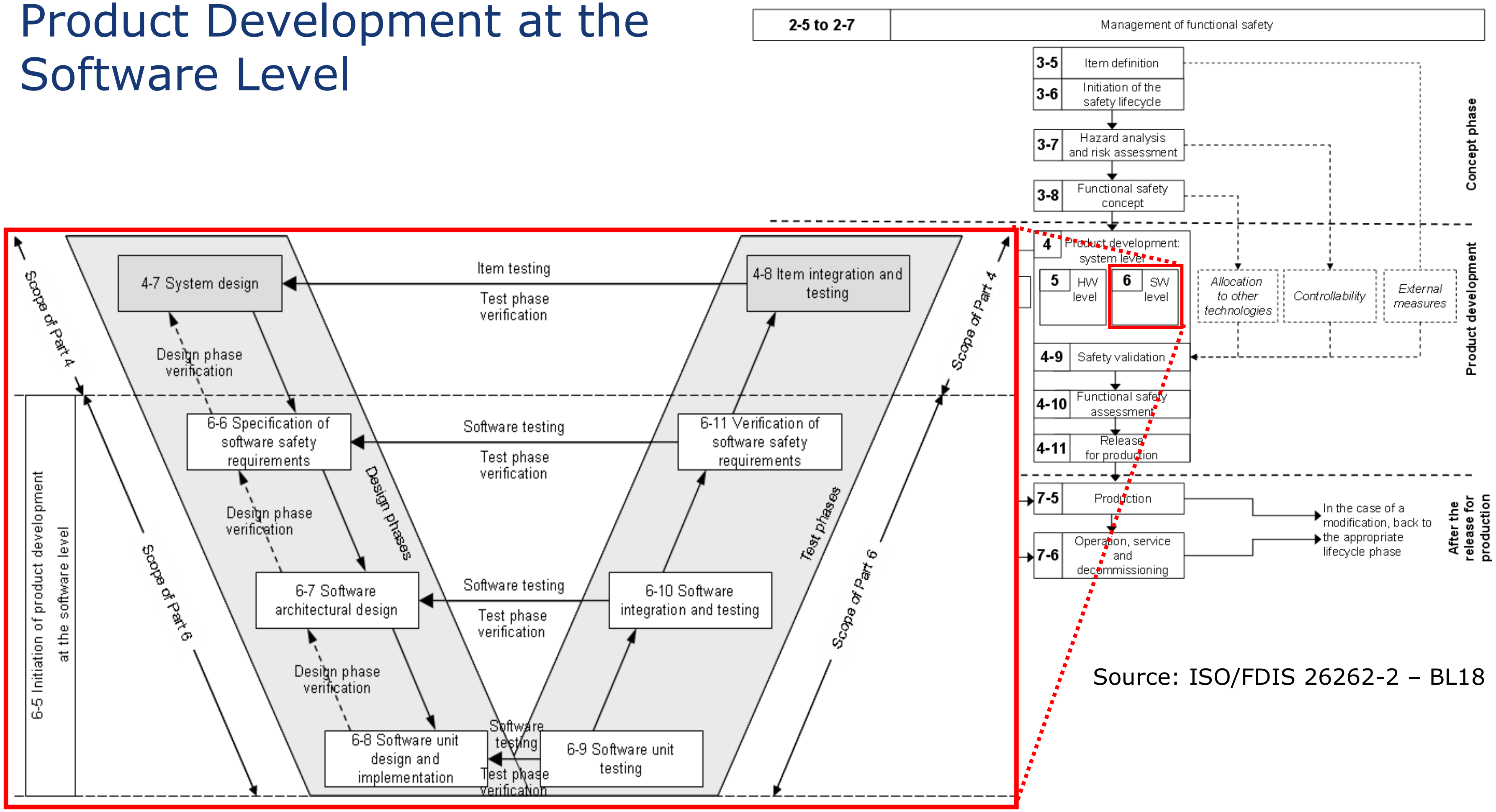

2-5 to 2-7
Management of functional safety
3-5
Item definition
3-6
Initiation of the safety lifecycle
3-7
Hazard analysis and risk assessment
3-8
Functional safety concept
Concept phase
4
Product development: system level
5
HW level
6
SW level
Allocation to other technologies
Controllability
External measures
4-9
Safety validation
4-10
Functional safety assessment
4-11
Release for production
Product development
7-5
Production
7-6
Operation, service and decommissioning
In the case of a modification, back to the appropriate lifecycle phase
After the release for production
ISO 26262-5: Product development at the hardware level
4.7
System Design
Scope of ISO 26262-5
5.5
Initiation of product development at the hardware level
5.6
Specification of hardware safety requirements
7.5
Production
5.7
Hardware design
7.6
Operation, service (maintenance and repair), and decommissioning
5.8
Evaluation of the hardware architectural metrics
5.9
Evaluation of safety goal violations due to random hardware failures
8.13
Qualification of hardware components
5.10
Hardware integration and testing
4.8
Item integration and testing

Source: ISO/FDIS 26262-2 – BL18

Source: ISO/FDIS 26262-5 – BL18

Product Development

Product Development at the Software Level


2-5 to 2-7
Management of functional safety
3-5
Item definition
3-6
Initiation of the safety lifecycle
3-7
Hazard analysis and risk assessment
3-8
Functional safety concept
Concept phase
4
Product development: system level
5
HW level
6
SW level
Allocation to other technologies
Controllability
External measures
4-9
Safety validation
4-10
Functional safety assessment
4-11
Release for production
Product development
7-5
Production
7-6
Operation, service and decommissioning
In the case of a modification, back to the appropriate lifecycle phase
After the release for production
4-7 System design
4-8 Item integration and testing
Item testing
Test phase verification
Design phase verification
6-6 Specification of software safety requirements
6-11 Verification of software safety requirements
Software testing
Test phase verification
Design phase verification
6-7 Software architectural design
6-10 Software integration and testing
Software testing
Test phase verification
Design phase verification
6-8 Software unit design and implementation
Software testing
Test phase verification
6-9 Software unit testing
Design phases
Test phases
Scope of Part 4
Scope of Part 4
Scope of Part 6
Scope of Part 6
6-5 Initiation of product development at the software level

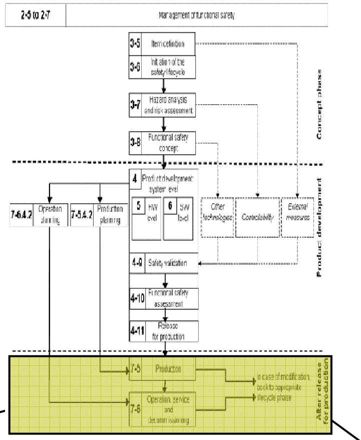
Source: ISO/FDIS 26262-2 – BL18

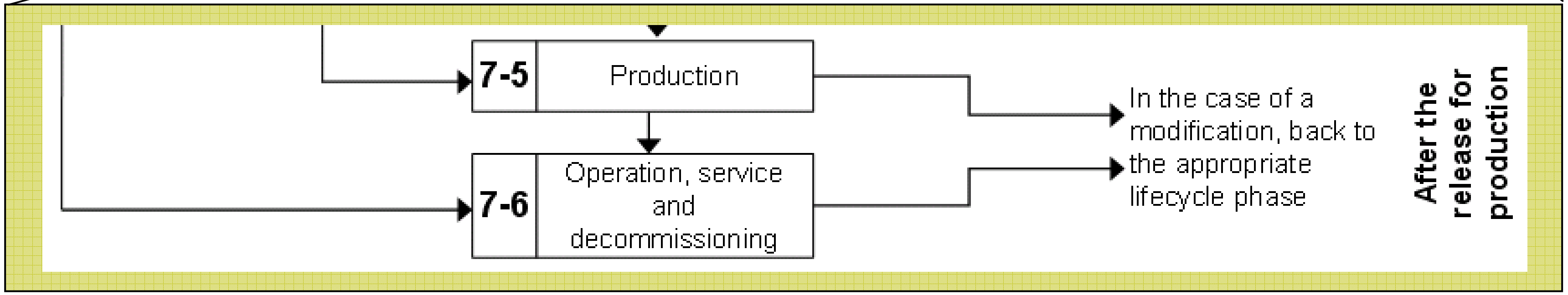
Source: ISO/FDIS 26262-6 – BL18

Safety Lifecycle Overview

After Release for Production

- Production
- Installation
- Operation
- Maintenance and reparation
- Disassembly


