Software Safety

Safety Standards

Prof. Dr.-Ing. Patrick Mäder, M.Sc. Martin Rabe

Contents

1. Safety Standards

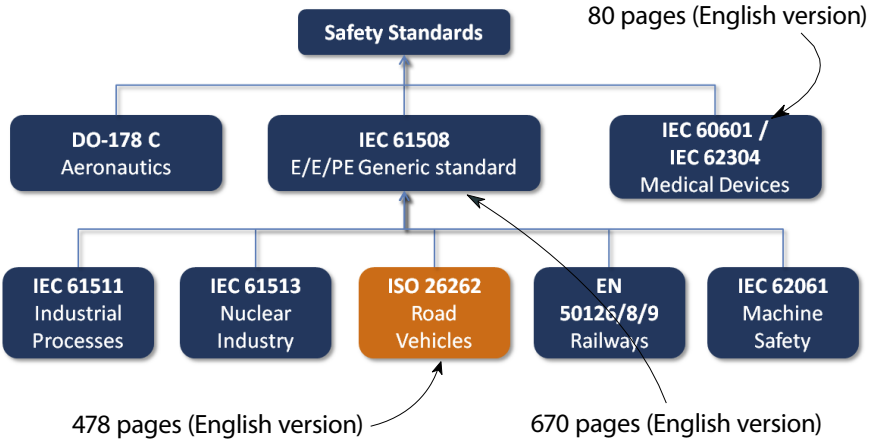
2. The IEC 61508 Standard

3. The ISO 26262 Standard

4. Summary

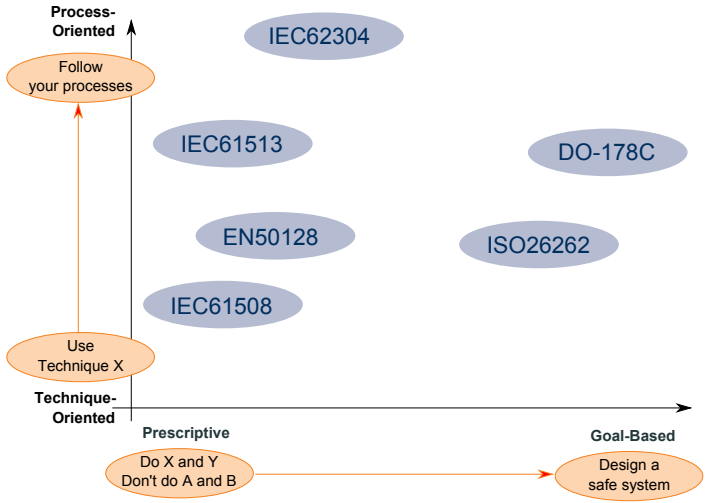
Safety Standards

A Selection of Safety Standards



E/E/PE = Electrical/Electronic/Programmable Electronic

Categorization of Safety Standards

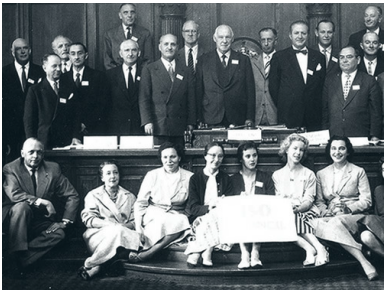


[Hobbs, C. (2019). Embedded Software Development for Safety-Critical Systems (2nd ed.). CRC Press.]

Organisations for Standardization

- International Organisation for Standardization (ISO) Founded in London in 1946
- International Electrotechnical Commission (IEC) Founded in 1906
- Both based in Geneva
- Each country gets one vote: Germany gets one vote, China gets one vote, Ethiopia gets one vote. But the Vatican City doesn't get a vote
- Collaborate as World Standards Cooperation (WSC) since 2007

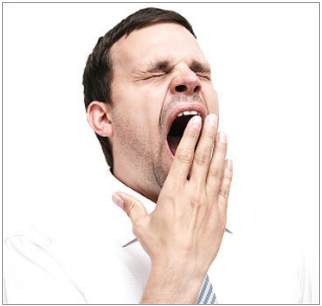
ISO International Organization for Standardization IEC



[Latinfundist Blog]

Safety Standards Discussed in this Course

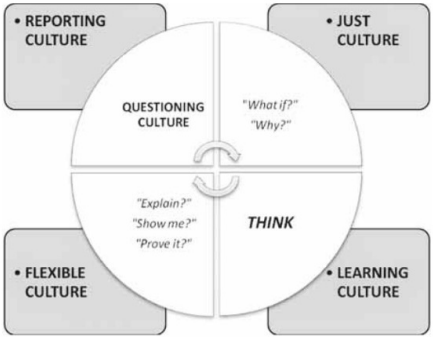
IEC 61508 and ISO 26262



[Glogster]

The Importance of Safety Culture

Standards call for a “safety culture” within the development organization.



