Introduction to ISO 26262

Even-André Karlsson

addalot

QUALITY IMPROVEMENT

Introduction

- Even-André Karlsson– 30 years of Process and Quality improvement
  - Model based Improvement CMMI, A-SPICE, COBIT, ISDS
  - System engineering, Architecture, Tools, Requirements engineering
  - Agile, Lean, Team based organisation and Coaching
  - Automotive, Mechanical, Mobile, Telecom
- Company changes but with continued focus and services:
  - Process improvement
  - Software Quality
  - Software Safety
  - Supplier Management
  - Open Source Software

- SPICE/CMMI references
  - Accel, Atlas Copco, Autoliv, BorgWarner, Consat, GM, Mecel, Stoneridge, Volvo
  - ABB, Ericsson, FMC, IKEA, Kongsberg, QLIK, SAAB, Thales, Visma

Introduction – participants

- Name, role/background
- Experience in ISO 26262
- Expectations for the day

Agenda

- 1300-1315 Introduction
- 1315-1400 Functional Safety Background
- 1400-1530 ISO 26262
- 1530-1600 ISO 26262 and Automotive SPICE
- 1600-1630 Implementing ISO 26262

addalot QUALITY IMPROVEMENT

Principles

- Focus
  - Respect times
  - Email/phone
  - Active
- Communication
  - Listen
  - Respect
  - Seek understanding
- Parking lot

addalot QUALITY IMPROVEMENT

Functional Safety Background

Consequences of un-safe software

Unintended acceleration

Experts determined after **18 months review** that the software was ”**substandard**” and that Toyota had not followed ”**best practice**”

Toyota has paid so far

- 1 Billion for to deceased
- 1 Billion to US authorities for concealing information
- 1 Billion for reduced second hand value

addalot
QUALITY IMPROVEMENT

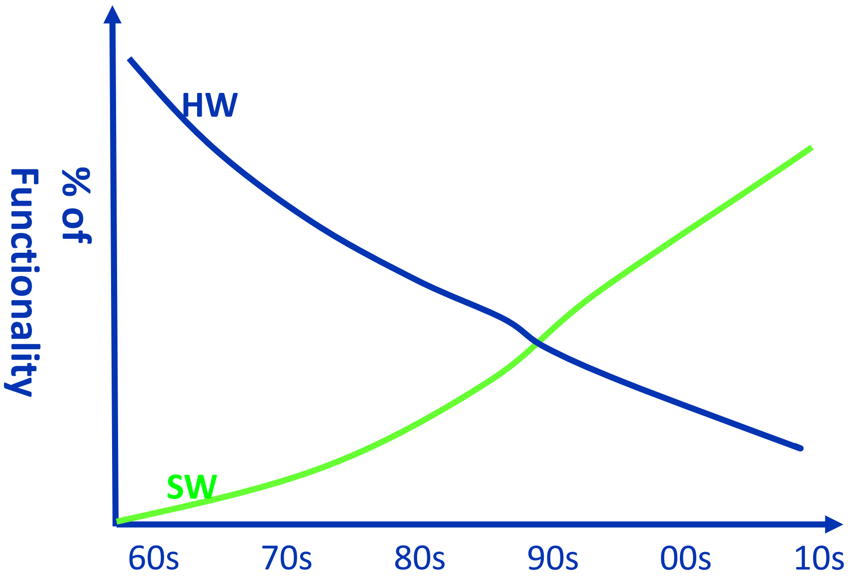
Recalls

- 2004, Jaguar recalls 67,798 cars for transmission fix. A Software defect slams the car into reverse gear if there is a major oil pressure drop.
- 2015, Nissan recalls 23.00 ”Micras” due to a software defect that causes the car to suddenly accelerate unintentionally
- 2016, GM recalls 4.3 million cars for airbag software defect. The bug, affecting all pickups and SUVs, can prevent the airbags from deploying in a crash
- 2016, Volvo recalls 59.000 cars due to a software bug after some owners experienced that their engines stopping and restarting while they were driving

Ref: Software Integrity

Automotive and the Standards

- 1985 ISO 9000 ➔ TS 16949
  - Product development
  - Product and Process Focus
- 1995 CMMI/SPICE ➔ Automotive SPICE
  - Software development
  - Software and Process Focus
- 2005 IEC 61508 ➔ ISO 26262
  - Safety critical development
  - Software, Hardware and Process Focus
- 201X – SECURITY ??



Safety

Software dependant systems is safe when:

- features ensure **predictable performance** under normal/abnormal condotions
- the **probability** of an undersirable event occuring is minimized
- an undesirable event does occur, the **consequences** are controlled

absence of unreasonable risk

D. Herrman "Software Safety and Reliability"

A problem has been detected and Windows has been shut down to prevent da
your computer.

MEMORY_MANAGEMENT

If this is the first time you've seen this Stop error screen, restart yo
computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. I
is a new installation, ask your hardware or software manufacturer for an
Windows updates you might need.

If problems continue, disable or remove any newly installed hardware or
software. Disable BIOS memory options such as caching or shadowing. If y
to use Safe Mode to remove or disable components, restart your computer,
F8 to select Advanced Startup Options, and then select Safe Mode.

Technical Information:

*** STOP: 0x0000001A

Beginning dump of physical memory
Physical memory dump complete.

Contact your system administrator or technical support group for further
assistance.

Blue screens…

A problem has been detected and Windows has been shut down to prevent damage to your computer.

MEMORY_MANAGEMENT

If this is the first time you've seen this Stop error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

Technical Information:

*** STOP: 0x0000001A

Beginning dump of physical memory
Physical memory dump complete.

Contact your system administrator or technical support group for further assistance.

… are annoying in windows computers…

…but could be **safety concerns** in embedded systems!

Safe systems

- How safe do our systems need to be?
- Safety requirements change over time

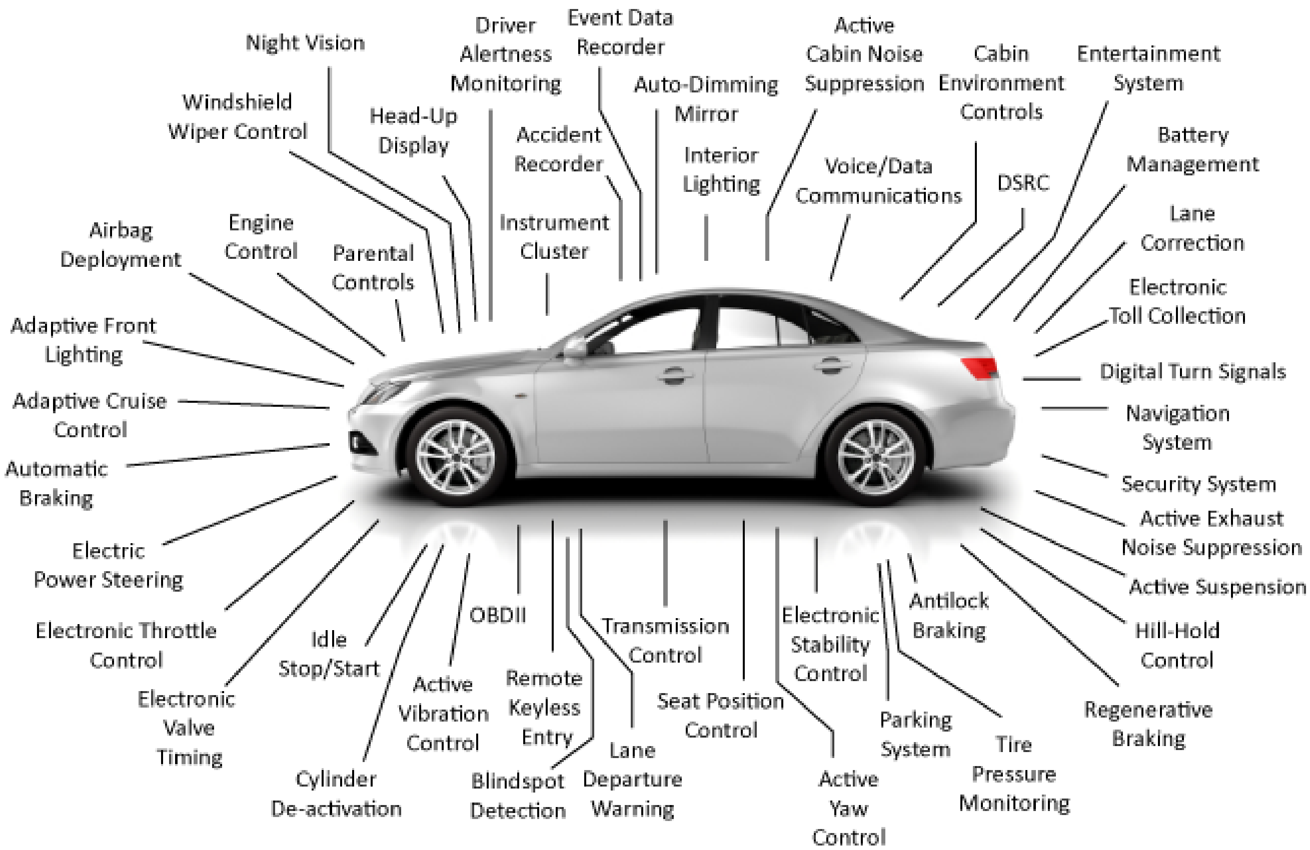
Safe system vs Safe usage....

addalot QUALITY IMPROVEMENT

Safety related automotive functionality

- Active systems
- Passive systems
- Information systems
- E/E enhanced mechanical systems
- Light control
- Powertrain
- Autonomous drive

Software based systems in cars



Safety related failure modes

- Absence of function when needed
  - Acceleration, break, turn, etc

- Unintended function
  - acceleration, break, turn, engine stop, air bag, etc.
  - or…sudden power seat movement

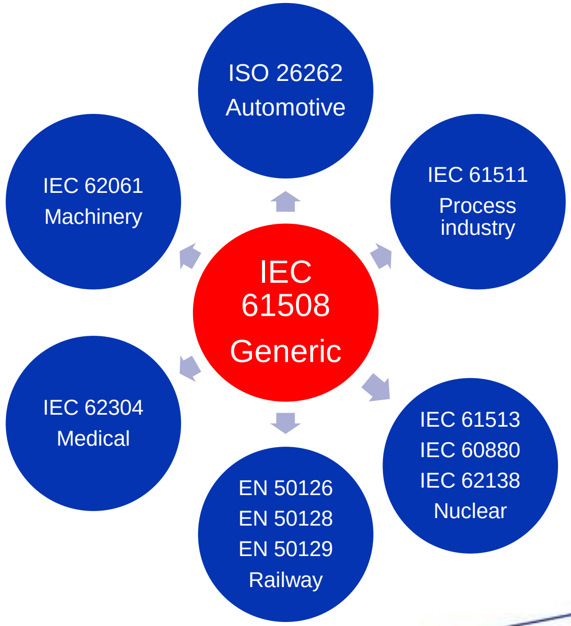
- Safety of the intended functionality (SOTIF)
  - Artificial intelligence (AI) and machine learning play key roles in the development of autonomous vehicles ➔ increase complexity!
  - New standard

addalot QUALITY IMPROVEMENT

History of Functional Safety Standards

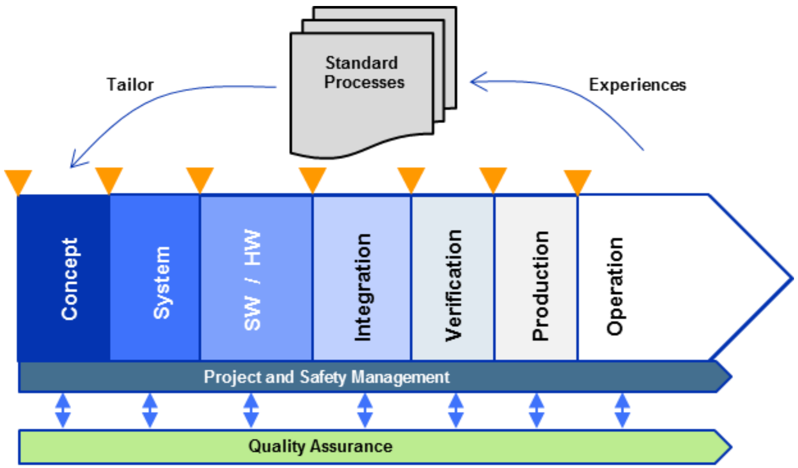
- The principles underpinning Functional Safety were developed in the military, nuclear and aerospace industries during the 1960-1970 ties
- **1995 IEC 1508**
  - New approach to functional safety – Risk based
  - Define safety requirements to reduce risk
- **1998-2000 IEC 61508**
  - New approach to functional safety – Risk based
  - Define safety requirements to reduce risk
- **IEC 61508 detailed in**

- Medical	IEC 62304
- Machinery	IEC 62061
- Railway	EN 5012X
- Nuclear Process	IEC 61513 …
- Automotive	IEC 26262



How do we “prove” that a system is safe?

