Cross-domain systems and safety engineering: is it feasible?

Composable safety: is it feasible?

Eric.Verhulst@altreonic.com
Altreonic NV

Content

- Intro to safety and its importance
- Some results from OPENCOSS project
- The issue with the SIL concept
- Introducing the ARRL concept
- A unified process pattern
- Conclusions

Some data for thought

- 35000 people killed in cars /yr /Europe
- 500 people killed in airplanes /yr /world
  - Why the difference? => many reasons
- The Renault Logan is the most reliable car
  - Why? Less electronics, proven in use design
  - Is it also safer?
- The US is considering to make black boxes a legal requirement in cars
  - What does this mean? What could be the impact?

Systems Engineering vs. Safety Engineering

- System = holistic
- Real goal is "Trustworthy Systems"
  - Cfr. Felix Baumgartner almost did not do it because he didn't trust his safe jumpsuit
- TRUST = by the user or stakeholders
  - Safety
  - Security
  - Usability (UI)
  - Privacy
  - Achieving intended Functionality
  - Meeting non-functional objectives
    - Cost, energy, volume, maintainability, scalability, Manufacturability,..
- So why focus on safety?

Safety

- **Safety** is the state of being "safe" the condition of being protected against physical, social, spiritual, financial, political, emotional, occupational, psychological, educational or other types or consequences of failure, damage, error, accidents, harm or any other event which could be considered non-desirable.
- **Safety** can also be defined to be the **control of recognized hazards** to achieve an **acceptable level** of **risk**.
- Safety is general property of a system
- Not just a concern when programmable elecronics are used !!!
- What are the consequences of not meeting the requirements?
  - Annoyance
  - Business lost
  - People can get killed or harmed
  - => it is complex but there are moral liabilities

Role of certification

- In depth review => safe to operate
  - “Conformity assessment” (for automotive)
- Not a technical requirement:
  - Confidence requirement
  - Legal requirement
  - Where is the evidence?
- Evidence:
  - Evidence is a **coherent** collection of **information** that relying on a number of **process artifacts** linked together by their **dependencies and sufficient structured arguments** provides an **acceptable proof** that a specific system goal has been reached.