[Nimrod report]

The IEC 61508 Standard

The IEC 61508 Standard

IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems (E/E/PE) (first version in 2000):

- concerns **functional safety** of developed systems

The IEC 61508 Standard

IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems (E/E/PE) (first version in 2000):

- concerns **functional safety** of developed systems
- intended to be the basis for other standards, e.g., ISO 26262

The IEC 61508 Standard

IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems (E/E/PE) (first version in 2000):

- concerns **functional safety** of developed systems
- intended to be the basis for other standards, e.g., ISO 26262
- defines a set of **Safety Integrity Levels (SILs)**

SIL	Probability of failure per hour (continuous operation)	Probability of failure on demand (on-demand operation)
1	$\geq 10^{-6}$ *to* 10^{-5}	$\geq 10^{-2}$ *to* 10^{-1}
2	$\geq 10^{-7}$ *to* 10^{-6}	$\geq 10^{-3}$ *to* 10^{-2}
3	$\geq 10^{-8}$ *to* 10^{-7}	$\geq 10^{-4}$ *to* 10^{-3}
4	$\geq 10^{-9}$ *to* 10^{-8}	$\geq 10^{-5}$ *to* 10^{-4}

The IEC 61508 Standard

IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems (E/E/PE) (first version in 2000):

- concerns **functional safety** of developed systems
- intended to be the basis for other standards, e.g., ISO 26262
- defines a set of **Safety Integrity Levels (SILs)**
- defines sets of:
  - **Processes**, e.g., impact analysis, regression testing
  - **Techniques**, e.g., (semi-)formal methods, boundary value analysis
  - **Tools**, e.g., certified code translator, coding standards, fault tree analysis

  that are **not recommended**, **recommended** or **highly recommended** at each SIL.

Exercise

- IEC 61508 defines SIL4 to mean that the probability of failure per hour is less than 10^{-8}. 10^8 hours is 11,408 years. Is it possible to justify a claim that a particular system meets SIL4?

- Section 7, appendix D of IEC 61508 provides a formula for random testing for a predefined period as $T = \frac{-\ln(\alpha)}{\lambda}$ where $1 - \alpha$ is a required confidence level and λ is the maximum tolerable probability of failure per hour

Exercise

- IEC 61508 defines SIL4 to mean that the probability of failure per hour is less than 10^{-8}. 10^8 hours is 11,408 years. Is it possible to justify a claim that a particular system meets SIL4?

- Section 7, appendix D of IEC 61508 provides a formula for random testing for a predefined period as $T = \frac{-\ln(\alpha)}{\lambda}$ where $1 - \alpha$ is a required confidence level and λ is the maximum tolerable probability of failure per hour
- **Result**: for SIL4
  - Confidence level $(1 - \alpha)$ 95%: 34,174 years (test duration T)
  - Confidence level $(1 - \alpha)$ 98%: 44,627 years (test duration T)

The ISO 26262 Standard

ISO 26262: The Automotive E/EE Safety Lifecycle Standard

ISO 26262: Road vehicles – Functional safety

- focusses on functional safety in the automotive domain (see next slide for scope)
- initial release: 2011, current version: ISO 26262:2018
- defines **safety** as *"absence of unreasonable risk"*.
- defines **unreasonable risk** as "risk judged to be unacceptable in a certain context according to valid societal moral concepts".
- defines **risk** as *"combination of the probability of occurrence of harm and the severity of that harm"*.