- Follow standards with requirements & guidelines for safe systems
  - Exhaustive testing not possible
- Typically, the standards require
  - defined process that cover the whole life cycle



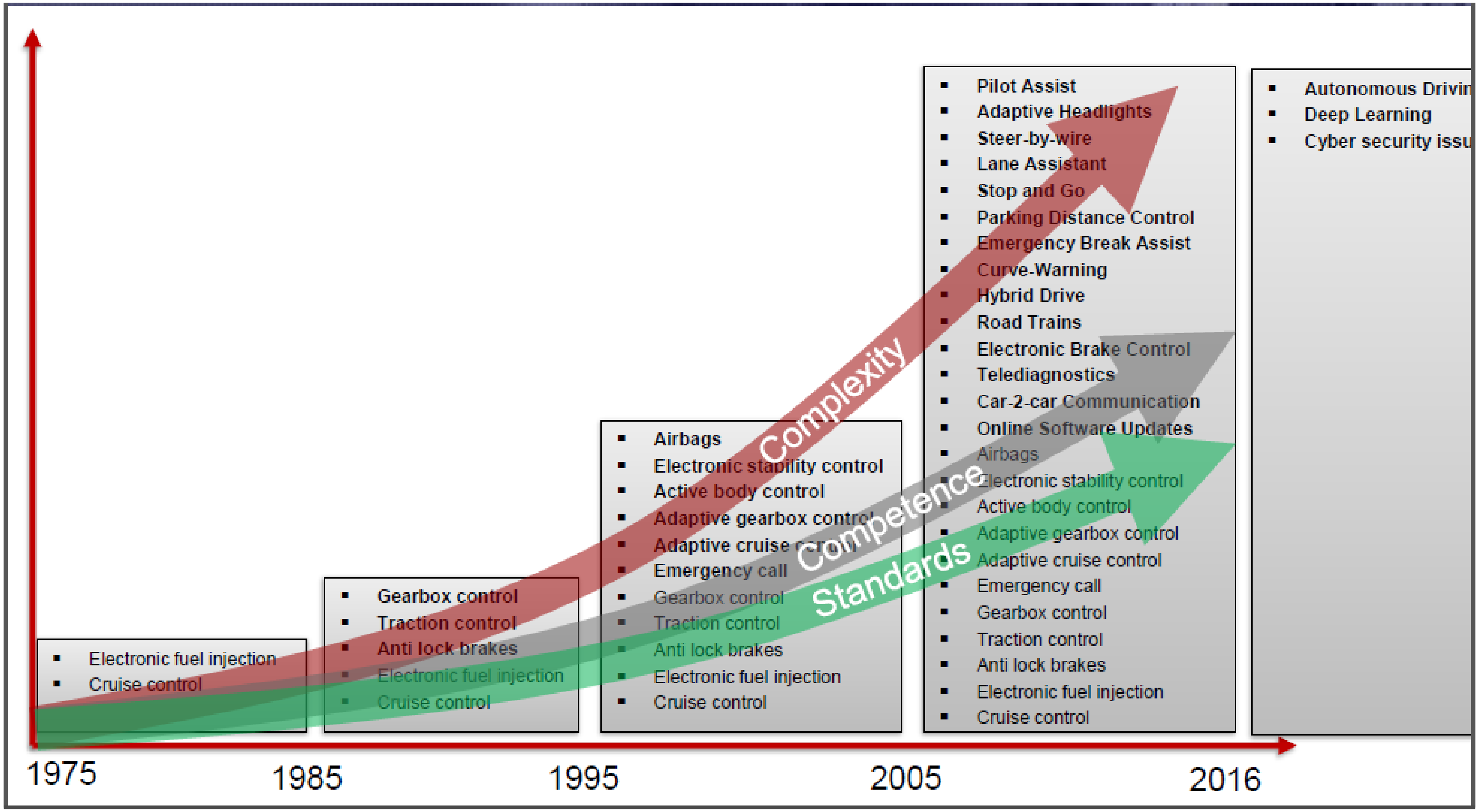
  - activities to ensure that the defined way of working is followed and complies with the standard
  - evidence of the safety related activities

addalot QUALITY IMPROVEMENT

Liability

- A manufacturer has to organize the company to ensure that design, development and documentation faults are eliminated or detected
- The manufacturer has to prove that it is not responsible for a fault
  ➔ By using state of the art science and technology
- "State of the art" in automotive
  - IATF 16949
  - Automotive SPICE
  - ISO 26262
- If the malfunction could not have been detected by the technical state of the art, the liability is excluded.

Standards are always behind…



Ref: Software Integrity

ISO 26262

addalot
QUALITY IMPROVEMENT

What is ISO 26262?

- A functional safety standard for E/E systems in road vehicles
- Addresses hazards caused by malfunctioning behavior of E/E systems
- Provides requirements on organization, processes, and methods
- Covers the product lifecycle from concept phase to decommissioning
- First edition was published November 2011.
- Second (and current) edition was published 2018
  - Inclusion of **all road vehicles**: busses, trucks and motorcycles
  - Safety of the Intended Functionality (SOTIF)
  - Cyber Security, Model Based Development and Agile SW developement
  - Development of random hardware failure metrics
  - Ensure confidence in the use of software tools to include vendor validation
  - Semiconductors guide

Basic principles

- Perform risk analysis
- Define safety goals/requirements to reduce identified risks
- Avoid systematic failures by following defined processes and using recommended methods
- Control systematic and random hardware failures during operation
- Manage the safety activities (plan, follow-up, etc.)
- Evidence of the safety related activities - a safety case
- Traceability
- Perform functional safety assessment to judge the functional safety achieved

Management of functional safety

Part 1: Vocabulary

Part 2: Management of functional safety

Part 3: Concept phase

Part 4: Product development at the system level

Part 7: Production & operation

Part 12: Adaptation of ISO 26262 for motorcycles

Part 5: Product development at the hardware level

Part 6: Product development at the software level

Part 8: Supporting processes

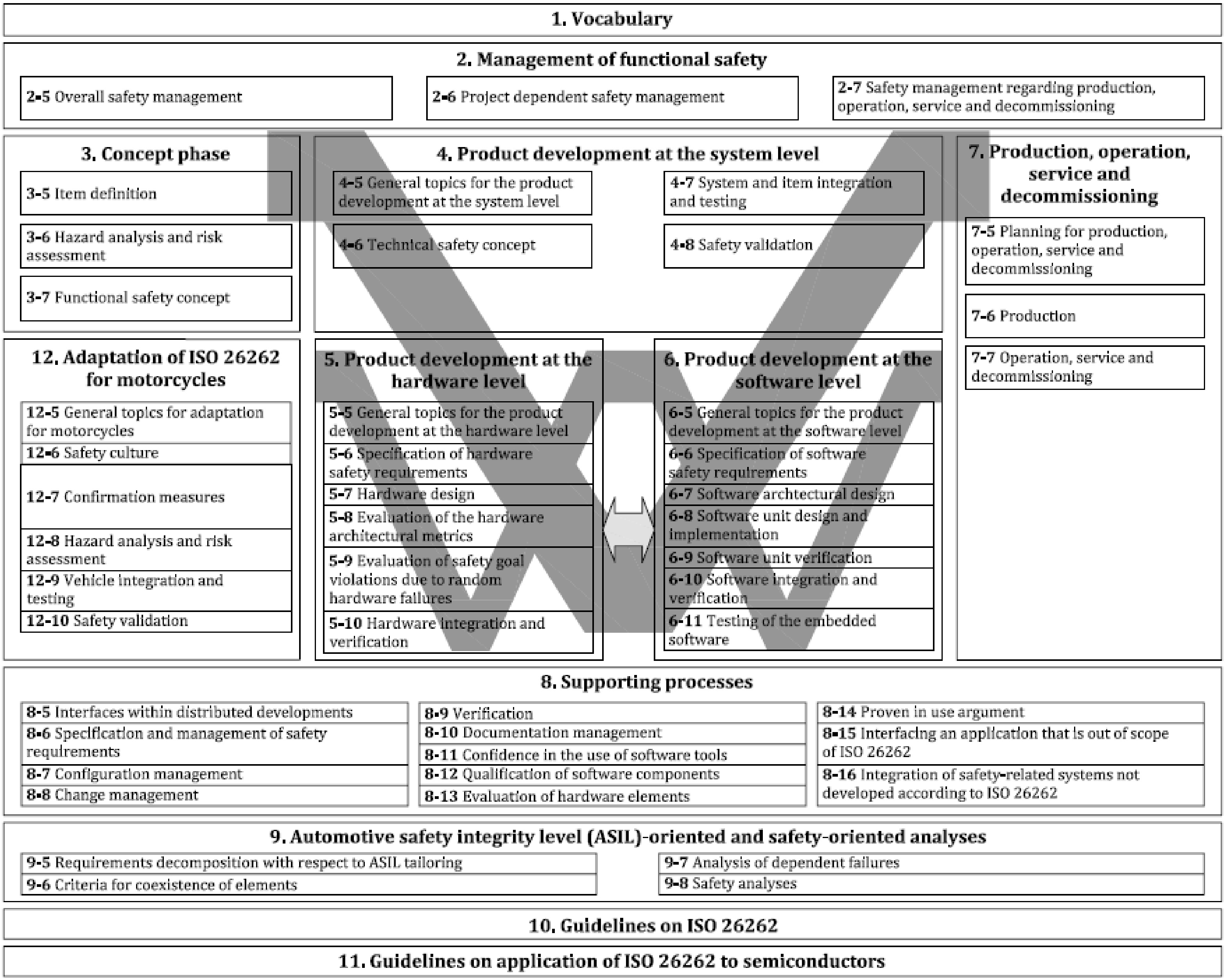
Part 9: ASIL-oriented and safety-oriented analyses

Part 10: Guideline on ISO 26262 (informative)

Part 11: Guidelines on application of ISO 26262 to semiconductors

addalot QUALITY IMPROVEMENT

ISO 26262 overview – clauses and lifecycle



Requirements for compliance

- Each clause contains requirements and recommendations
- Each requirement shall be complied with unless:
  a) tailoring shows that the requirement does not apply, or
  b) rationale for non-compliance has been assessed and accepted

- Method tables
  - ++ highly recommended
  - + recommended
  - o no recommendation (for or against)
  - ASIL dependent
  - Use "appropriate combination" for alternative entries and give rationale for selection

Table 1 — Topics to be covered by modelling and coding guidelines

	Topics	ASIL			
		A	B	C	D
1a	Enforcement of low complexity[a]	++	++	++	++
1b	Use of language subsets[b]	++	++	++	++
1c	Enforcement of strong typing[c]	++	++	++	++
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	++	++	++	++

[a] An appropriate compromise of this topic with other methods in this part of ISO 26262 may be required.

[b] The objectives of method 1b are

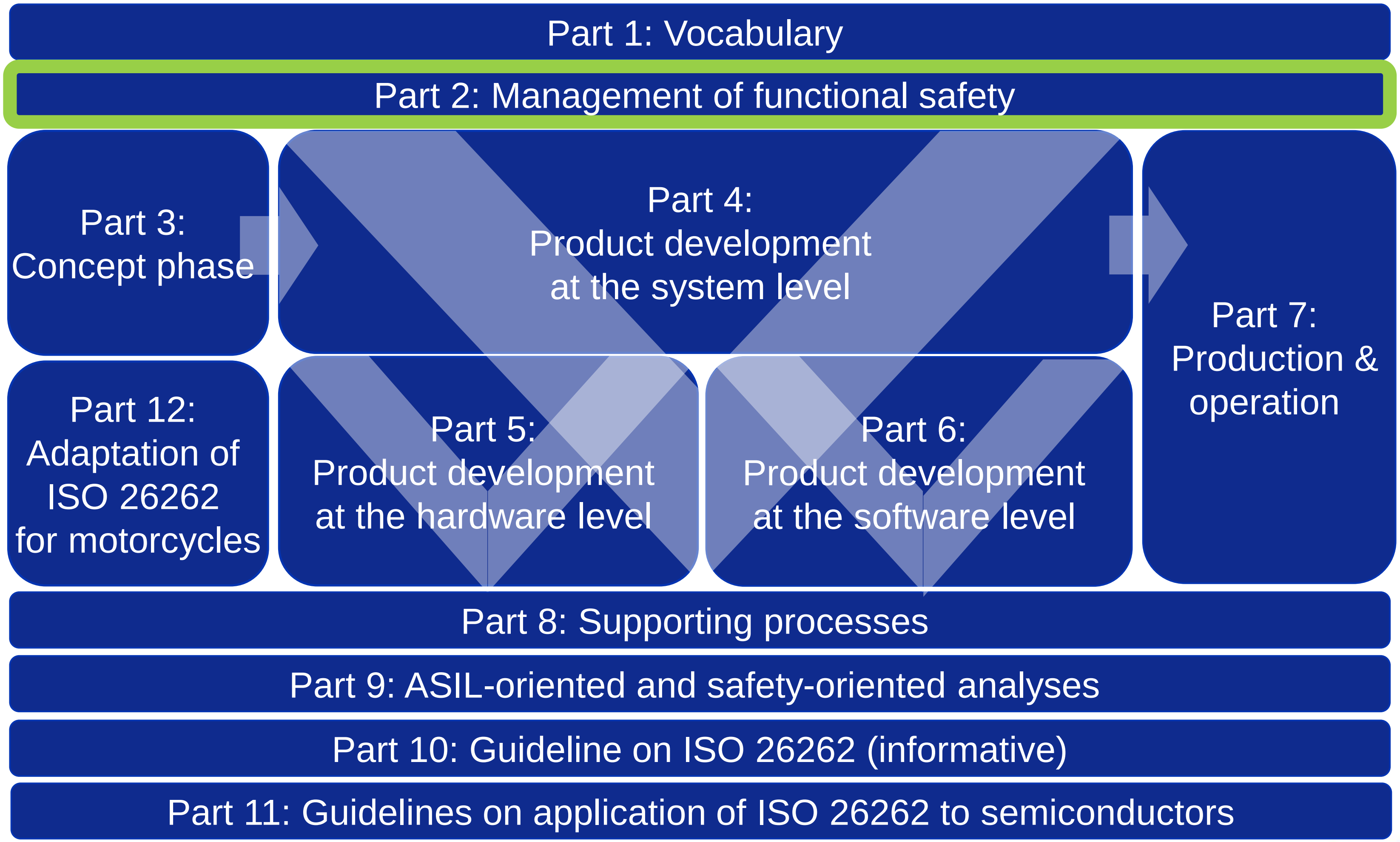
— Exclusion of ambiguously defined language constructs which may be interpreted differently by different modellers, programmers, code generators or compilers.

— Exclusion of language constructs which from experience easily lead to mistakes, for example assignments in conditions or identical naming of local and global variables.

— Exclusion of language constructs which could result in unhandled run-time errors.

[c] The objective of method 1c is to impose principles of strong typing where these are not inherent in the language.