2-5 to 2-7
Management of functional safety
3-5 Item definition
3-6 Initiation of the safety lifecycle
3-7 Hazard analysis and risk assessment
3-8 Functional safety concept
Concept phase
4 Product development system level
5 HW level
6 SW level
7-6.4.2 Operation planning
7-5.4.2 Production planning
Other technologies
Controllability
External measures
Product development
4-9 Safety validation
4-10 Functional safety assessment
4-11 Release for production
7-5 Production
7-6 Operation, service and decommissioning
In case of modification, back to appropriate lifecycle phase
After release for production

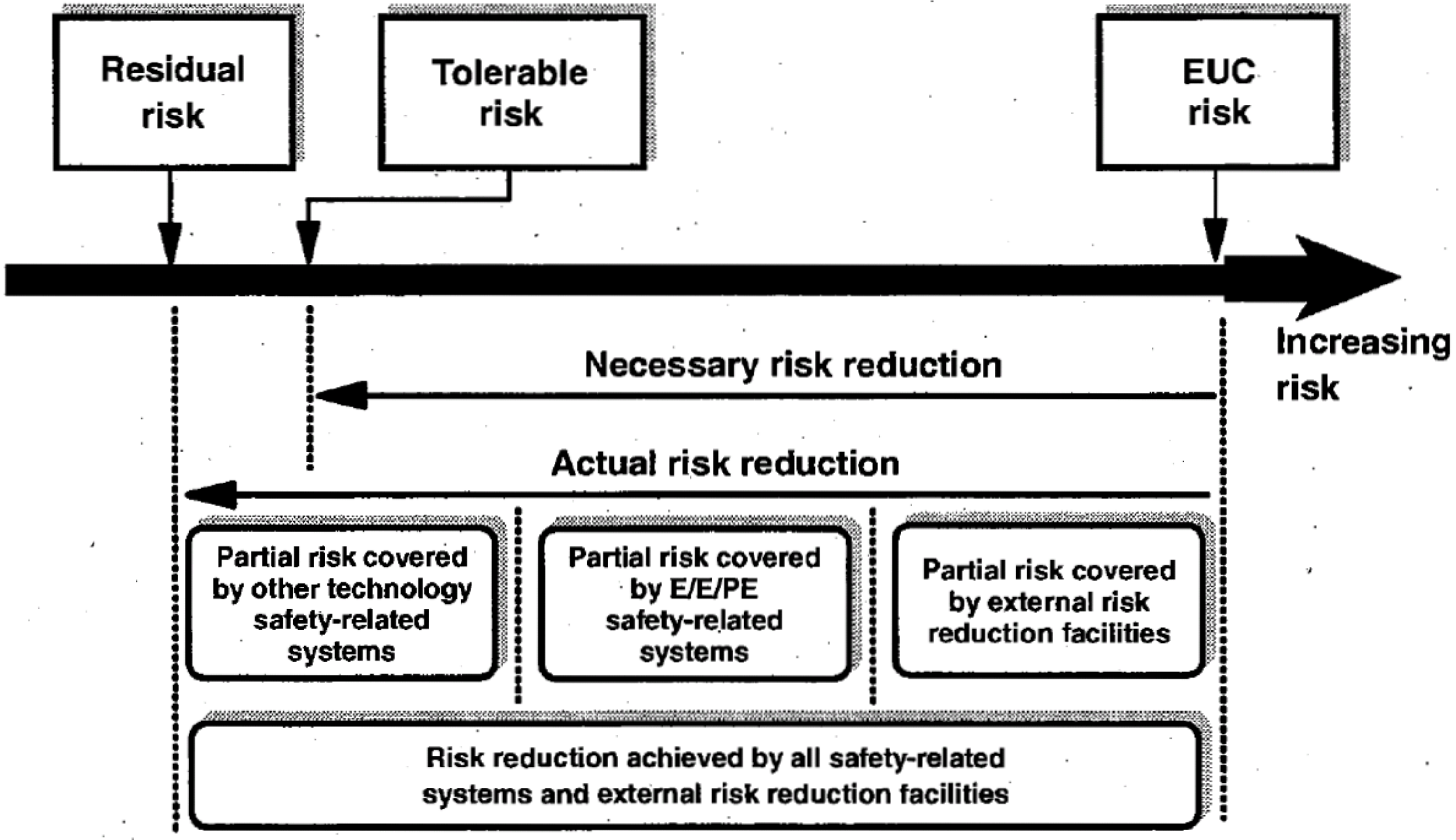

7-5
Production
7-6
Operation, service and decommissioning
In the case of a modification, back to the appropriate lifecycle phase
After the release for production

Contents

- Who is Method Park?
- Why do we need Safety Standards?
- Process and Safety demands in Automotive
- **Hazard Analysis and Risk Assessment**
- Functional and Technical Development
- Software Process in detail
- Tool Qualification
- Summary

Hazard Analysis and Risk Assessment

Risk reduction to an acceptable level


Residual risk
Tolerable risk
EUC risk
Increasing risk
Necessary risk reduction
Actual risk reduction
Partial risk covered by other technology safety-related systems
Partial risk covered by E/E/PE safety-related systems
Partial risk covered by external risk reduction facilities
Risk reduction achieved by all safety-related systems and external risk reduction facilities

Source: IEC 61508-5

Hazard Analysis and Risk Assessment

Situation analysis and hazard identification

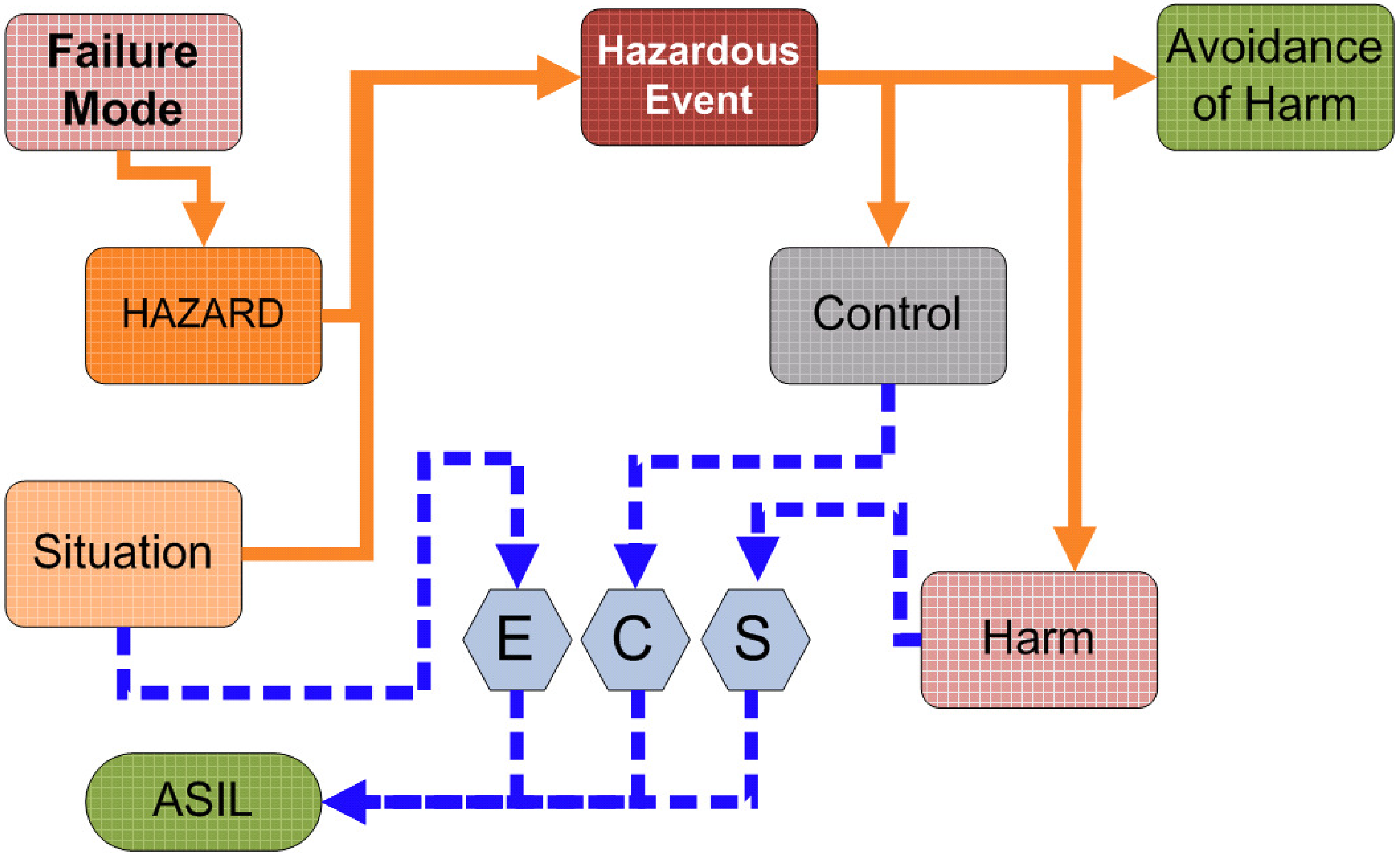
- List of driving and operating situations
  → Estimation of the probability of **E**xposure
- Detailing failure modes leading to hazards in specific situations
  → Estimation of **C**ontrollability
- Evaluating consequences of the hazards
  → Estimation of potential **S**everity

→ Respect only the plain item (do not take risk-reducing measures into account!)