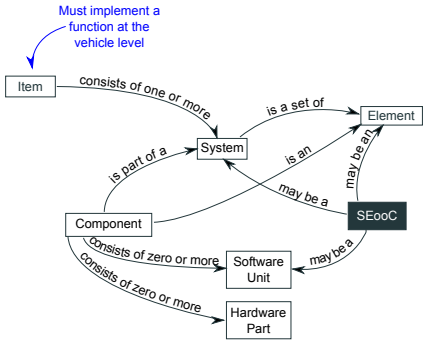
ISO 26262: Scope

- *"The ISO 26262 series of standards is the **adaptation of IEC 61508** series of standards to address the sector specific needs of **electrical and/or electronic (E/E) systems within road vehicles**."*
- applied to safety-related systems that include … electrical and/or electronic (E/E) systems … **installed in series production road vehicles, excluding mopeds**
- does not address unique E/E systems in special vehicles … for drivers with disabilities

[A Dairy Crest ex-Unigate Wales & Edwards Rangemaster milk float., Wikipedia]

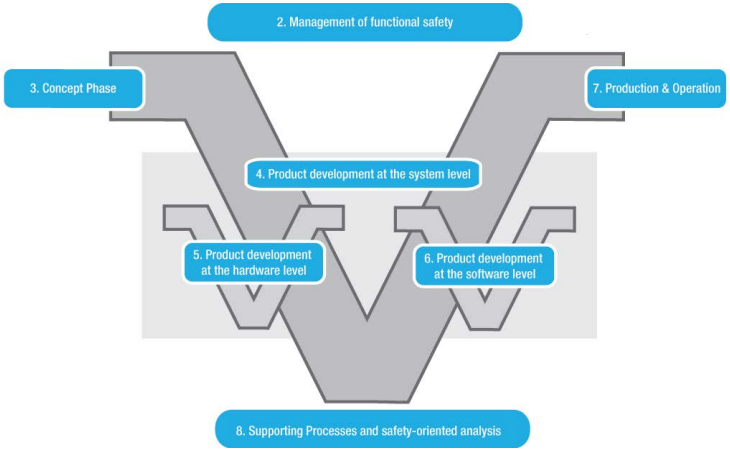
ISO 26262: Items, Elements, and SEooCs

- **Item**: system or array of systems to implement a function at the vehicle level
- **Element**: system or part of a system including components, hardware, software, hardware parts, and software units
- **Safety Element out of Context (SEooC)**: a safety-related element which is not developed for a specific item … not developed in the context of a particular vehicle



[Hobbs, C. (2019). **Embedded Software Development for Safety-Critical Systems (2nd ed.)**. CRC Press.]

ISO 26262: Structure



[ISO 26262]

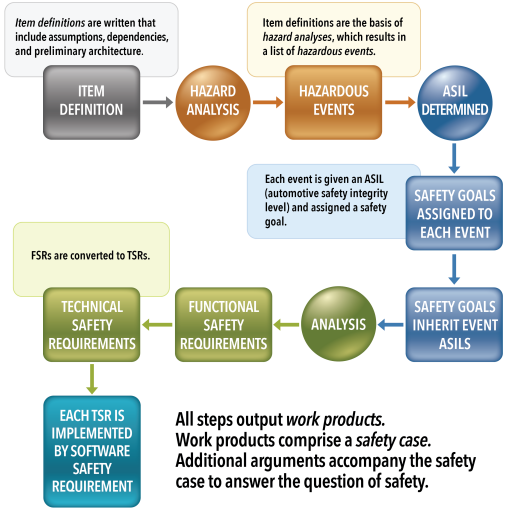
1. Vocabulary
2. Management of functional safety
3. Concept
4. System
5. Hardware
6. Software
7. Production
8. Supporting processes
9. ASIL analyses
10. Guideline on ISO 26262
11. Application to semiconductor
12. Adaption for motorcycles

What does ISO 26262 demand?

That we demonstrate that:

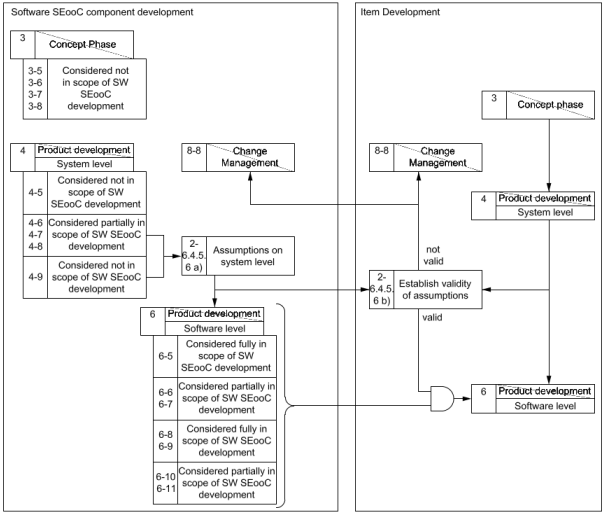
- the product is safe if used in accordance with the Safety Manual ← to some extent it's a goal-based standard.
- a predefined development approach is followed: Concept, System-Level Development, Software-Level Development, Production and Operating, Decommissioning.
- "recommended" or "highly recommended" tools and techniques are used or justification is given for doing something else.