Management of functional safety



addalot QUALITY IMPROVEMENT

Management of functional safety

Overall safety management

- Allocate safety responsibilities
- Create safety culture
- Training and qualification
- Quality management system

Project dependent safety management

- Appoint roles (PM & Safety Manager)
- Tailor safety activities
- Establish and follow up safety plan
- Develop safety case
- Confirmation measures

Safety management regarding production, operation, service and decommissioning

- Appoint roles
- Establish processes, e.g. field monitoring

addalot
QUALITY IMPROVEMENT

Safety plan and safety case

- Safety plan
  - Plan to manage and guide the execution of the safety activities of a project including dates, milestones, tasks, deliverables, responsibilities and resources
  - Created and followed up by the (project) safety manager

- Safety case
  - Arguments that the safety requirements for an item are complete and satisfied by evidence compiled from work products of the safety activities during development
  - Input to functional safety assessment

Confirmation measures

- **Confirmation review**
  Checks the compliance of work products to the ISO 26262 requirements
- **Functional safety audit**
  Evaluates the implementation of the processes required for functional safety
- **Functional safety assessment**
  Evaluates the functional safety achieved by the item. Shall consider:
  - work products in safety plan
  - processes required for safety
  - appropriateness and effectiveness of the implemented safety measures

ISO 26262-2, Table 1

I0..I3 What does it mean?

Confirmation measures		Degree of independency				
Type	Target	QM	ASIL A	ASIL B	ASIL C	ASIL D
Review	**Impact analysis**	**I3**	**I3**	**I3**	**I3**	**I3**
Review	Hazard analysis and risk assessment	**I3**	I3	I3	I3	I3
Review	Safety plan (proven in use arguments)	—	**I1**	I1	I2	I3
Review	**Functional safety concept**	—	**I1**	**I1**	**I2**	**I3**
Review	**Technical safety concept**	—	**I1**	**I1**	**I2**	**I3**
Review	Integration and test **strategy**	—	I0	I1	I2	I2
Review	**Safety validation specification**	—	**I0**	**I1**	**I2**	**I2**
Review	Safety analyses and the dependent failure analysis	—	I1	I1	I2	I3
Review	Completeness of the safety case	—	**I1**	I1	I2	I3
Audit	Functional safety audit	—	—	I0	I2	I3
Assessment	Functional safety assessment	—	—	**I0**	I2	I3

Ref: TüV training

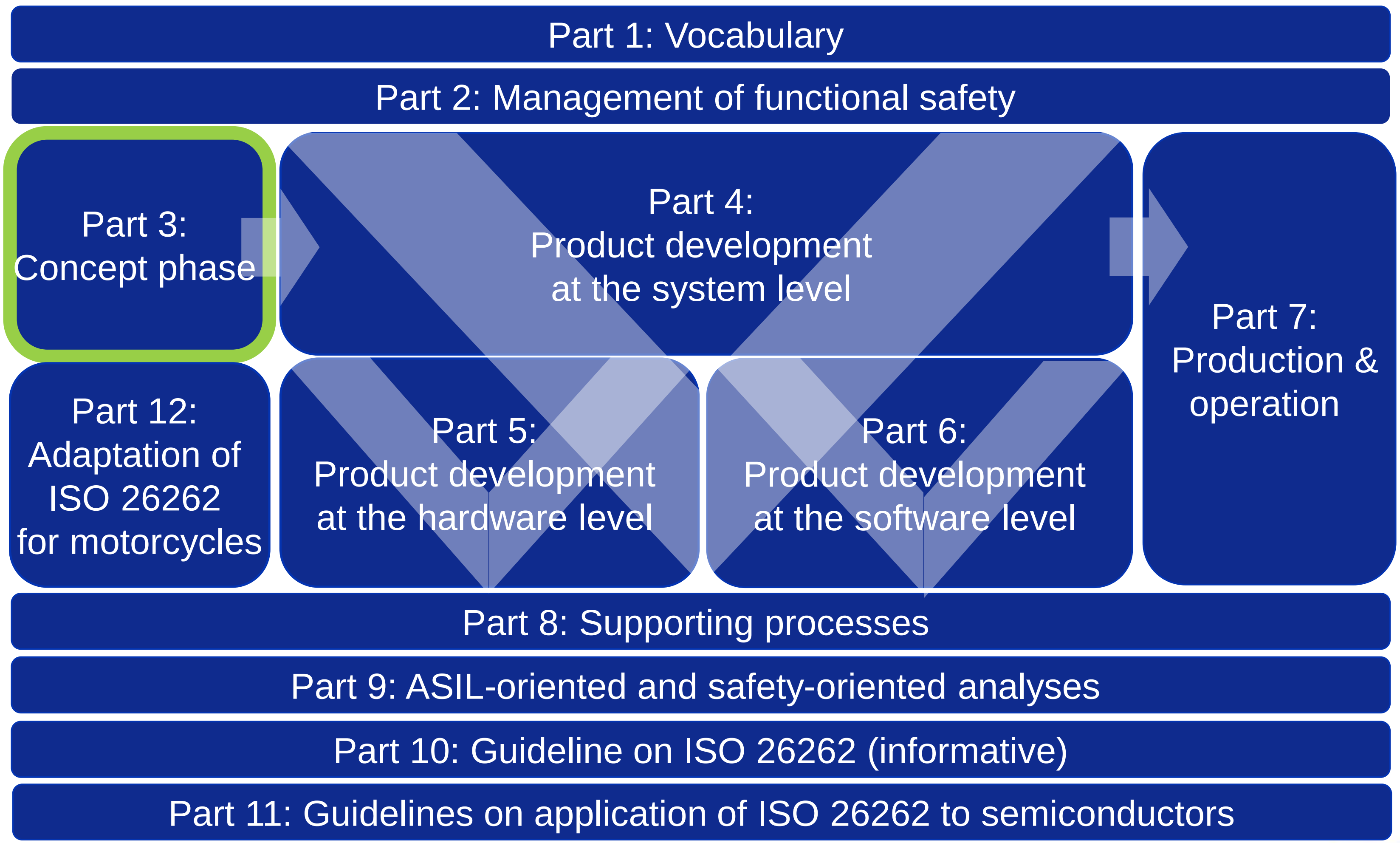
Examples for evaluating a safety culture

Accountability not traceable	Accountability traceable
Cost/time highest priority	Safety highest priority
Reward system favors cost/time	Reward system favors safety
Dependent assessor	Independent assessor
Passive attitude (problem driven)	Proactive attitude
Resources not planned	Planned resources
“Group think”	Diversity encouraged
No defined processes	Defined processes are followed
No process improvement	Continuous process improvement

Poor safety culture ⟷ Good safety culture

addalot QUALITY IMPROVEMENT

Concept phase



Concept phase

Item definition

↓

Hazard analysis and risk assessment

↓

Functional safety concept

Item definition

- Functional and non-functional requirements
  - Operating modes and states
  - Operational and environmental constraints
  - Legal requirements and standards
  - Assumptions
  - Potential consequences of failures
- Item boundaries and interaction with other items or elements
- Determine if it is new development or modification of an existing item
- Impact Analysis

Hazard analysis and risk assessment

1. Situation analysis and hazard identification
2. Classification of hazardous events

Severity:

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

Probability of exposure:

E0	E1	E2	E3	E4
Incredible	Very low probability	Low probability	Medium probability	High probability

Controllability:

C0	C1	C2	C3
Generally controllable	Simply controllable	Normally controllable	Difficult to control or uncontrollable

3. Determination of ASIL and safety goals

Safety Goals and ASIL

		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

Example 1: Airbag does not deploy during crash

- Severe injuries -> S3
- Very low exposure -> E1
- Not controllable -> C3

➡ ASIL A

- Safety goal: airbag shall deploy during crash

Example 2: Unwanted airbag deployment

- Severe injuries -> S3
- High exposure -> E4
- Difficult to control -> C3

➡ ASIL D

- Safety goal: No unwanted airbag deployment

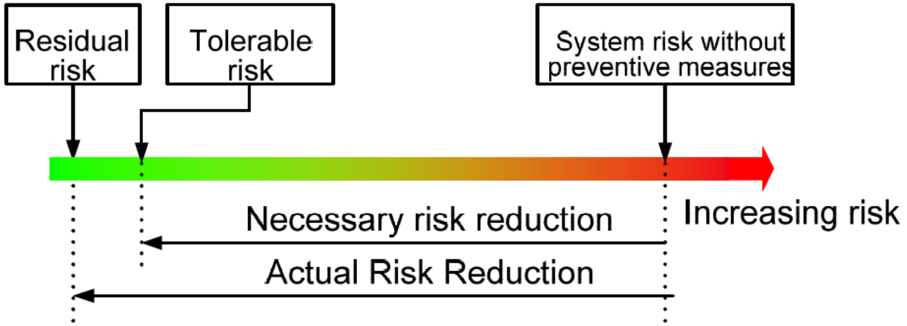
addalot QUALITY IMPROVEMENT

ASIL – Automotive Safety Integrity Level

- Represent how dangerous a hazardous event is
- Determines the required degree of safety measures to avoid unreasonable risk (which requirements in ISO 26262 that shall be applied)

ASIL = Severity x Exposure x Controllability

- ASIL D is the most stringent level and ASIL A the least stringent level
- The ASIL is an attribute of a safety requirement



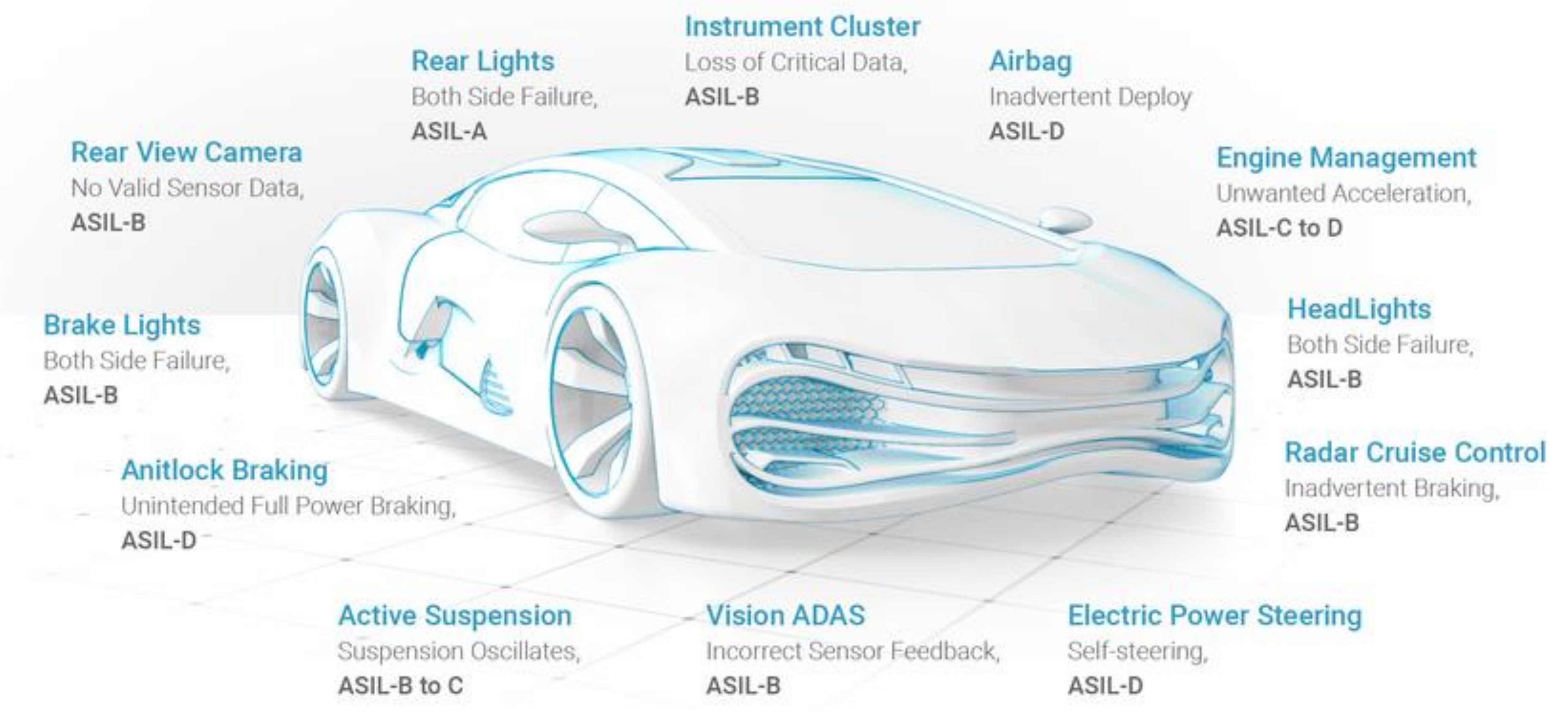
Ref: TüV training

Hazard analysis is context dependent

Adapted HAZOP												
ID	Function	Parameter	Guide-word	Deviation	Hazardous event	Operational situation	Conseq	S	E	C	ASIL	Causes
H1	Fuel level estimation (AE201)	Total fuel level	Supplied too high	Total fuel level supplied too high	Fuel gauge Indicates higher fuel level than actual fuel level in the tank during driving	Free way	Vehicle is driven until no more fuel could be collected from the tank. Resulting in engine stop suddenly. Thus crush by other cars coming from behind is expected.	3	2	2	A	1) Erroneous fuel estimation by Kalman filter 2) Bug in gauge function 3) Mechanical Fault in gauge
						High way- heavy traffic		3	4	3	D	
						City driving, slippery road- high traffic		2	3	2	A	
						City driving- snow and ice- driving speed 50 km/h		3	2	3	B	

Ref: R. Dardar, "Building a Safety Case in Compliance with ISO 26262," Master Thesis, Mälardalen University, 2013.

Typical Automotive Classifications



Ref: Synopsys, Mentor

Impact of an ASIL?

- For all ASILs: Safety mechanisms to detect and handle the relevant failure modes at system level shall be introduced.