→ Involve persons with good knowledge and domain experience

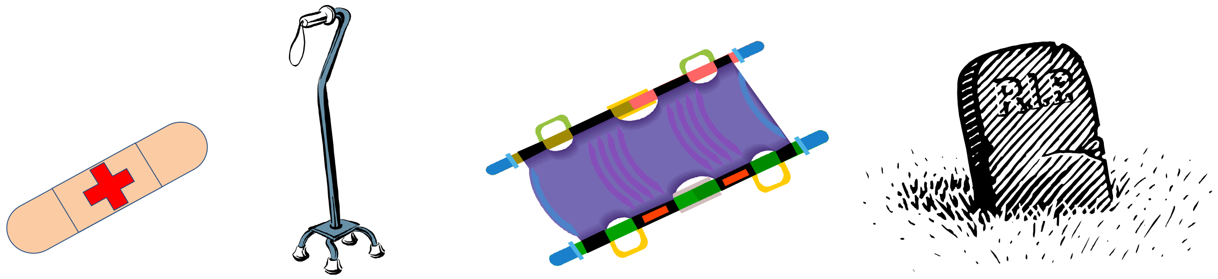
Hazard Analysis and Risk Assessment

Associations of the central concepts



Hazard Analysis and Risk Assessment

Severity – Measure of the extent of harm to an individual in a specific situation



Class	S0	S1	S2	S3
Description	No injuries	Light and moderate injuries	Severe and life-threatening injuries (survival probable)	Life-threatening injuries (survival uncertain), fatal injuries

Hazard Analysis and Risk Assessment

Exposure – State of being in an operational situation that can be hazardous if coincident with the failure mode under analysis

Class	E0	E1	E2	E3	E4
Description	Incredible	Very low probability	Low probability	Medium probability	High probability
Time		Not specified	Less than 1% of average operating time	1% - 10% of average operating time	> 10% of average operating time
Event		Situations that occur less often than once a year for the great majority of drivers	Situations that occur a few times a year for the great majority of drivers	Situations that occur once a month or more often for an average driver	All situations that occur during almost every drive on average

Hazard Analysis and Risk Assessment

Controllability – Avoidance of the specified harm or damage through the timely reactions of the persons involved

Class	C0	C1	C2	C3
Description	Controllable in general	Simply controllable	Normally controllable	Difficult to control or uncontrollable
Definition	Controllable in general	99% or more of all drivers or other traffic participants are usually able to avoid a specific harm.	90% or more of all drivers or other traffic participants are usually able to avoid a specific harm.	Less than 90% of all drivers or other traffic participants are usually able, or barely able, to avoid a specific harm.

Hazard Analysis and Risk Assessment

Combinations of Severity, Exposure and Controllability result in the applicable ASIL.

The ASIL's influence the development process of the items.

QM = Quality Management

No specific ISO 26262 requirement has to be observed

If S0 or E0 or C0 is set, no ASIL is required (QM).

		C1	C2	C3
S1	E1	**QM**	**QM**	**QM**
	E2	**QM**	**QM**	**QM**
	E3	**QM**	**QM**	**A**
	E4	**QM**	**A**	**B**
S2	E1	**QM**	**QM**	**QM**
	E2	**QM**	**QM**	**A**
	E3	**QM**	**A**	**B**
	E4	**A**	**B**	**C**
S3	E1	**QM**	**QM**	**A**
	E2	**QM**	**A**	**B**
	E3	**A**	**B**	**C**
	E4	**B**	**C**	**D**

Hazard Analysis and Risk Assessment

Safety Goals

- Top-level safety requirements as a result of the hazard analysis and risk assessment
- Assigned to each identified hazard rated with an ASIL A-D
- Lead to item characteristics needed to avert hazards or to reduce risks associated with the hazards to an acceptable level

Example for safety goals: Park Brake System

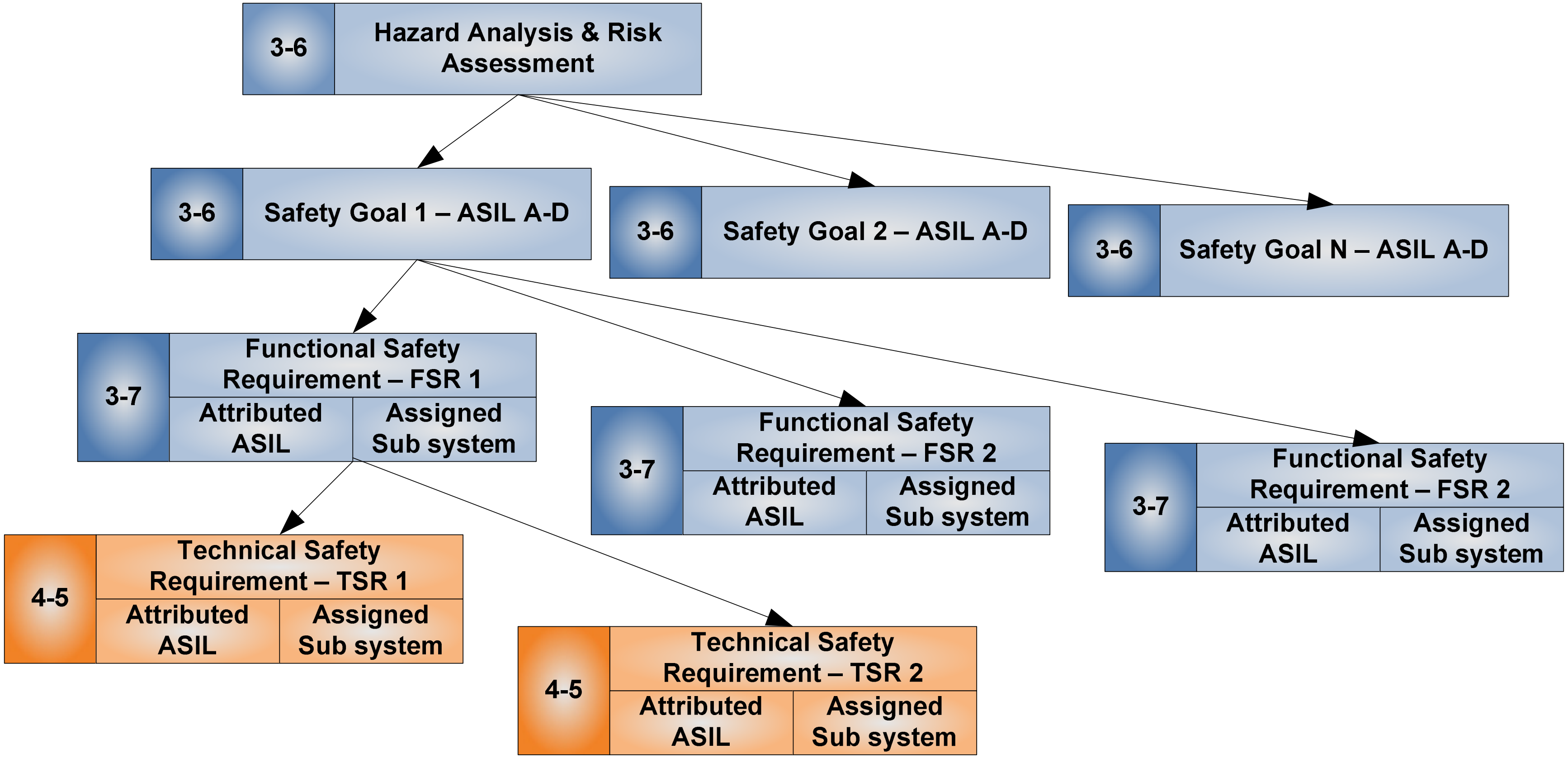
ID	Safety Goal	ASIL
G1	Avoidance of unintended maximum brake force build up at one or several wheels during drive and in all environmental conditions	D
G2	Guarantee the specified parking brake function in use case situation "parking on slope" in all environmental conditions	A
G3	Avoidance of unintended release of the parking brake in use case situation "parking on slope" in all environmental conditions	C