ISO 26262: Software Development Lifecycle



[OpenSystems Media: Leveraging automotive development standards to mitigate risk, part 2, Embedded Computing Design

Requirements on a SEooC



[ISO 26262 Part 10, Figure 21]

ASIL Basics

- **Automotive Safety Integrity Level (ASIL)** – expresses the criticality associated with a function of the system

- Range from A (lowest) to D (highest)
- The ASIL is not calculated for a physical system component – it is calculated for a **function**
- Software and hardware elements that realize the function inherit the ASILs of function they realize
- When realizing multiple function, the highest ASIL is inherited

[Synopsys, 2017]

- The indirect correlation of ISO 26262 to IEC 61508 safety integrity levels
- ISO 26262 is “not a reliability standard” – no precise numbers for acceptable probabilities of failure

Let’s calculate an ASIL

Step 1a: Assess Severity

- AIS 1: light injuries such as skin-deep wounds, muscle pains, whiplash, etc.
- AIS 2: moderate injuries such as deep flesh wounds, concussion with up to 15 minutes of unconsciousness, uncomplicated long bone fractures, etc.
- AIS 3: severe but not life-threatening injuries such as skull fractures without brain injury, spinal dislocations below the fourth cervical vertebra, etc.
- AIS 4: severe injuries (life-threatening, survival probable)
- AIS 5: critical injuries (life-threatening, survival uncertain) such as spinal fractures below the fourth cervical vertebra with damage to the spinal cord, etc.
- AIS 6: extremely critical or fatal injuries such as fractures of the cervical vertebrae above the third cervical vertebra with damage to the spinal cord, etc.

Let's calculate an ASIL

Step 1b: Determine Severity Class

Table 1 — Classes of severity

	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

- S0: < 10% probability of AIS 1-6
- S1: > 10% probability of AIS 1-6 (and not S2 or S3)
- S2: > 10% probability of AIS 3-6 (and not S3)
- S3: > 10% probability of AIS 5-6

[Photobucket]

Let’s calculate an ASIL

Step 2: Determine Probability Class

Table 2 — Classes of probability of exposure regarding operational situations

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

- E1: Occurs less often than once a year for the great majority of drivers
- E2: Occurs a few times a year for the great majority of drivers
- E3: Occurs once a month or more often for an average driver
- E4: Occurs during almost every drive on average

Ok, let's calculate an ASIL

Step 3: Determine Controllability Class

Table 3 — Classes of controllability

	Class			
	C0	C1	C2	C3
Description	Controllable in general	Simply controllable	Normally controllable	Difficult to control or uncontrollable

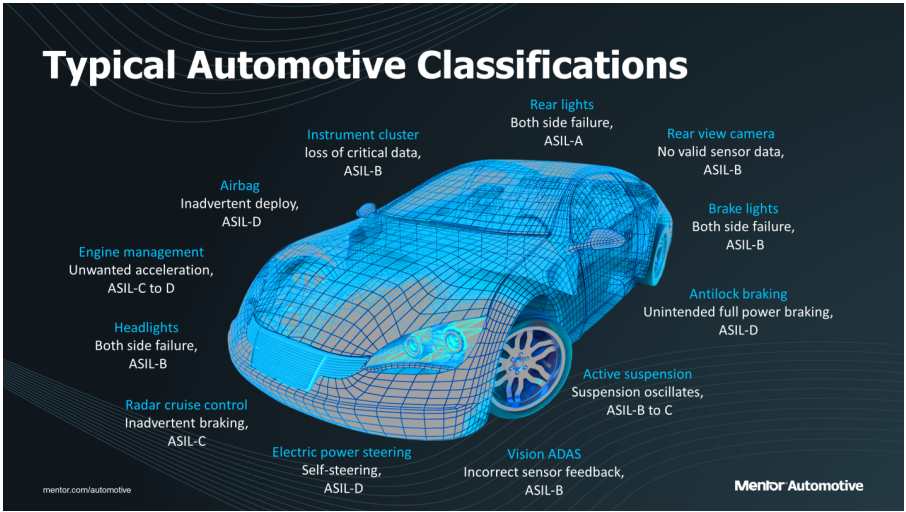
- C0: Controllable in general
- C1: 99% or more of drivers or other participants are usually able to avoid harm
- C2: 90% or more of drivers or other participants are usually able to avoid harm
- C3: Fewer than 90% of drivers or other traffic participants are usually able, or barely able, to avoid harm

So, what's the A-SIL?