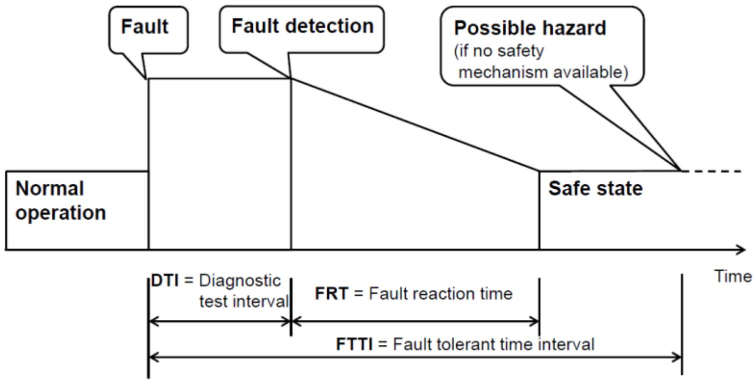
- **For ASIL A and ASIL B**
  - Emphasis on additional development activities and for quality assurance of introduced safety mechanisms. (e.g. Reviews and V&V activities)

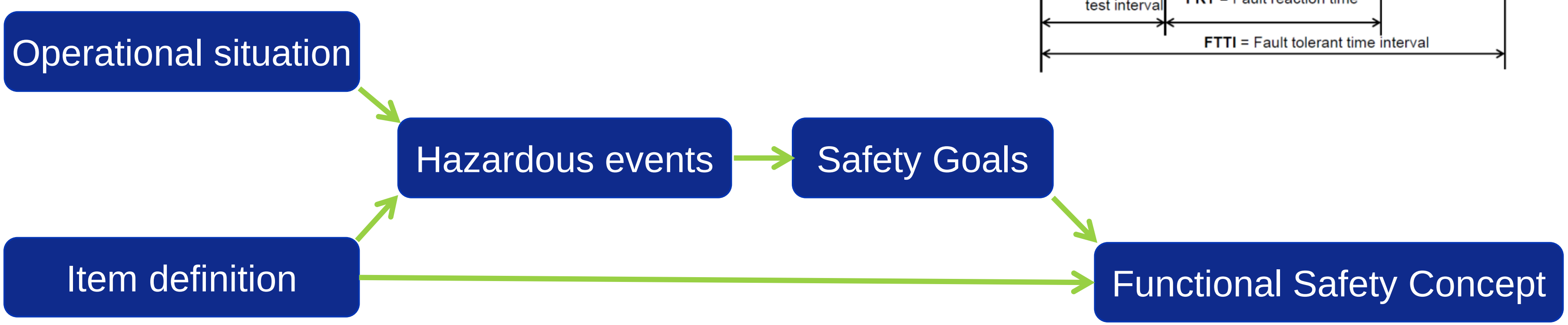
- **For ASIL C and ASIL D**
  - Further emphasis on additional development activities and for quality assurance of introduced safety mechanisms.
  - Requirements on performance of safety mechanisms. (Typically require HW redundancy)
  - Independent confirmation measurements

addalot QUALITY IMPROVEMENT

Functional safety concept

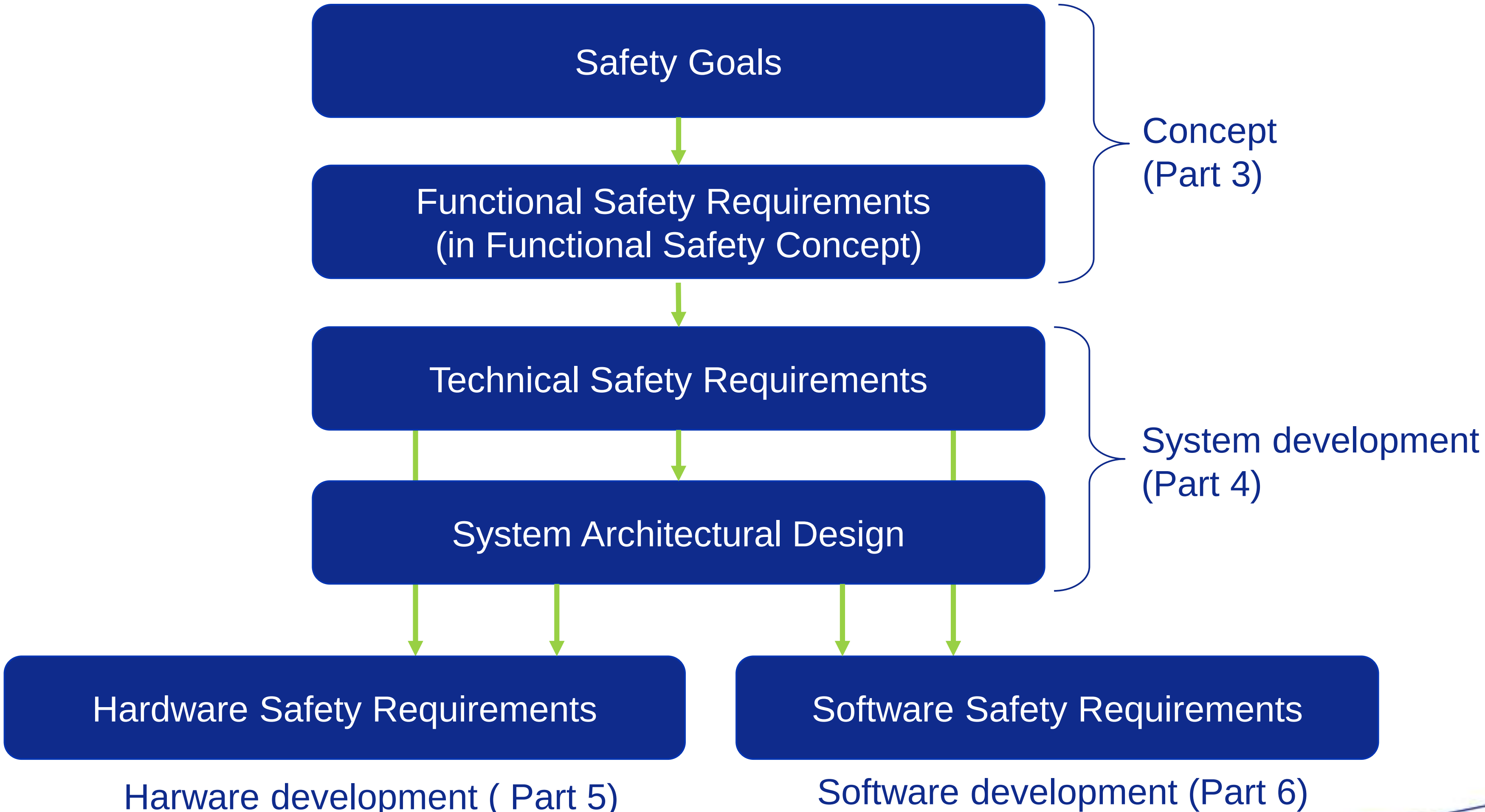
- Functional safety requirement derived from the safety goals
- Functional safety requirements allocated to system architecture
- Input to the product development phase





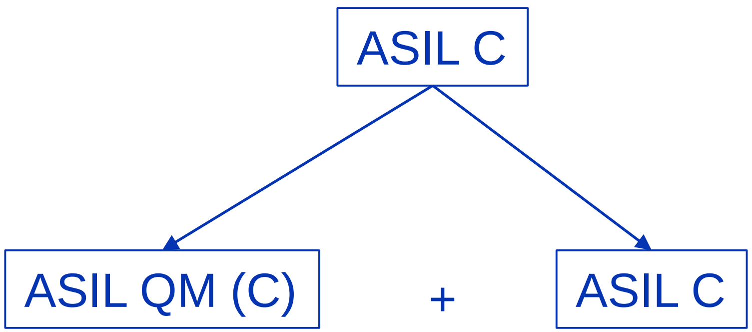
addalot QUALITY IMPROVEMENT

Safety requirements hierarchy



ASIL Decomposition

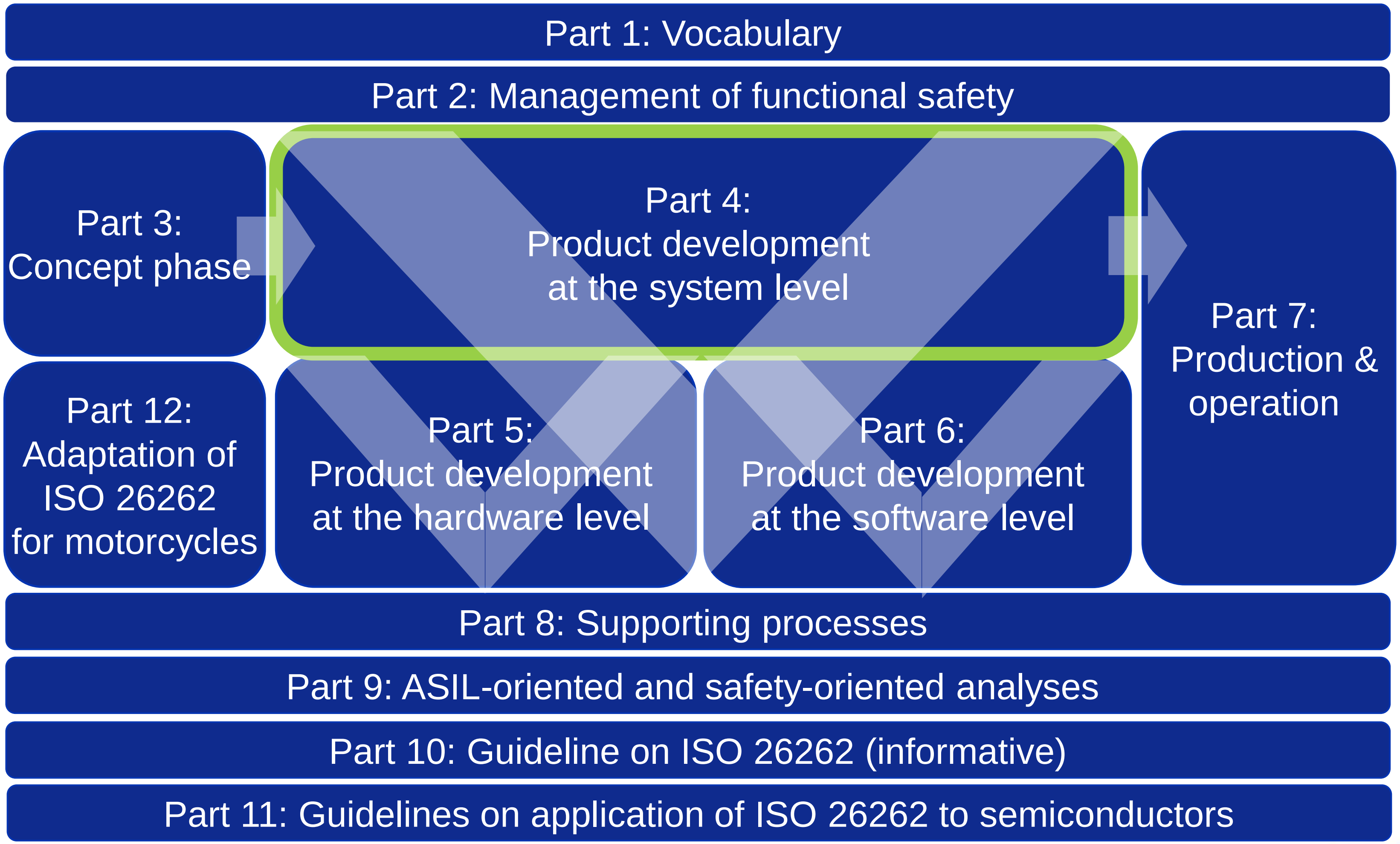
- Divide the architecture into **redundant** and **independent** parts



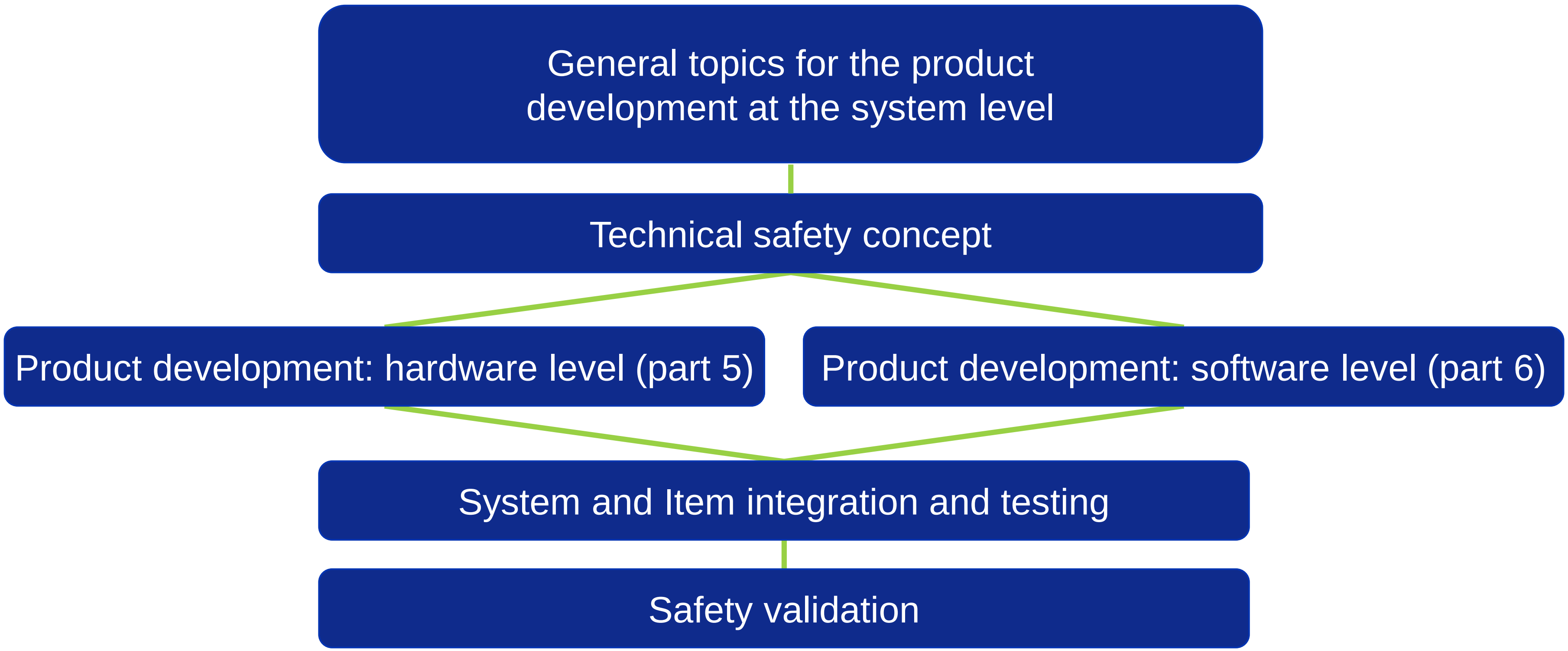
Coupling factor class	ISO 26262-9, 7.4.4	Examples			
		system level	hardware level	software level	semiconductor level
Shared Resource The same software, hardware, or system element instance is used by two elements, which are therefore affected by the failure or unavailability of that shared resource.	a) random hardware failures g) failures of common external resources	— Power supply (see also Insufficient Environmental Immunity) — Wiring harness — Data and communication busses — Powerstage	— Clock — Same H-Bridge used by two shut-down paths — Sockets, plug connectors	— SW component used by 2 other SW components — maths or other libraries — I/O routines, drivers — Hardware resource used by more than one software element	“Failure of shared resources” and “single physical root cause“ in ISO 26262-11