Contents

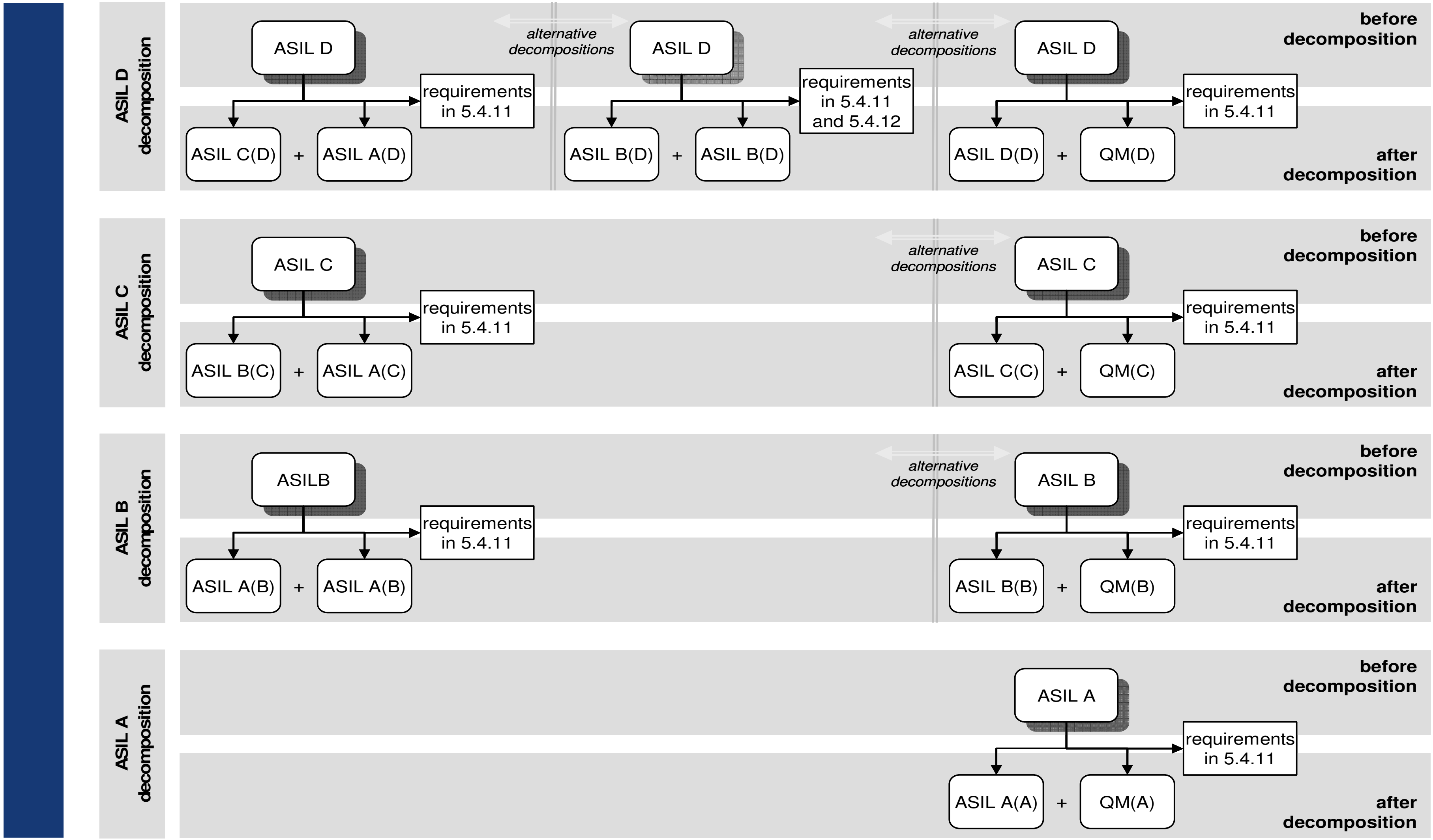
- Who is Method Park?
- Why do we need Safety Standards?
- Process and Safety demands in Automotive
- Hazard Analysis and Risk Assessment
- **Functional and Technical Development**
- Software Process in detail
- Tool Qualification
- Summary