		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

QM → Quality Management (ISO 26262 prescriptions do not apply)

Typical ASIL Classifications



[Mentor Automotive: Typical Automotive Classifications , Semiconductor Engineering, 2017]

Component Reuse: ASIL by “Proven-In-Use”

Basis for use of a component or system as part of a safety integrity level (SIL) rated safety instrumented system (SIS) that has **not been designed in accordance with IEC 61508**. It requires sufficient product operational hours, revision history, fault reporting systems, and field failure data to determine if the is evidence of systematic design faults in a product.

Table 7 — Limits for observable incident rate

ASIL	Observable incident rate
D	$<10^{-9}$/h
C	$<10^{-8}$/h
B	$<10^{-8}$/h
A	$<10^{-7}$/h

10^9 hours $\approx$ 114,077 years.

Assuming $F = 1e^{ft}$, probability of failure of an ASIL-D system in 114,077 years is about 0.6.

Note – we need enough results for statistical significance (10 or more occurrences). Using PIU data in a Safety Case can be very powerful (because it is real) but it can **take a long time to gather and bias is likely** to be introduced into the numbers. How accurate are the hours of field usage? What fraction of the failures that have happened in the field have been reported?

Freedom from interference (1/3)

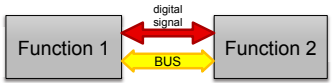
Definition (ISO 26262, Part 1, Definition 1.49): Absence of **cascading failures** between two or more **elements** that could lead to the violation of a safety requirement.

- **cascading failure** = “failure of an element of an item causing another element or elements of the same item to fail” (ISO 26262, Part 1, Definition 1.13)
- comprises (1) assigning safety requirements to architecture elements, (2) software partitioning, (3) analysis of dependent failures

D.2.2 Timing and execution

D.2.3 Memory

D.2.4 Exchange of information



[Stefan Hoch, ICS, 2014]

Freedom from interference (2/3)

ISO 26262-6, annex D

2. Timing and execution
   - blocking of execution
   - deadlocks
   - livelocks
   - incorrect allocation of execution time
   - incorrect synchronization between software elements
3. Memory
   - corruption of content
   - read or write access to memory allocated to another software element

Freedom from interference (3/3)

ISO 26262-6, annex D

- D.2.4 Exchange of information
  - repetition, loss or delay of information
  - insertion of information
  - masquerade or incorrect addressing of information
  - incorrect sequence of information
  - corruption of information
  - asymmetric information sent from a sender to multiple receivers
  - information from a sender received by only a subset of the receivers
  - blocking access to a communication channel

Example: Modelling and Coding

For each ++ not done, a concession and a justification is needed.

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	++	++	++	++

Example: Design Verification

Table 6 — Methods for the verification of the software architectural design

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

For each ++ not done, a concession and a justification is needed.

Example: Unit and Integration Testing

Table 12 — 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)	+	+	+	++

Table 13 — Methods for software integration testing

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

Don’t forget ISO 29119: Software Testing

IEC 61508 Example: Quantification

Table D.1 – Necessary history for confidence to safety integrity levels

SIL	Low demand mode of operation	Number of treated demands		High demand or continuous mode of operation	Hours of operation in total	
	(Probability of failure to perform its design function on demand)	1-α = 0,99	1-α = 0,95	(Probability of a dangerous failure per hour)	1-α = 0,99	1-α = 0,95
4	$\geq 10^{-5}$ to $< 10^{-4}$	$4{,}6 \times 10^5$	3×10^5	$\geq 10^{-9}$ to $< 10^{-8}$	$4{,}6 \times 10^9$	3×10^9
3	$\geq 10^{-4}$ to $< 10^{-3}$	$4{,}6 \times 10^4$	3×10^4	$\geq 10^{-8}$ to $< 10^{-7}$	$4{,}6 \times 10^8$	3×10^8
2	$\geq 10^{-3}$ to $< 10^{-2}$	$4{,}6 \times 10^3$	3×10^3	$\geq 10^{-7}$ to $< 10^{-6}$	$4{,}6 \times 10^7$	3×10^7
1	$\geq 10^{-2}$ to $< 10^{-1}$	$4{,}6 \times 10^2$	3×10^2	$\geq 10^{-6}$ to $< 10^{-5}$	$4{,}6 \times 10^6$	3×10^6
NOTE 1 1-α represents the confidence level.						
NOTE 2 See D.2.1 and D.2.3 for prerequisites and details of how this table is derived.						