- Can be applied on all levels, and repeatedly
- But we need to ensure no common failures

Management of functional safety



Product development: System level



Technical Safety Concept

- The **technical safety concept** is an aggregation of the technical safety requirements and the corresponding system architectural design that provides rationale as to why the system architectural design is suitable to fulfil safety requirements

Requirements

- Specification of the technical safety requirements
- Safety mechanisms (detection, indication and control of faults)
- System architectural design specification
- Safety Analyses and avoidance of systematic failures
- Measures for control of random hardware failures
- Allocation to hardware and software
- Hardware-software interface (HSI) specification
- Verification methods

System and item integration and testing

- The integration of the item's elements is carried out in a systematic way starting from software-hardware integration and verification through system integration and verification to vehicle integration

Requirements

- Specification of integration and test strategy
- Hardware-software integration and testing
- System integration and testing

Test methods, examples:

- Requirement based tests
- Fault injections tests
- Resource usage test
- Stress test

Safety Validation

- The purpose of safety validation is to provide evidence that the **safety goals** are achieved and that the **safety concepts** (FSC TSC) are appropriate

Requirements

- Safety validation environment
- Specification of safety validation
- Execution of safety validation
- Evaluation

Methods to be used for validation

- Analysis (e.g. FMEA, FTA, simulation)
- Long term tests
- User test
- Reviews

addalot QUALITY IMPROVEMENT

Management of functional safety

Part 1: Vocabulary

Part 2: Management of functional safety

Part 3: Concept phase

Part 4: Product development at the system level

Part 7: Production & operation

Part 12: Adaptation of ISO 26262 for motorcycles

Part 5: Product development at the hardware level

Part 6: Product development at the software level

Part 8: Supporting processes

Part 9: ASIL-oriented and safety-oriented analyses

Part 10: Guideline on ISO 26262 (informative)

Part 11: Guidelines on application of ISO 26262 to semiconductors

Product development: hardware level

General topics for the product development at the hardware level

Specification of hardware safety requirements

Hardware design

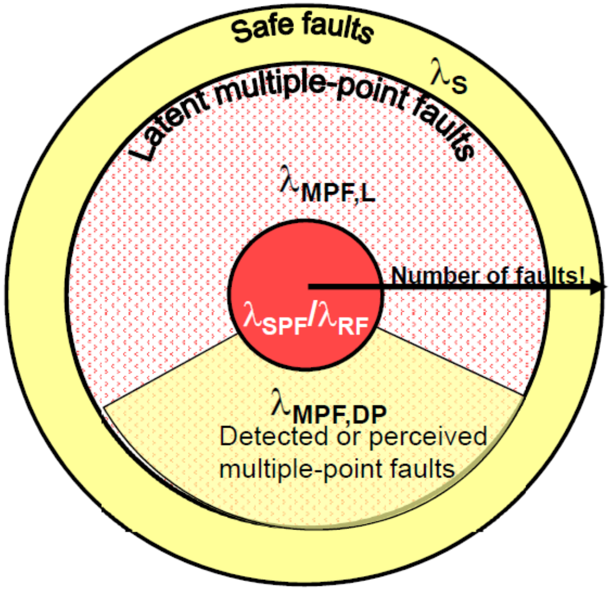
- Evaluation of the hardware architectural metrics
- Evaluation of safety goal violation due to random hardware failure

Hardware integration and testing

Hardware architecture metrics

Single-point fault metrics $= 1 - \dfrac{\sum(\lambda_{SPF} + \lambda_{RF})}{\sum \lambda}$

Latent-fault metrics $= 1 - \dfrac{\sum \lambda_{MPF\_Latent}}{\sum(\lambda - \lambda_{SPF} - \lambda_{RF})}$



λ: Fault frequency
SPF: Singe-Point Fault
MPF: Multiple-Point Fault
RF: Residual Fault

addalot QUALITY IMPROVEMENT

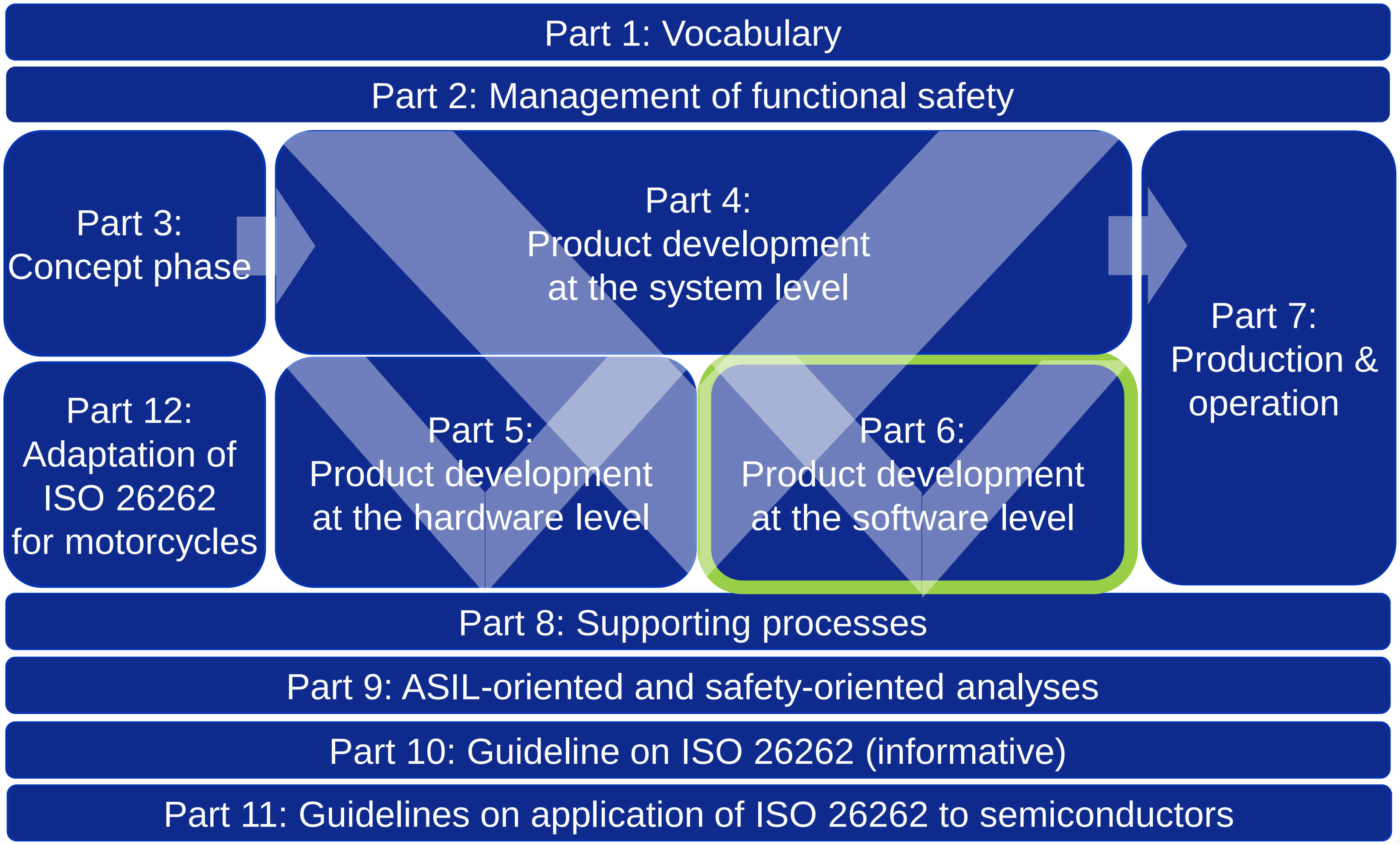
Random HW failure goals

	Random hardware failure target values for violation of safety goal	
ASIL	**Failure rate / h^{-1}**	**FIT / 10^{-9} h^{-1}**
D	$< 1 \times 10^{-8}$	< 10
C	$< 1 \times 10^{-7}$	< 100
B	$< 1 \times 10^{-7}$	< 100

What does this mean?

- 10 FIT = 10 Error per 10^9 hours = 10 Errors per 114,155 years
- But with 2.000.000 cars on the road, it means that 175 cars will experience this fault every year…
- Now it is not so bad, as the cars don't run 24/7, but assume they run an hour a day, we still have 7 exploding airbags every year…
- Calculations are mainly to show that you have done an analysis.

Management of functional safety



Product development at the software level

General topics for the product development at the hardware level

Specification of the software safety requirements

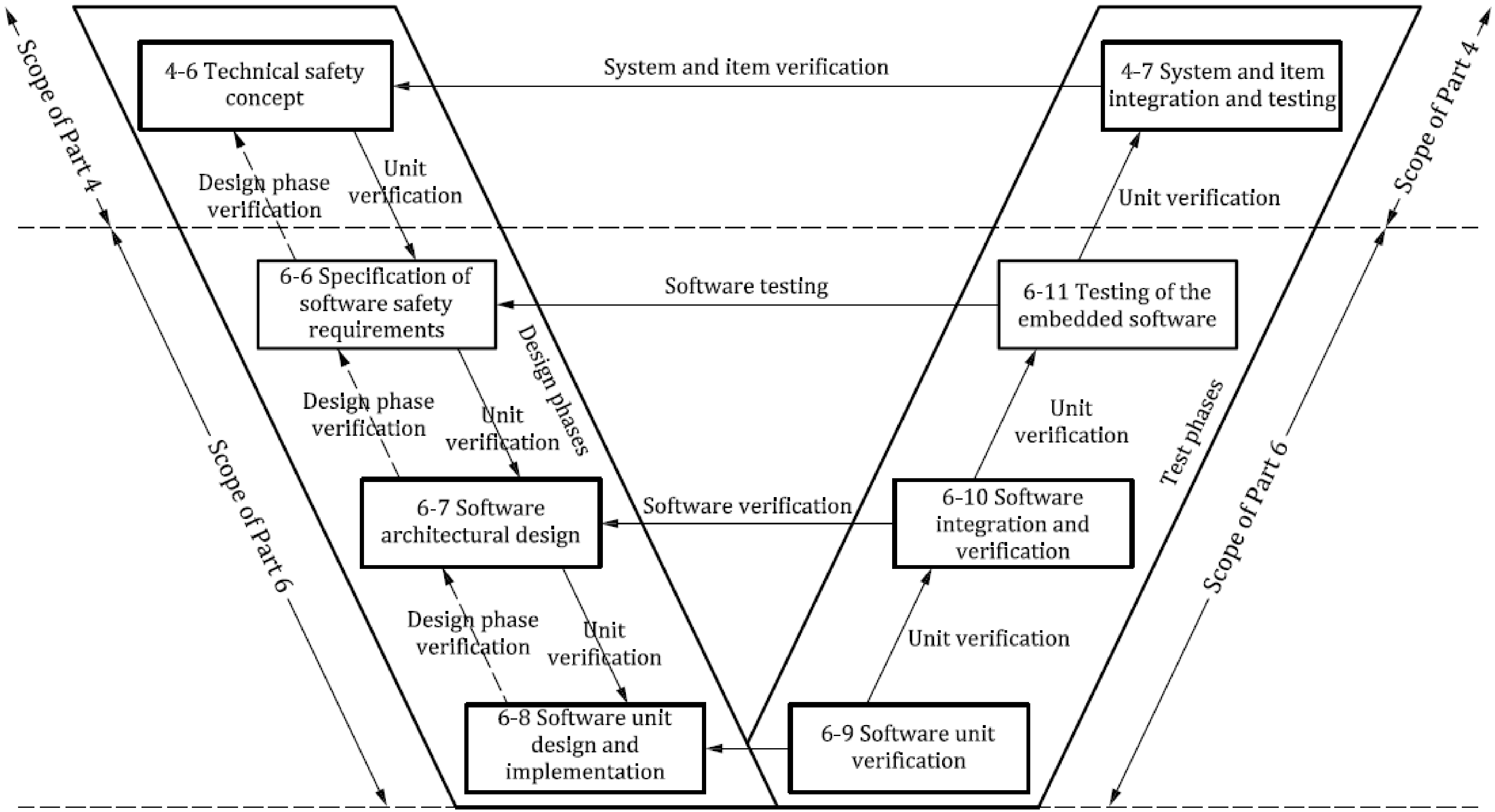
Software architectural design

Software unit design and implementation

Software integration and verification

Testing of the embedded software

Overview



4-6 Technical safety concept
System and item verification
4-7 System and item integration and testing
Design phase verification
Unit verification
Unit verification
6-6 Specification of software safety requirements
Software testing
6-11 Testing of the embedded software
Design phase verification
Unit verification
Design phases
Unit verification
Test phases
6-7 Software architectural design
Software verification
6-10 Software integration and verification
Design phase verification
Unit verification
Unit verification
6-8 Software unit design and implementation
6-9 Software unit verification
Scope of Part 4
Scope of Part 6
Scope of Part 4
Scope of Part 6

General topics for the product development at the software level

Objective

- to ensure a suitable and consistent software development process; and
- to ensure a suitable software development environment

Requirements

- Software development processes and software development environments
  - suitable for developing safety-related embedded software
  - support consistency across the sub-phases of the software development lifecycle
  - are compatible with the system and hardware development phases
- Criteria for selecting a design, modelling or programming language

Specification of the software **safety** req's

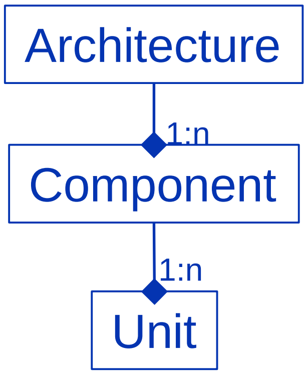
Objectives

- Specify software safety requirements derived from the technical safety concept and the system design specification
- Detail the hardware-software interface requirements
- Verify that the software safety requirements and the hw-sw interface req's are consistent with the technical safety concept and the system design spec.