Functional Safety Concept

Safety Goals and Functional Safety Requirements

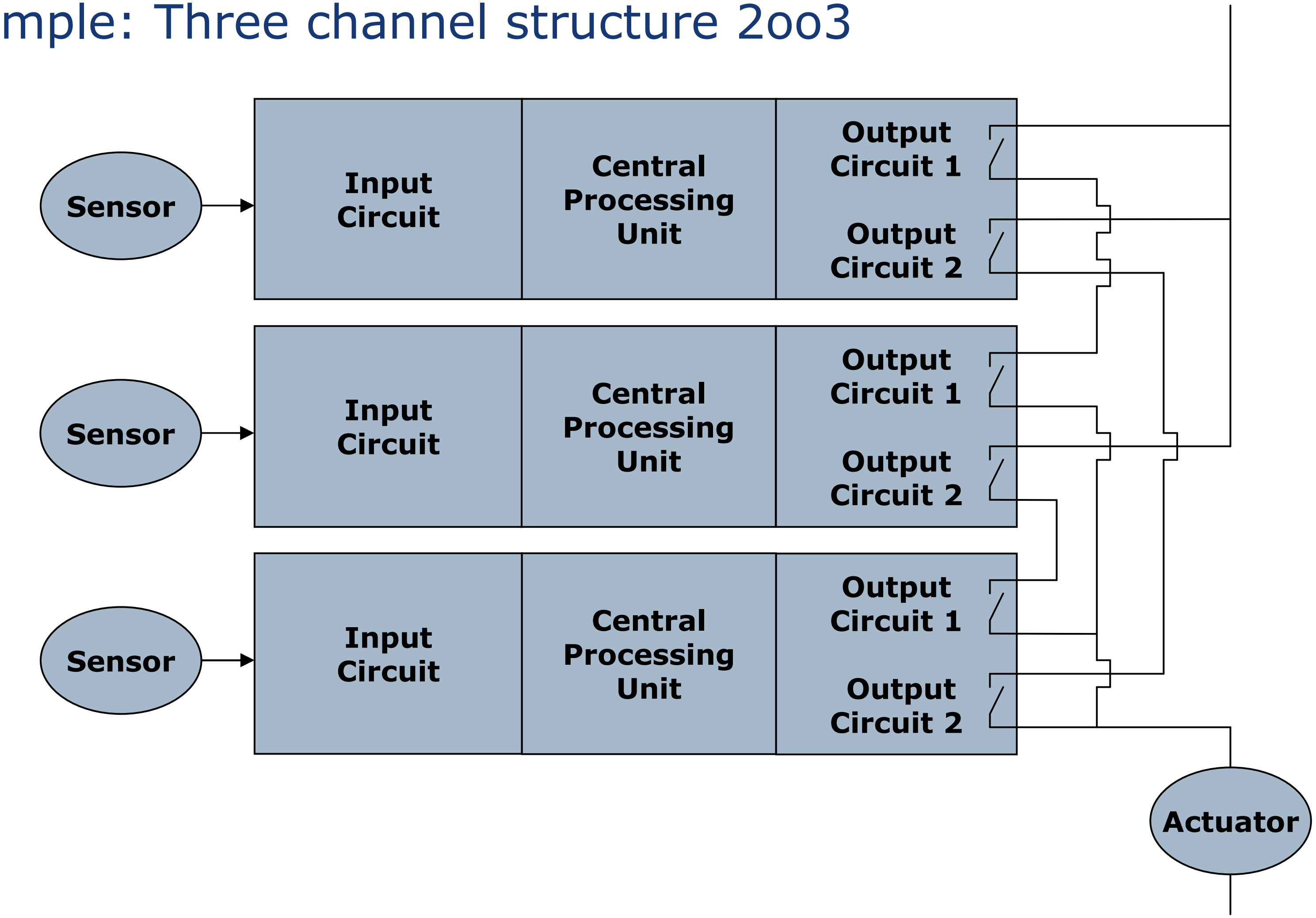


ASIL Decomposition



Source: ISO/FDIS 26262-9 – BL18

Example: Three channel structure 2oo3

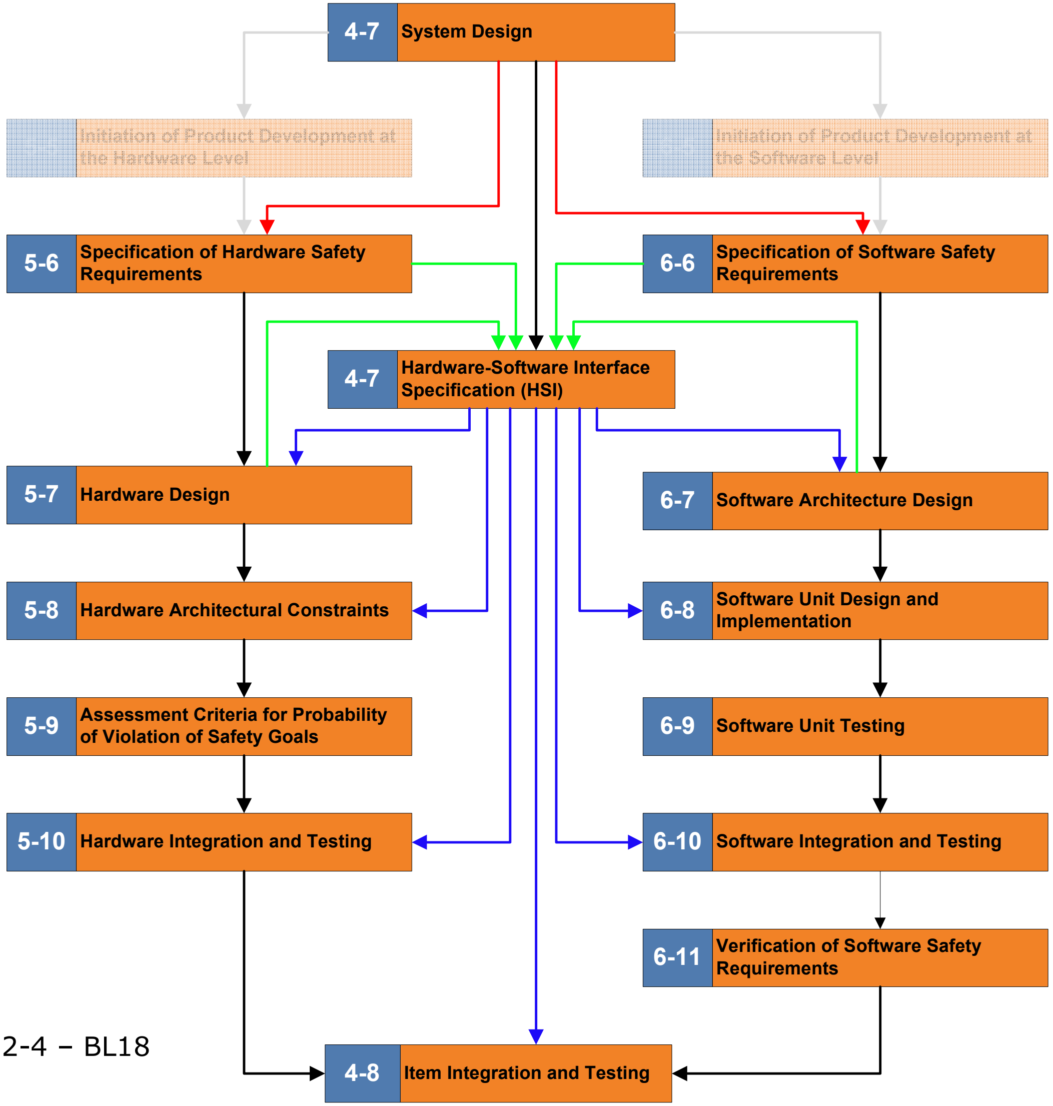


Contents

- Who is Method Park?
- Why do we need Safety Standards?
- Process and Safety demands in Automotive
- Hazard Analysis and Risk Assessment
- Functional and Technical Development
- **Software Process in detail**
- Tool Qualification
- Summary

Product Development at Hardware & Software Level

Important part:
Hardware-Software Interface Specification (HSI)


4-7 System Design
Initiation of Product Development at the Hardware Level
Initiation of Product Development at the Software Level
5-6 Specification of Hardware Safety Requirements
6-6 Specification of Software Safety Requirements
4-7 Hardware-Software Interface Specification (HSI)
5-7 Hardware Design
6-7 Software Architecture Design
5-8 Hardware Architectural Constraints
6-8 Software Unit Design and Implementation
5-9 Assessment Criteria for Probability of Violation of Safety Goals
6-9 Software Unit Testing
5-10 Hardware Integration and Testing
6-10 Software Integration and Testing
6-11 Verification of Software Safety Requirements
4-8 Item Integration and Testing

Source: ISO/FDIS 26262-4 – BL18

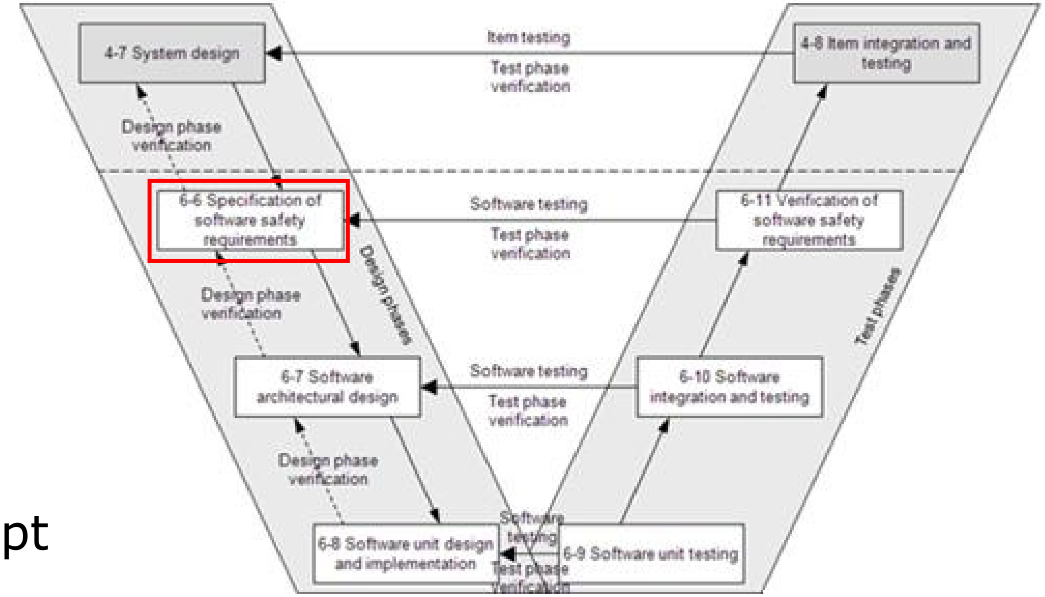
Initiation of Product Development at the Software Level

Topics to be covered by modeling and coding guidelines

Topics		ASIL			
		A	B	C	D
1a	Enforcement of low complexity	++	++	++	++
1b	Use of language subsets	++	++	++	++
1c	Enforcement of strong typing	++	++	++	++
1d	Use of defensive implementation techniques	o	+	++	++
1e	Use of established design principles	+	+	+	++
1f	Use of unambiguous graphical representation	+	++	++	++
1g	Use of style guides	+	++	++	++
1h	Use of naming conventions	++	++	++	++

Source: ISO/FDIS 26262-6:2011

Specification of Software Safety Requirements


4-7 System design
Item testing
Test phase verification
4-8 Item integration and testing
Design phase verification
6-6 Specification of software safety requirements
Software testing
Test phase verification
6-11 Verification of software safety requirements
Design phases
Design phase verification
Test phases
6-7 Software architectural design
Software testing
Test phase verification
6-10 Software integration and testing
Design phase verification
Software testing
6-8 Software unit design and implementation
Test phase verification
6-9 Software unit testing

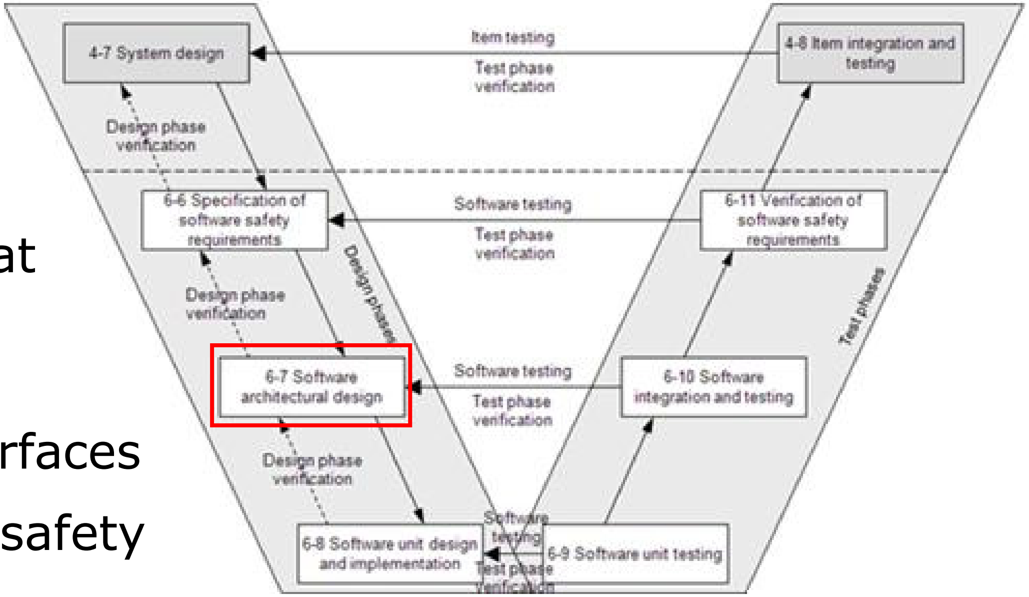
Goals

- Derive Software Safety Requirements from and ensure consistency with
  - System Design
  - Technical Safety Concept
- Detail the hardware-software interface requirements