For continuous operation $t \approx \frac{-ln(\alpha)}{\lambda}$ for small failure rates.

How much testing do I need to do to be 90% confidence that my device meets SIL2?

Example Calculation

For continuous operation $t \approx \frac{-ln(\alpha)}{\lambda}$ for small failure rates.

How much testing do I need to do to be 90% confidence that my device meets SIL2? $t \approx \frac{-ln(\alpha)}{\lambda} = \frac{-ln(0.1)}{10^{-6}} \approx 2302585$

2,302,585 hours is about 263 years. With 100 devices in the test laboratory, we need 2 years 8 months of error-free operation.

Lowering the ASIL

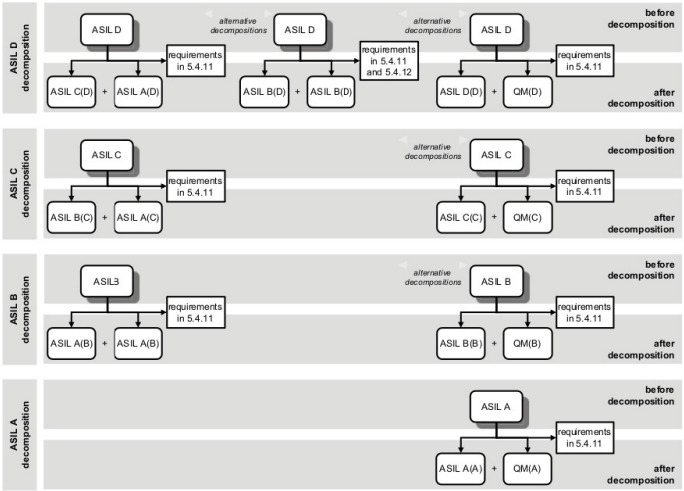
- Under certain circumstances, the ASIL can be lowered through the technique of **ASIL Decomposition**
- The concept already existed in IEC 61508 – it is not entirely new!
- This can be advantageous – for example, with respect to production costs
  - It usually costs less (labor, time, tools) to develop according to a lower ASIL
- But there are strict underlying concepts and rules that must be respected

[Favaro 2011]

ASIL Decomposition Basics

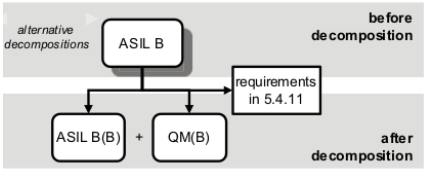
- An element implemented to address a given safety goal, with a given ASIL may be decomposed into **two independent** elements, with possibly lower ASIL
  - Each must address the **same safety goal**
  - And each must take on the **same safe state**
- ASIL Decomposition can be used in the following phases
  - Functional safety concept
  - System design
  - Hardware design
  - Software design
- ASIL decomposition is a **qualitative** concept, more addressing systematic issues (architecture) than random errors (hardware reliability)
  - It can be a way of making architectures more robust
  - Similar to 61508 fault-tolerant architecture concepts

Valid ASIL Decompositions

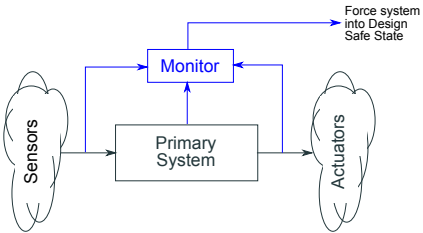


[ISO 26262]

ASIL Decomposition Example



[ISO 26262]



[Hobbs, C. (2019). Embedded Software Development for Safety-Critical Systems (2nd ed.). CRC Press.]

Summary

Summary

- The IEC 61508 standard deals with functional safety and defines safety integrity levels with a set of recommended and highly recommended processes and techniques
- The ISO 26262 standard build upon IEC 61508 and prescribes an automotive development lifecycle for safety-critical E/EE systems

Questions?