Requirements

- Scope of software safety requirements
- Derivation of software safety requirements
- ASIL decomposition
- HW/SW interface specification
- Non safety related functions
- Verification of software safety requirements

Software architectural design



Objectives

1. Develop a software architectural design that realizes the software safety requirements
2. Verify the software architectural design

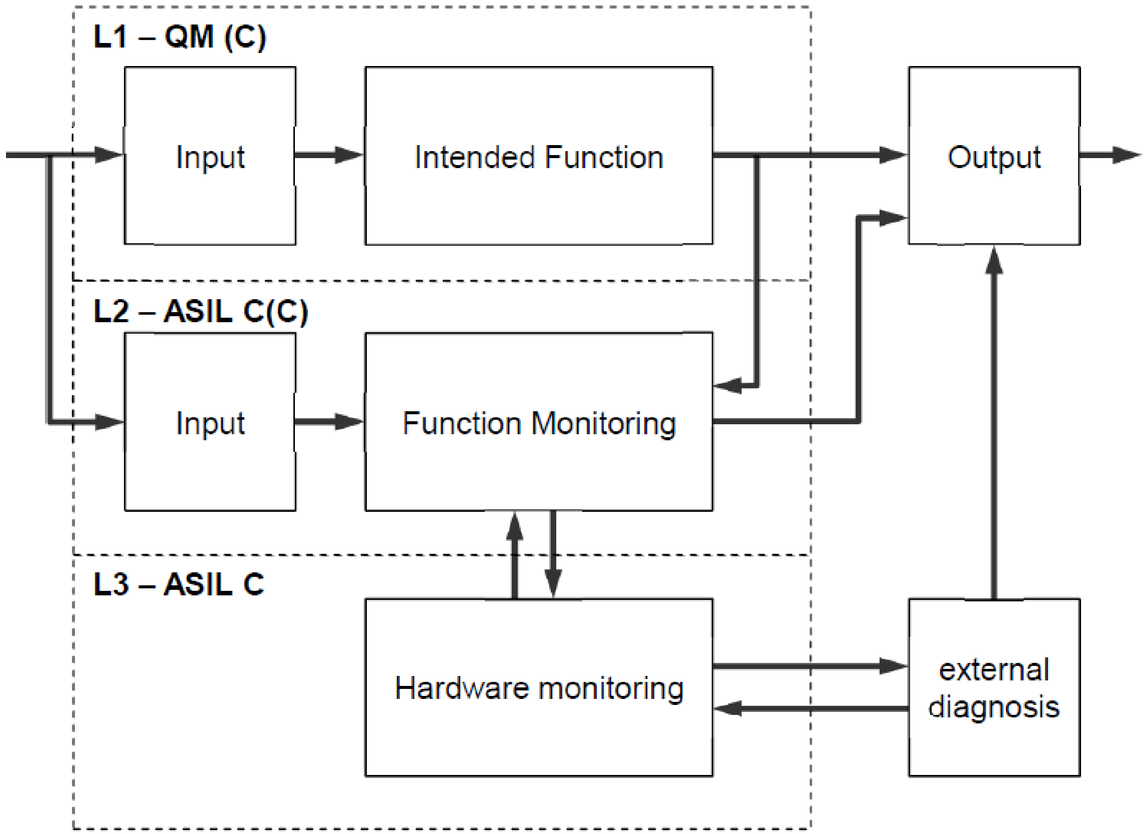
General

- The software architectural design represents all software components and their interactions in a hierarchical structure.
  - **Static aspects**, such as interfaces and data paths between all software components
  - **Dynamic aspects**, such as process sequences and timing behavior are described
- In order to develop a software architectural design both software safety requirements as well as all non-safety-related requirements are implemented.
- The software architectural design provides the means to implement the software safety requirements and to manage the complexity of the software development.

Architecture and SW Safety analysis

- Use well known architecture
- Keep it simple
- Basis for SW Safety-oriented analysis
- SW Safety-oriented analysis can be very cumbersome if at too detailed level.
- Whole Appendix E discuss this

3 LSM



Req's and recommendations

- Use of appropriate notation
- Design considerations
- Modular design
- Identification of sw units
- Design aspects
- Component categorization
- New/modified components
- Re-used components
- Allocation of Safety req's
- ASIL of combined components
- Software partitioning
- Dependent failure analysis
- Safety analysis
- Error detection
- Error handling
- New hazards
- Resource usage
- Architectural design verification

addalot QUALITY IMPROVEMENT

SW unit design and implementation

Objectives

- Develop a software unit design in accordance with the software architectural design
- Implement the software units as specified.

This sub-phase safety-related and non-safety-related requirements are handled within one development process.

Requirements

- Suitable and consistent unit design
- Unit design notation (natural, informal, semi-formal, formal)
- Specification of the software units
- Design principles for software unit design

Software unit verification

Objective

- Provide evidence that the software unit design satisfies the allocated software requirements and is suitable for the implementation

Requirements

- The software unit testing methods
- Methods for deriving software unit test cases
- Code coverage
- The test environment for software unit

addalot
QUALITY IMPROVEMENT

Software integration and testing

Objectives

- Integrate the software
- Provide evidence that the integrated software units and sw components fulfil their requirements according to the software architectural design

Requirements

- The software integration approach
- Software integration test methods
- Methods for deriving software integration test cases
- Coverage of requirements
- Methods for structural coverage
- The test environment for software integration testing

addalot QUALITY IMPROVEMENT

Testing of the embedded software

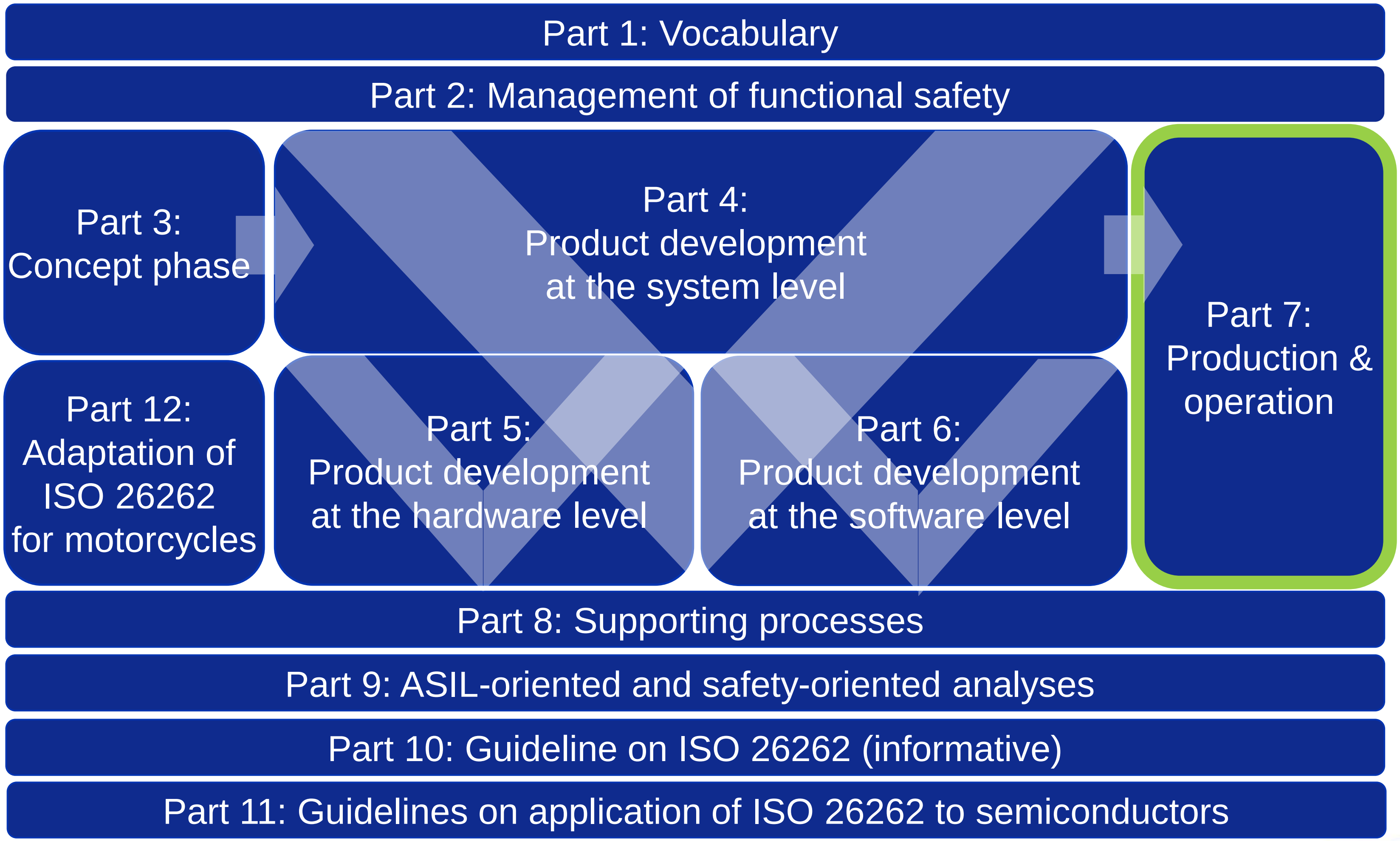
Objective

- Fulfils the safety-related requirements when executed in the target environment

Requirements

- Test environments (Hardware-in-the-loop , ECU/Bench, Vehicle)
- Methods for tests
- Methods for deriving test cases
- Evaluation of test result