Software Architectural Design

Goals

- Develop an Architecture that implements the Software Safety Requirements
  - Static and dynamic interfaces
  - Safety-related and non safety related requirements


4-7 System design
Item testing
Test phase verification
4-8 Item integration and testing
Design phase verification
6-6 Specification of software safety requirements
Software testing
Test phase verification
6-11 Verification of software safety requirements
Design phases
Test phases
Design phase verification
6-7 Software architectural design
Software testing
Test phase verification
6-10 Software integration and testing
Design phase verification
6-8 Software unit design and implementation
Software testing
Test phase verification
6-9 Software unit testing

- Verify the Software Architecture
  - Compliance with the requirements
  - Compatibility with hardware
  - Respect of design principles and standards

Software Architectural Design

Principles for software architectural design

Methods		ASIL			
		A	B	C	D
1a	Hierarchical structure of software components	++	++	++	++
1b	Restricted size of software components	++	++	++	++
1c	Restricted size of interfaces	+	+	+	+
1d	High cohesion within each software component	+	++	++	++
1e	Restricted coupling between software components	+	++	++	++
1f	Appropriate scheduling properties	++	++	++	++
1g	Restricted use of interrupts	+	+	+	++

Source: ISO/FDIS 26262-6:2011

Software Architectural Design

Based on the results of the safety analysis the mechanisms for error detection and error handling shall be applied

Methods		ASIL			
		A	B	C	D
1a	Range checks of input and output data	++	++	++	++
1b	Plausibility check	+	+	+	++
1c	Detection of data errors	+	+	+	+
1d	External monitoring facility	o	+	+	++
1e	Control flow monitoring	o	+	++	++
1f	Diverse software design	o	o	+	++

Error detection

Methods		ASIL			
		A	B	C	D
1a	Static recovery mechanism	+	+	+	+
1b	Graceful degradation	+	+	++	++
1c	Independent parallel redundancy	o	o	+	++
1d	Correcting codes for data	+	+	+	+

Error handling

Source: ISO/FDIS 26262-6:2011

Software Architectural Design

Methods for the verification of the software architectural design

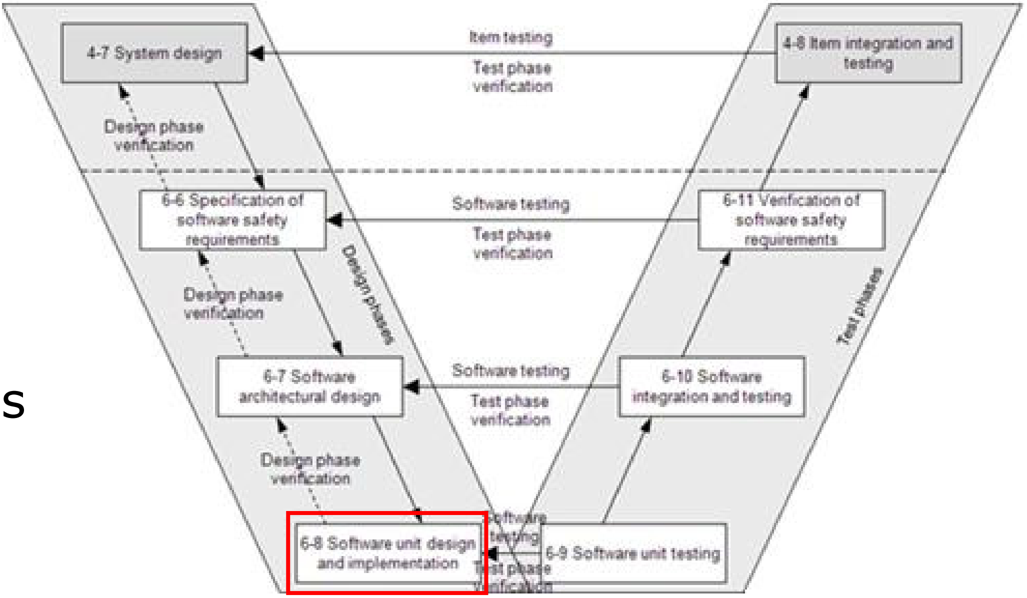
Methods		ASIL			
		A	B	C	D
1a	Walk-through of the design	++	+	o	o
1b	Inspection of the design	+	++	++	++
1c	Simulation of dynamic parts of the design	+	+	+	++
1d	Prototype generation	o	o	+	++
1e	Formal verification	o	o	+	+
1f	Control flow analysis	+	+	++	++
1g	Data flow analysis	+	+	++	++

Source: ISO/FDIS 26262-6:2011

Software Unit Design and Implementation

Goals

- Specify SW Units based on:
  - SW Architecture
  - SW Safety Requirements
- Implement the SW Units
- Verify SW Units
  - Code reviews / inspections


4-7 System design
Item testing
Test phase verification
4-8 Item integration and testing
Design phase verification
6-6 Specification of software safety requirements
Software testing
Test phase verification
6-11 Verification of software safety requirements
Design phases
Test phases
Design phase verification
6-7 Software architectural design
Software testing
Test phase verification
6-10 Software integration and testing
Design phase verification
6-8 Software unit design and implementation
Software testing
Test phase verification
6-9 Software unit testing

Software Unit Design and Implementation

Design principles for software unit design and implementation

Methods		ASIL			
		A	B	C	D
1a	One entry and one exit point in subprograms and functions	++	++	++	++
1b	No dynamic objects or variables, or else online test during their creation	+	++	++	++
1c	Initialization of variables	++	++	++	++
1d	No multiple use of variable names	+	++	++	++
1e	Avoid global variables or else justify their usage	+	+	++	++
1f	Limited use of pointers	o	+	+	++
1g	No implicit type conversions	+	++	++	++
1h	No hidden data flow or control flow	+	++	++	++
1i	No unconditional jumps	++	++	++	++
1j	No recursions	+	+	++	++

Source: ISO/FDIS 26262-6:2011

Software Unit Design and Implementation

Example: MISRA C

- Programming standard developed by Motor Industry Software Reliability Association
- Avoidance of runtime errors due to unsafe C constructs
- The respect of MISRA C shall be demonstrated ➔ static code analysis

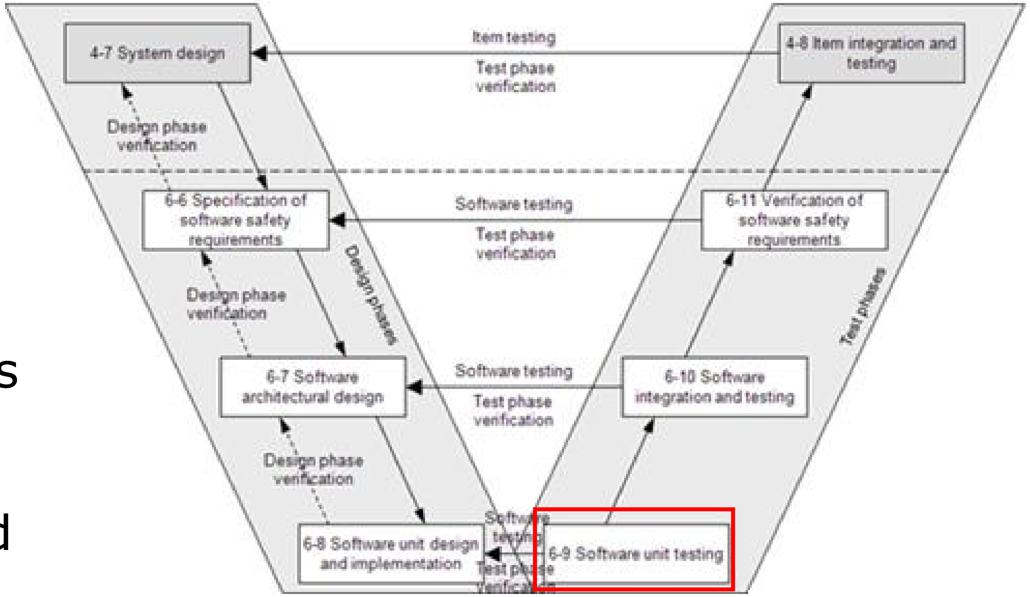
Infos: www.**misra**.org

Software Unit Testing

Goals

- Demonstrate that the software units fulfil the Software Unit Specifications

- Verify absence of undesired functionalities


4-7 System design
Item testing
Test phase verification
4-8 Item integration and testing
Design phase verification
6-6 Specification of software safety requirements
Software testing
Test phase verification
6-11 Verification of software safety requirements
Design phases
Test phases
Design phase verification
6-7 Software architectural design
Software testing
Test phase verification
6-10 Software integration and testing
Design phase verification
6-8 Software unit design and implementation
Software testing
Test phase verification
6-9 Software unit testing