Production & operation



addalot QUALITY IMPROVEMENT

Production, operation, service and decommissioning

Planning for production, operation, service and decommissioning

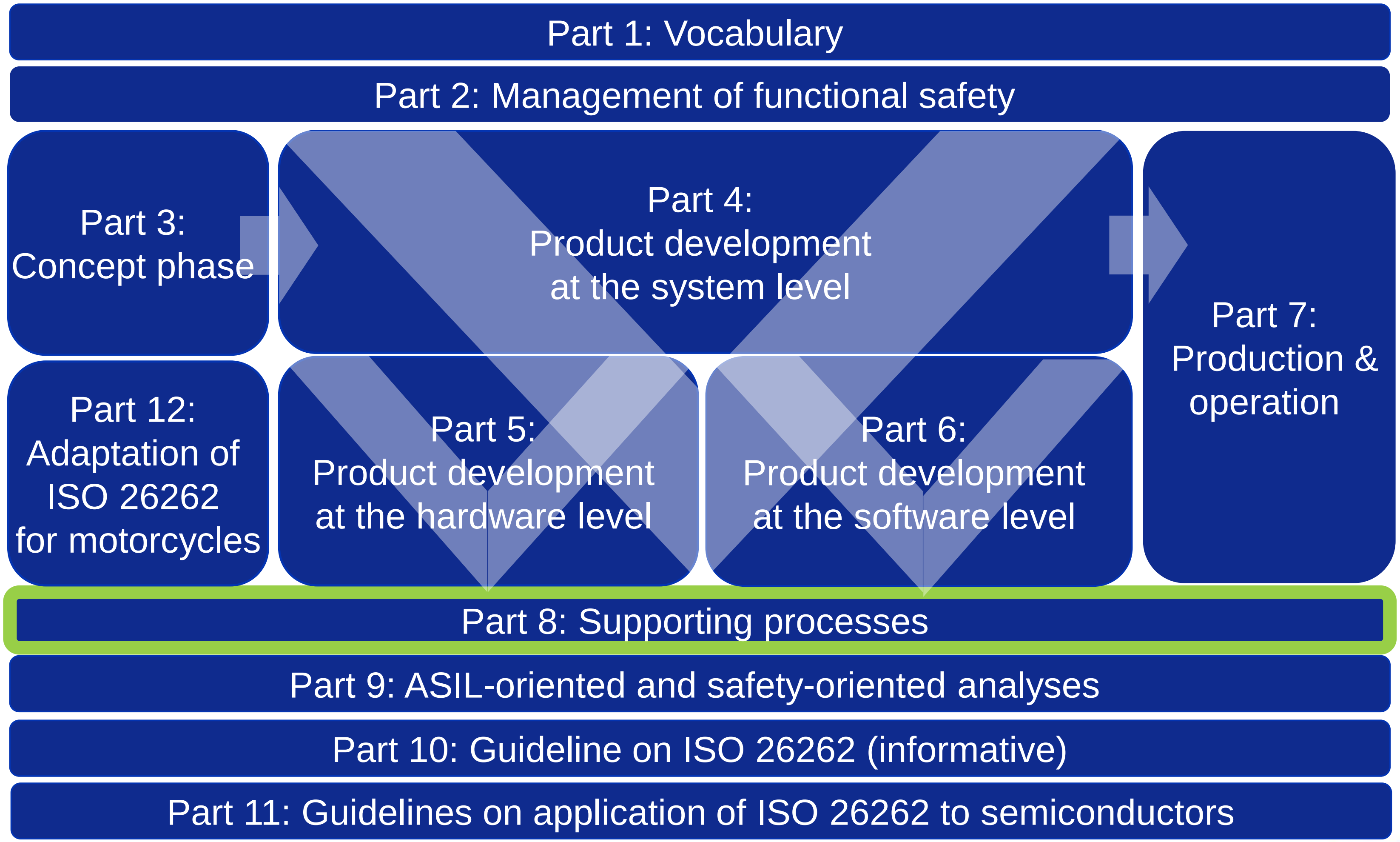
↓

Production

↓

Operation, service and decommissioning

Supporting processes



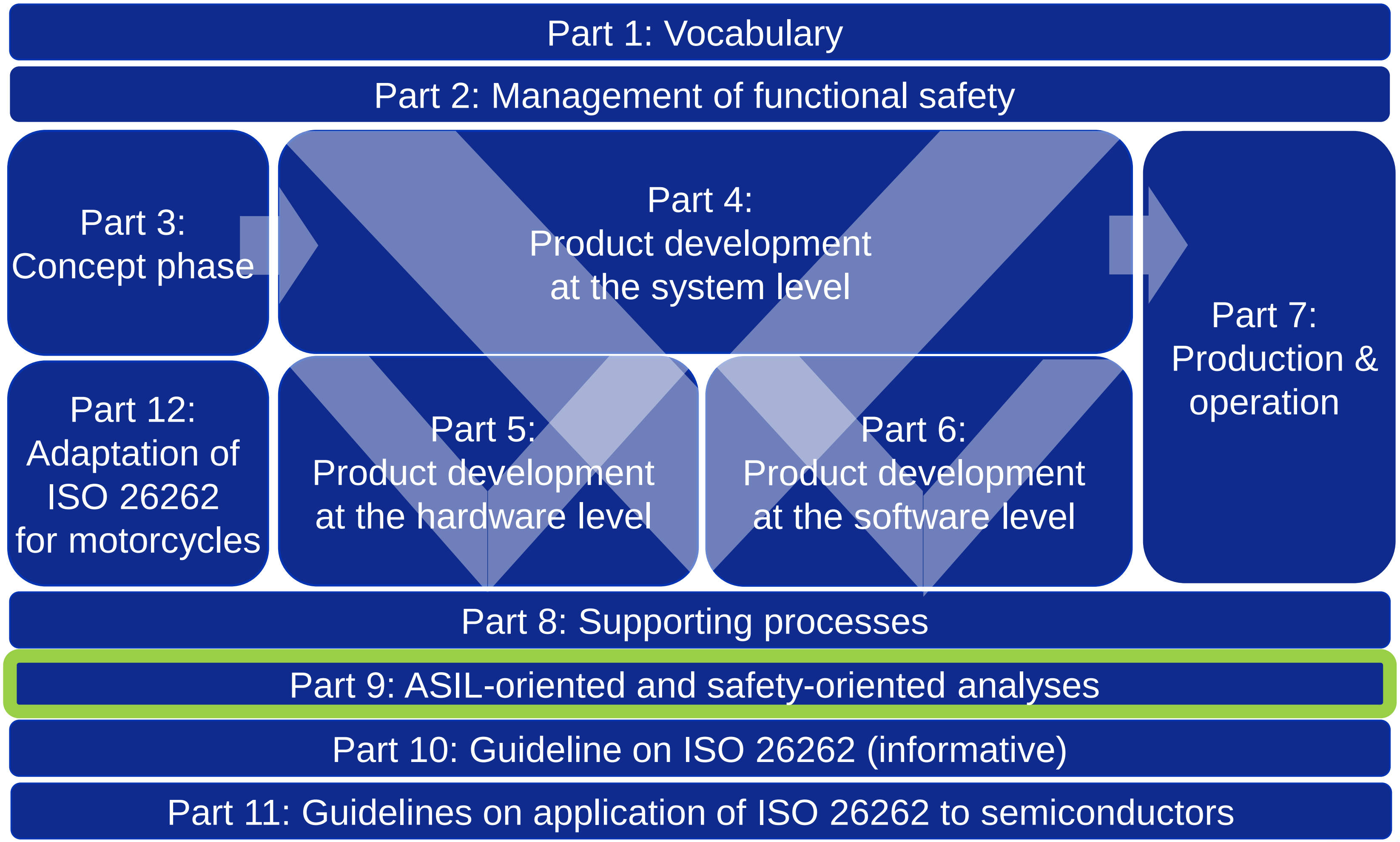
addalot
QUALITY IMPROVEMENT

Supporting processes

- Interfaces within distributed developments
- Specification and management of safety requirements
- Configuration management
- Change management
- Verification
- Documentation management
- Confidence in the use of software tools
- Qualification of software components
- Evaluation of hardware elements
- Proven in use argument
- Interfacing an application that is out of scope of ISO 26262
- Integration of safety-related systems not developed according to ISO 26262

addalot
QUALITY IMPROVEMENT

ASIL-oriented and safety-oriented analyses



ASIL-oriented and safety-oriented analyses

- Requirements decomposition with respect to ASIL tailoring
- Criteria for coexistence of elements
- Analysis of dependent failures
- Safety analysis

Interpretation of tables (1/2)

Table 2 — Notations for software architectural design Methods

	Methods	A	B	C	D
1a	Informal notations	++	++	+	+
1b	Semi-formal notations	+	++	++	++
1c	Formal notations	+	+	+	+

Alternative methods, choose **appropriate combination**

ASIL level

o = No recommendation
\+ = Recommended
++ = Highly recommended

Here it is natural to choose one, i.e. Informal notation for ASIL A and B and Semi-formal notations for ASIL B, C and D, thus for ASIL B we can chose.
If we want to use Informal notation for ASIL C or D we have to document a rationale.

addalot QUALITY IMPROVEMENT

Interpretation of tables (2/2)

Table 3 — Principles for software architectural design

	Methods	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	+	+	+	++

Here most should be used, all the time, or we need to argue why not

addalot QUALITY IMPROVEMENT

ISO 26262 and Automotive SPICE

Automotive SPICE in a nutshell

Automotive SPICE is an adaption of ISO 33001 for automotive domain with

…is a model / framework good practices being used throughout automotive industry. It describes “What” should be done” not “how”.

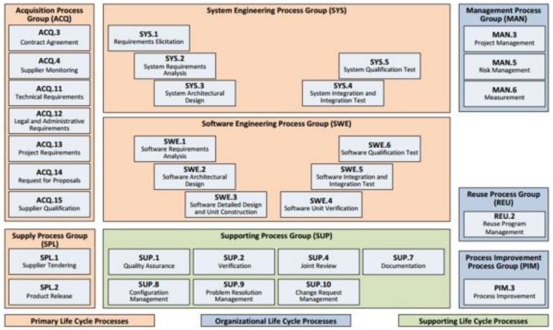
… …is a collection of process areas of the whole product life cycle: Acquisition & Supply, Systems & Software Engineering, Support & Organization, and Project & Process Management

…is a capability model for rating and improving process capability

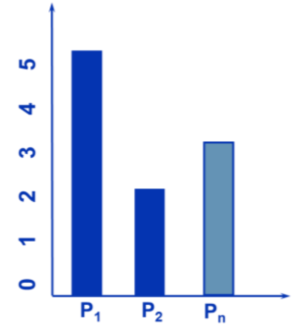
…provides guidance for improving the organization’s processes

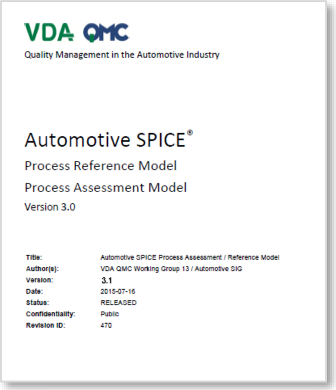
Automotive SPICE in a nutshell (cont'd)

- A set of processes and process groups



- A framework to determine process capability




VDA QMC
Quality Management in the Automotive Industry

Automotive SPICE®
Process Reference Model
Process Assessment Model
Version 3.0

Title:	Automotive SPICE Process Assessment / Reference Model
Author(s):	VDA QMC Working Group 13 / Automotive SIG
Version:	3.1
Date:	2015-07-16
Status:	RELEASED
Confidentiality:	Public
Revision ID:	470

addalot
QUALITY IMPROVEMENT

Automotive SPICE (Version 3.1)

VDA Scope

Extended VDA Scope

Acquisition Process Group (ACQ)

- **ACQ.3** Contract Agreement
- **ACQ.4** Supplier Monitoring
- **ACQ.11** Technical Requirements
- **ACQ.12** Legal and Administrative Requirements
- **ACQ.13** Project Requirements
- **ACQ.14** Request for Proposals
- **ACQ.15** Supplier Qualification

System Engineering Process Group (SYS)

- **SYS.1** Requirements Elicitation
- **SYS.2** System Requirements Analysis
- **SYS.3** System Architectural Design
- **SYS.4** System Integration and Integration Test
- **SYS.5** System Qualification Test

Management Process Group (MAN)

- **MAN.3** Project Management
- **MAN.5** Risk Management
- **MAN.6** Measurement

Software Engineering Process Group (SWE)

- **SWE.1** Software Requirements Analysis
- **SWE.2** Software Architectural Design
- **SWE.3** Software Detailed Design and Unit Construction
- **SWE.4** Software Unit Verification
- **SWE.5** Software Integration and Integration Test
- **SWE.6** Software Qualification Test

Reuse Process Group (REU)

- **REU.2** Reuse Program Management

Supply Process Group (SPL)

- **SPL.1** Supplier Tendering
- **SPL.2** Product Release

Supporting Process Group (SUP)

- **SUP.1** Quality Assurance
- **SUP.2** Verification
- **SUP.4** Joint Review
- **SUP.7** Documentation
- **SUP.8** Configuration Management
- **SUP.9** Problem Resolution Management
- **SUP.10** Change Request Management

Process Improvement Process Group (PIM)

- **PIM.3** Process Improvement

Primary Life Cycle Processes | Organizational Life Cycle Processes | Supporting Life Cycle Processes

addalot QUALITY IMPROVEMENT

Capability Levels and Process Attributes

Capability Level		Process Attributes
Level 5 - Innovated	Continuous Improvement of the Defined Process	PA - 5.2 Process Optimization
		PA - 5.1 Process Innovation
Level 4 - Predicted	Predictable performance of the Defined Process	PA - 4.2 Quantitative Control
		PA - 4.1 Quantitative Analysis
Level 3 - Established	Established a Defined Process tailored from a Standard Process	PA - 3.2 Process Deployment
		PA - 3.1 Process Definition
Level 2 - Managed	Manage that the base practices are performed	PA - 2.2 Work Product Management
		PA - 2.1 Performance Management
Level 1 - Performed	Perform all base practices	PA - 1.1 Process Performance
Level 0 - Incomplete		

addalot
QUALITY IMPROVEMENT

Purpose and scope

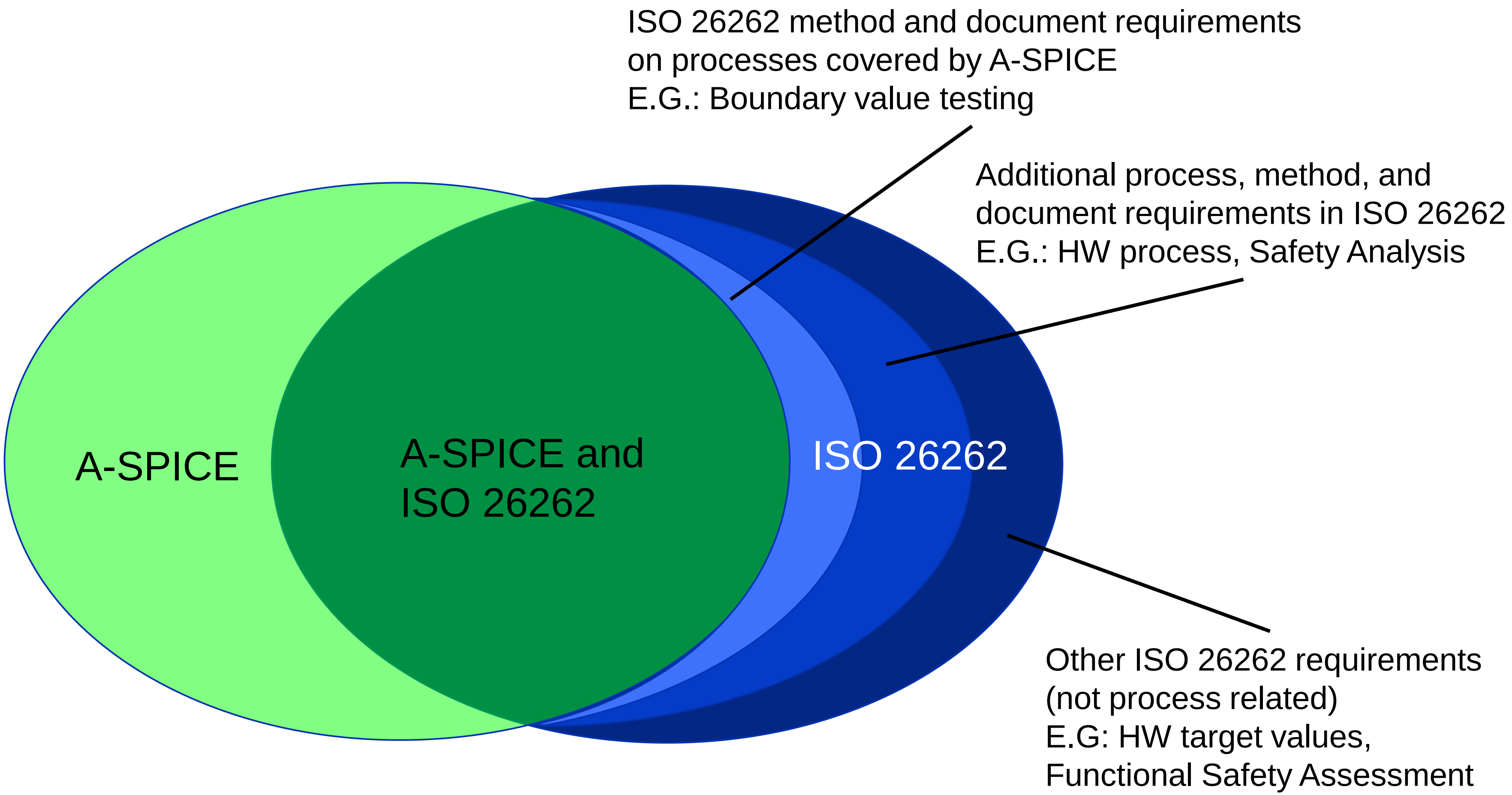
Maturity models

- Purpose
  - Improve processes based on business goals
  - Assess process capability/maturity
    - Provide assessment results that are repeatable, objective and comparable
- Scope/Coverage
  - Development (ASPICE, CMMI-DEV)
  - Development & oper. (SPICE, CMMI-SVC)
  - Products and services (CMMI)
  - Systems and software (SPICE)
  - Process capability/maturity
  - Process assessment (incl. method)