Software Unit Testing

The software unit testing methods shall be applied to demonstrate that the software units achieve:

- Compliance with the software unit design specification
- Compliance with the specification of the hardware-software interface
- Correct implementation of the functionality
- Absence of unintended functionality
- Robustness
- Sufficiency of the resources to support the functionality

Methods		ASIL			
		A	B	C	D
1a	Requirements-based test	++	++	++	++
1b	Interface test	++	++	++	++
1c	Fault injection test	+	+	+	++
1d	Resource usage test	+	+	+	++
1e	Back-to-back comparison test between model and code, if applicable	+	+	++	++

Source: ISO/FDIS 26262-6:2011

Software Unit Testing

Methods for deriving test cases for software unit testing

Methods		ASIL			
		A	B	C	D
1a	Analysis of requirements	++	++	++	++
1b	Generation and analysis of equivalence classes	+	++	++	++
1c	Analysis of boundary values	+	++	++	++
1d	Error guessing	+	+	+	+

Source: ISO/FDIS 26262-6:2011

Software Unit Testing

Structural coverage metrics at the software unit level

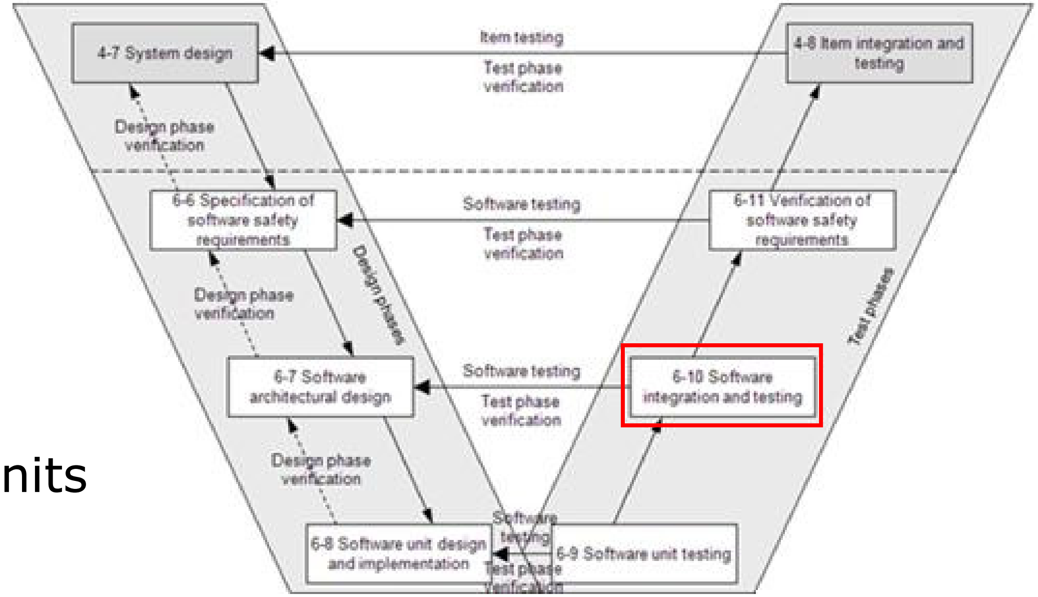
Methods		ASIL			
		A	B	C	D
1a	Statement coverage	++	++	+	+
1b	Branch coverage	+	++	++	++
1c	MC/DC (Modified Condition/Decision Coverage)	+	+	+	++

Source: ISO/FDIS 26262-6:2011

Software Integration and Testing

Goals

- Integrate SW components
  - Integration sequence
  - Testing of interfaces between components/units
- Verify correct implementation of the SW Architecture


4-7 System design
Item testing
Test phase verification
4-8 Item integration and testing
Design phase verification
6-6 Specification of software safety requirements
Software testing
Test phase verification
6-11 Verification of software safety requirements
Design phases
Test phases
Design phase verification
6-7 Software architectural design
Software testing
Test phase verification
6-10 Software integration and testing
Design phase verification
6-8 Software unit design and implementation
Software testing
Test phase verification
6-9 Software unit testing

Software Integration and Testing

The software integration test methods shall be applied to demonstrate that both the software components and the embedded software achieve:

- Compliance with the software architectural design
- Compliance with the specification of the hardware-software interface
- Correct implementation of the functionality
- Robustness and sufficiency of the resources to support the functionality

Methods		ASIL			
		A	B	C	D
1a	Requirements-based test	++	++	++	++
1b	Interface test	++	++	++	++
1c	Fault injection test	+	+	++	++
1d	Resource usage test	+	+	+	++
1e	Back-to-back comparison test between model and code, if applicable	+	+	++	++

Source: ISO/FDIS 26262-6:2011

Structural coverage metrics at the software architectural level

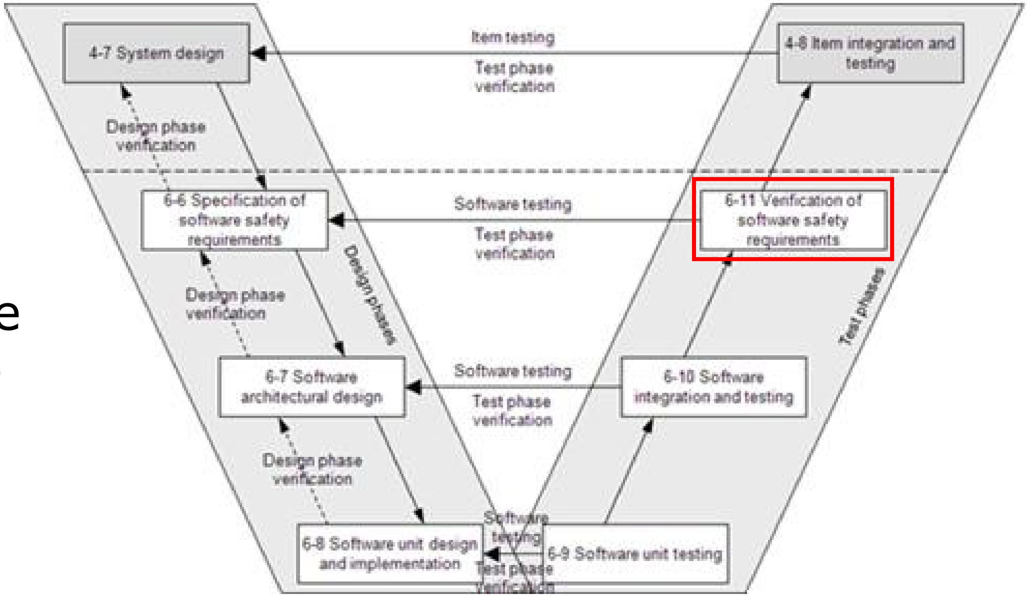
Methods		ASIL			
		A	B	C	D
1a	Function coverage	+	+	++	++
1b	Call coverage	+	+	++	++

Source: ISO/FDIS 26262-6:2011

Verification of Software Safety Requirements

Goals

- Verify that the embedded software fulfils the Software Safety Requirements in the target environment


4-7 System design
Item testing
Test phase verification
4-8 Item integration and testing
Design phase verification
6-6 Specification of software safety requirements
Software testing
Test phase verification
6-11 Verification of software safety requirements
Design phases
Test phases
Design phase verification
6-7 Software architectural design
Software testing
Test phase verification
6-10 Software integration and testing
Design phase verification
Software testing
6-8 Software unit design and implementation
Test phase verification
6-9 Software unit testing