Functional safety standards

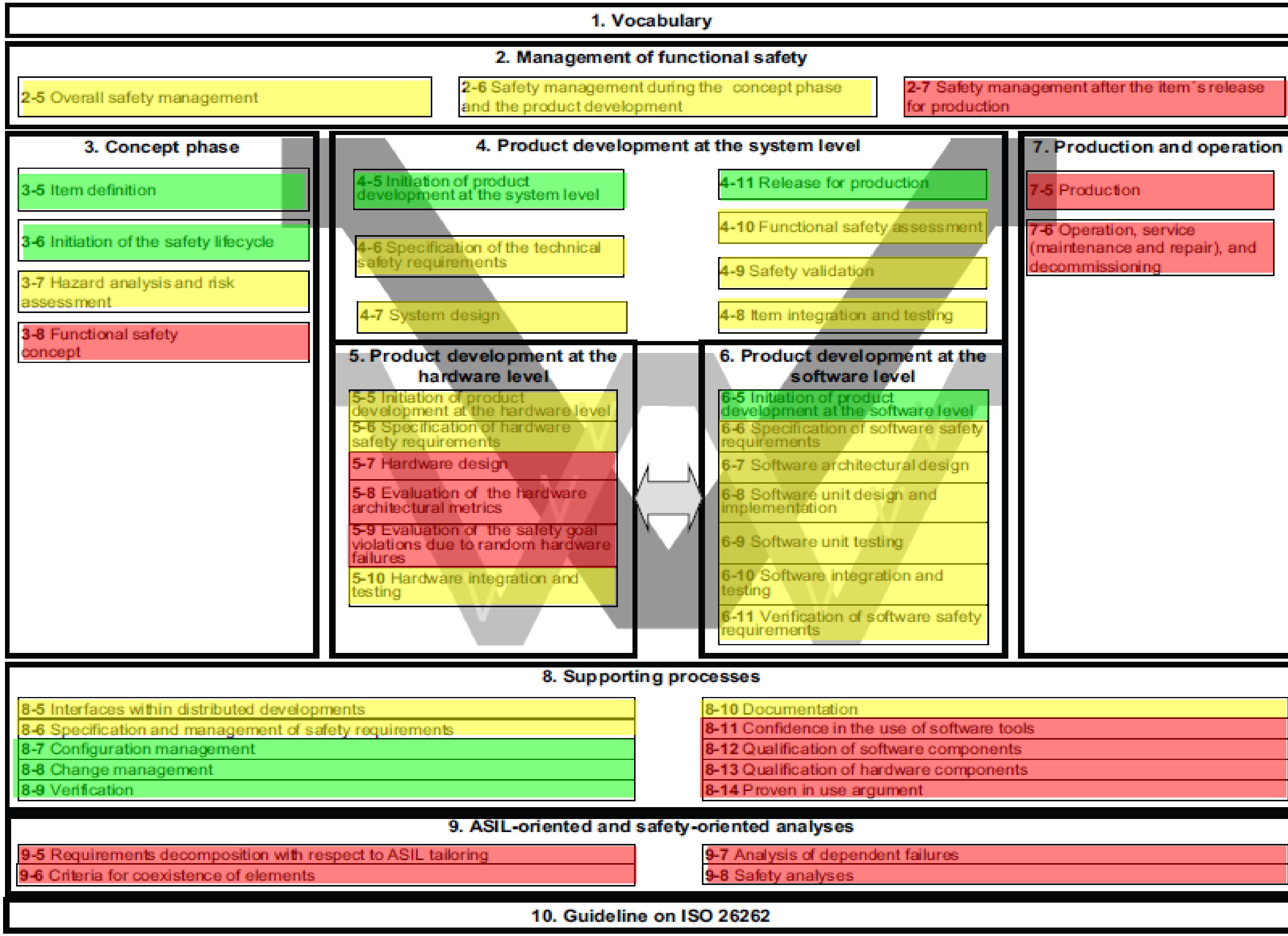
- Purpose
  - Develop safe products
  - Assess functional safety
- Scope/Coverage
  - Development, production, and operation
  - Safety critical E/E systems
  - Processes, methods and techical/product aspects
  - Safety integrity levels
  - Safety culture
  - Functional safety assessment

Coverage of A-SPICE and ISO 26262



A-SPICE support for ISO 26262

Strong support
Medium support
Weak support



A-SPICE capability levels needed for functional safety

- ISO 26262 expects that organizational process exist that are tailored for the project => many processes have to be on capability level 3.

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

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

		Management Process Group (MAN)
0		MAN.1 Organizational alignment
1		MAN.2 Organization management
3	A	MAN.3 Project management
2		MAN.4 Quality management
2	A	MAN.5 Risk management
0	A	MAN.6 Measurement

		Supply Process Group (SPL)
2	A	SPL.1 Supplier tendering
3	A	SPL.2 Product release
0		SPL.3 Product acceptance support

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

		Reuse Process Group (REU)
0		REU.1 Asset management
1	A	REU.2 Reuse program management
0		REU.3 Domain engineering

		Resource & Infrastructure Process Group (RIN)
1		RIN.1 Human resource management
2		RIN.2 Training
1		RIN.3 Knowledge management
2		RIN.4 Infrastructure

		Operation Process Group (OPE)
2		OPE.1 Operational use
0		OPE.2 Customer support

		Process Improvement Process Group (PIM)
2		PIM.1 Process establishment
1		PIM.2 Process assessment
1	A	PIM.3 Process improvement

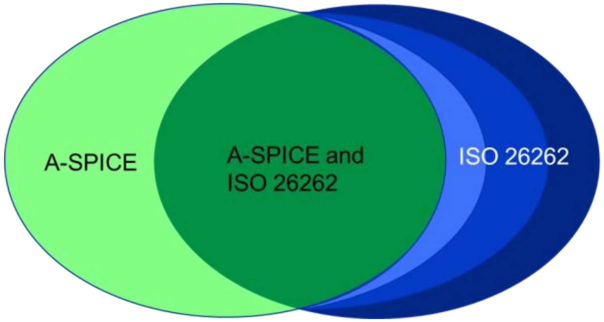
Capability Level

Reference: ISO 26262 Essentials, KMC

addalot QUALITY IMPROVEMENT

In summary

- A-SPICE and ISO26262
  - Has large overlap
  - No contradiction
  - A-SPICE can be seen as a prerequisite for ISO 26262
- A-SPICE
  - Focus on System and SW development processes
- ISO 26262
  - Focus on safe product



Self Assessment Exercise

QUALITY IMPROVEMENT

Self Assessment Exercise

Fully Implemented	**(85-100%)**	FI
Largely Implemented	**(51-84%)**	LI
Partially Implemented	**(16-50%)**	PI
Not Implemented	**(0-15%)**	NI

	ISO 26262 Requirement	Rating
1	The organization shall create, foster, and sustain a safety culture	
2	The organization shall establish a continuous improvement process	
3	The organization shall have an operational quality management system	
4	A safety case shall be developed in accordance with the safety plan	
5	An ASIL shall be determined for each hazardous event.	
6	A safety goal shall be determined for hazardous events with an ASIL.	
7	The functional safety requirements shall be derived from the safety goals	
8	The technical safety requirements shall specify necessary safety mechanisms	
9	Safety analyses on the system design to identify causes of systematic failures	
10	Diagnostic coverage of safety-related hardware elements shall be estimated	
11	Software architectural design described with appropriate levels of abstraction	
12	Every safety-related software component shall be categorized	

Implementing ISO 26262

addalot
QUALITY IMPROVEMENT

Exclusive Survey Results

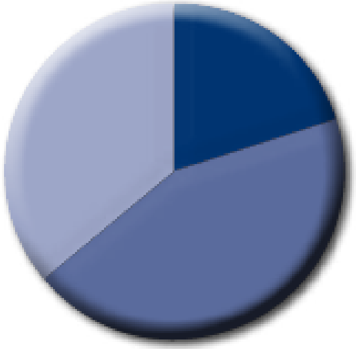
3rd International Conference Applying ISO 26262

The survey was sent out to over 600 professionals in the automotive industry. 90% of companies surveyed have at least started to implement the ISO 26262 but only 1/5 of those have managed to fully implement the standard into their processes.

20% Fully Implemented

44% Mostly Implemented

36% Starting to be Implemented



What Part of the ISO 26262 does your organization find most challenging?

32.14% Safety Management for the Organization

3.57% Concept Phase

32.14% System Development with Technical Safety Concepts

7.14% Product Development at the Hardware Level

10.71% Product Development at the Software Level

14.29% ASIL & Safety-Oriented Analysis (ex. FTA)

What is highest ASIL category your company is dealing with?

0 % ASIL A

15,38% ASIL B

3,85% ASIL C

80,77% ASIL D

"Made us think more about human factors."

Safety/ISO 26262 specific challenges

- Establish **safety culture**
- A-SPICE **Level-3** capability needed for many **processes**
- **Safety analysis** techniques, e.g. HARA, FTA, FMEA
- **Design for safety**, i.e. design patterns, HW/SW design
- **Test methods**, e.g., fault injection, struct coverage, and equivalence classes
- Handling/qualification of **legacy systems**, SW&HW **components**, **tools**
- Development of **safety case**
- Functional safety assessment
- Many **organizational** parts involved
  - System, SW, HW, Test, Production, Legal, Sourcing

A typical improvement journey

Why? What? When? How much?	Where are we today?	How?	Just do it…
Business goals Improvement goals Improvement metrics	Appraisal results Measurements	Improvement approach Improvement plan	Change management

Common Improvement Pitfalls

- Improvement goals are not aligned with business goals
- Management not committed to Improvement
  - Adequate resources not provided
  - Premature delegation of process improvement responsibilities
- Process Theory
  - Improvement run from an Process Group away from projects
  - Neglecting existing practices
  - Lots of diagrams but little content
- Overconfidence in or misinterpretation of models
  - There are no ”silver bullets”
  - The check list syndrome
- Everything done at the same time - big bang strategy
- Neglecting the “human side” of the change
  - People change not organizations

Specific recommendations

- Takes time….
  "1-step" in A-Spice takes 9-12 months at 5-10% of the engineering capacity
- Don't separate A-SPICE/ISO implementation
  - Same Standard process and same people
- Take it in steps - what order? …It depends…
  - Establish and ensure usage of standard process
  - Initiate Safety culture/activities
- ISO 26262 require more top down
- Process deployment >> Process definition
- Drive introduction as project with clear goals and follow up
- Don't neglect emotional aspects – "what's in it for me"
- Communication

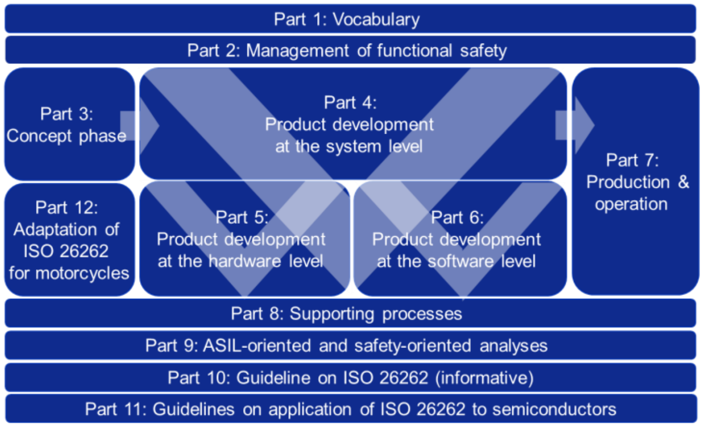
addalot
QUALITY IMPROVEMENT

Prerequisites for change

Summary

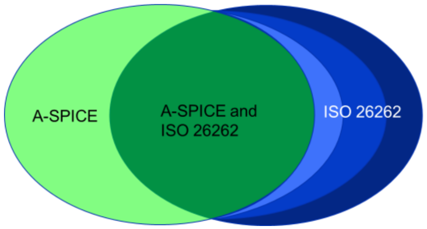
Summary

- ISO 26262 is expected by the Automotive industry as “State of the art”
- Extensive standard covering several areas:



- Required degree of safety measures:

ASIL= Severity x Exposure X Controllability



- Fit well together with Automotive SPICE
- Challenge for Improvement to be successful

“Excellent firms don't believe in excellence - only in constant improvement and change.”

In Search of Excellence - Tom Peters

Even-andre.karlsson@addalot.see
+46 706 800 533

addalot
QUALITY IMPROVEMENT