Verification of Software Safety Requirements

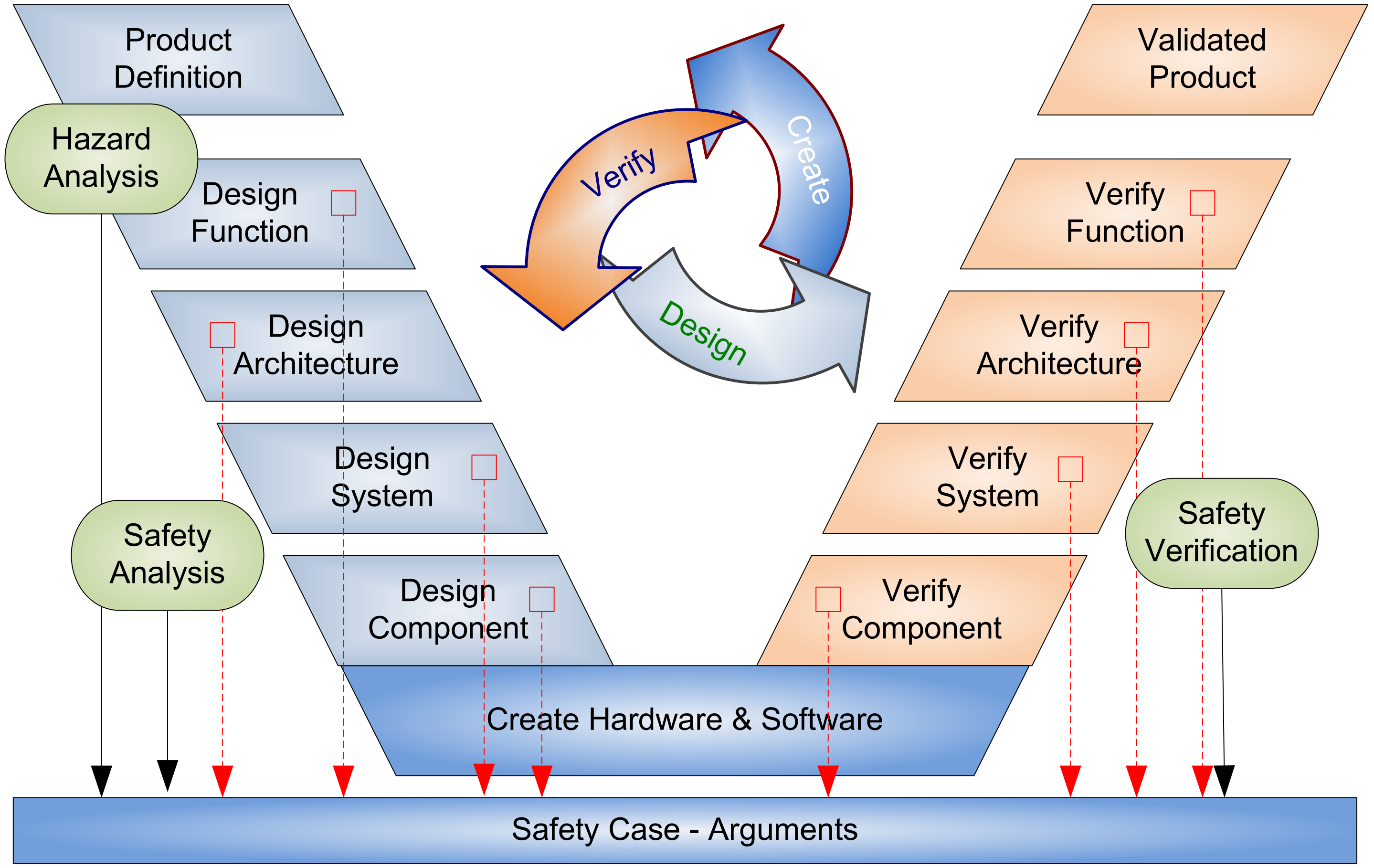
- Verify that the embedded software fulfils the software safety requirements
- Verification of the software safety requirements shall be executed on the target hardware
- The results of the verification of the software safety requirements shall be evaluated in accordance with:
  - Compliance with the expected results
  - Coverage of the software safety requirements
  - A pass or fail criteria

Methods		ASIL			
		A	B	C	D
1a	Hardware-in-the-loop	+	+	++	++
1b	Electronic control unit network environments	++	++	++	++
1c	Vehicles	++	++	++	++

Source: ISO/FDIS 26262-6:2011

Functional Safety Assessment

What shall be provided to support the Safety Case?



Contents

- Who is Method Park?
- Why do we need Safety Standards?
- Process and Safety demands in Automotive
- Hazard Analysis and Risk Assessment
- Functional and Technical Development
- Software Process in detail
- **Tool Qualification**
- Summary

Qualification of Software Tools

To determine the required level of confidence in a software tool, perform a use case analysis:

- Evaluate if a malfunctioning software tool and its erroneous output can lead to the violation of any safety requirement allocated to the safety-related item or element to be developed
- Establish probability of preventing or detecting such errors in its output
  - Considers measures internal to the software tool (e.g. monitoring)
  - Measures external to the software tool implemented in the development process for the safety-related item or element (e.g. guidelines, tests, reviews)

Qualification of Software Tools

Tool Impact (TI) – Possibility that a safety requirement, allocated to the safety-related item or element, is violated if the software tool is malfunctioning or producing erroneous output

TI1 – no such possibility

TI2 – all other cases

Tool error Detection (TD) – Probability of preventing or detecting that the software tool is malfunctioning or producing erroneous output

TD1 – high degree of confidence for prevention or detection

TD2 – medium degree …

TD3 – all other cases

Qualification of Software Tools

Tool Confidence Level (TCL) – Based on the values determined for the classes of TI and TD

	TD1	TD2	TD3
TI1	TCL1	TCL1	TCL1
TI2	TCL1	TCL2	TCL3

Source: ISO/FDIS 26262-8:2011

Qualification of Software Tools

methodpark

Qualification methods:

Qualification methods of software tools classified TCL3		ASIL			
		A	B	C	D
1a	Increased confidence from use	++	++	+	+
1b	Evaluation of the tool development process	++	++	+	+
1c	Validation of the software tool	+	+	++	++
1d	Development in accordance with a safety standard	+	+	++	++

Qualification methods of software tools classified TCL2		ASIL			
		A	B	C	D
1a	Increased confidence from use	++	++	++	+
1b	Evaluation of the tool development process	++	++	++	+
1c	Validation of the software tool	+	+	+	++
1d	Development in accordance with a safety standard	+	+	+	++

Source: ISO/FDIS 26262-8:2011

Contents

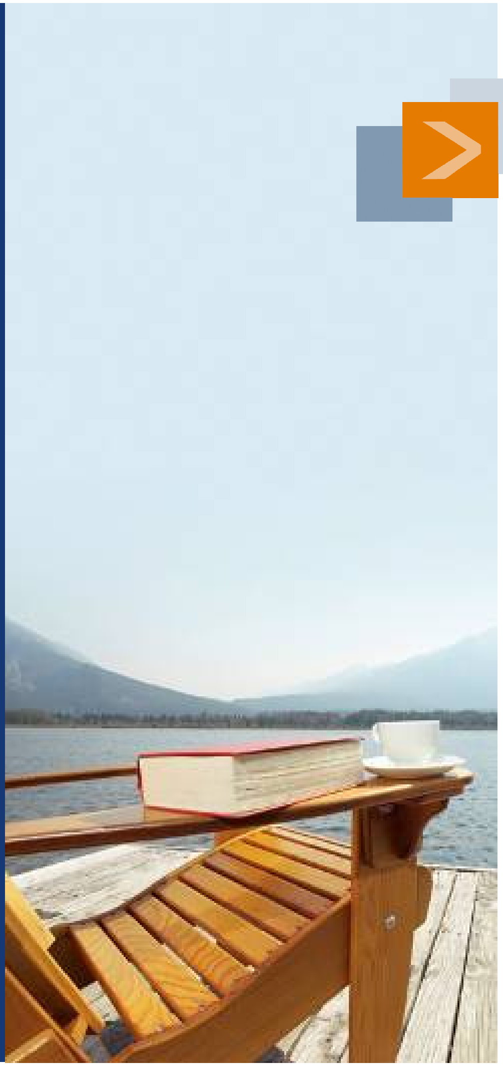
- Who is Method Park?
- Why do we need Safety Standards?
- Process and Safety demands in Automotive
- Hazard Analysis and Risk Assessment
- Functional and Technical Development
- Software Process in detail
- Tool Qualification
- **Summary**

Summary

- Today’s electronic systems are too complex to understand all potential hazards

- An approach for Functional Safety is needed to avoid severe injuries and damages in human lives and property

- A standardized way to show that your product is safe is needed – best practice yet not fully established – guidance needed



Thank you !

Bernhard Sechser
Principal Consultant SPICE & Safety

Method Park Software AG
Wetterkreuz 19a
91058 Erlangen
Germany

Phone: +49 9131 97206-427
Mobile: +49 173 3882055

Bernhard.Sechser@methodpark.com
http://www.xing.com/profile/Bernhard_Sechser
http://www.methodpark.com