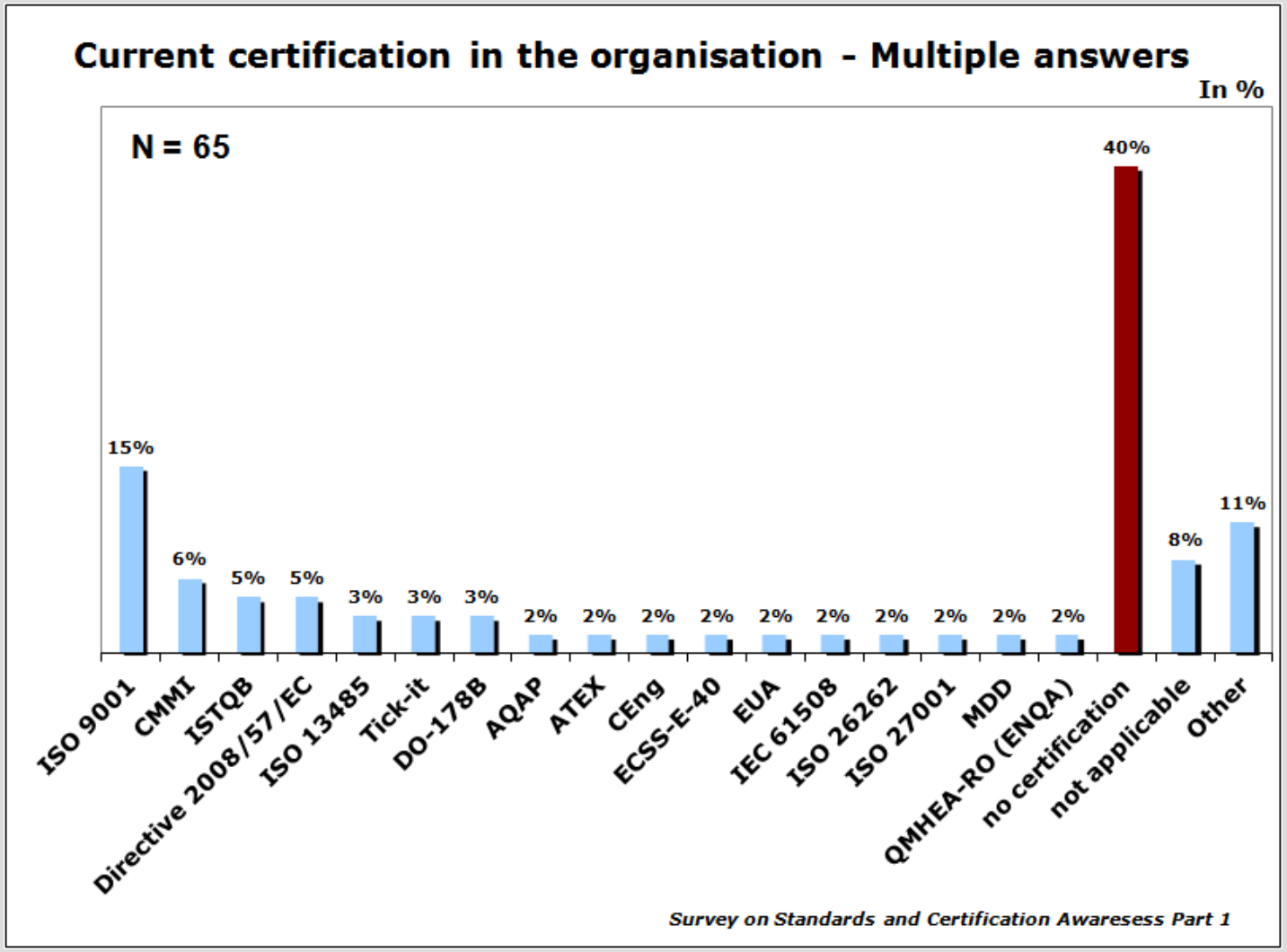
Some background projects

- ASIL project
  - Project with Flanders Drive to develop a common "automotive" safety engineering methodology
  - IEC-61508, IEC-62061, **ISO-26262**, ISO-13849, ISO-25119 and ISO-15998. (+ CMMI, Automotive SPICE)
  - About 350 steps, 100 workproducts, ...
  - ASIL imported in GoedelWorks portal
- EU FP7 IP OPENCOSS
  - Project with 17 EU partners (avionics, raillway, automotive) on reducing the cost and effort of certification
    - ISO-26262, DO-178C/254/..., CENELEC 50126-128-129
    - Cross-domain
    - Product families
- => there is interest and a growing awareness

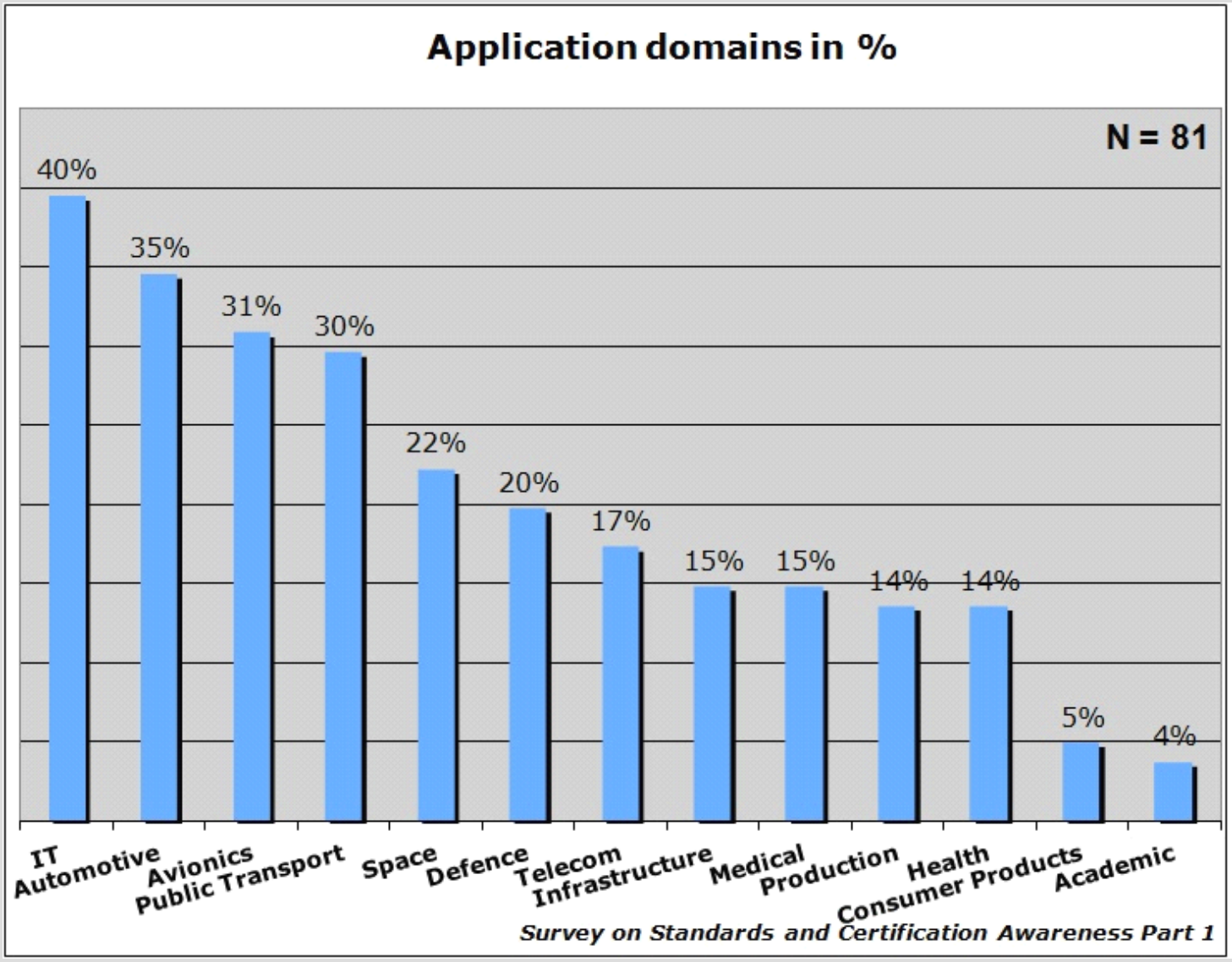
- LinkedIn discussion grops

Altreonic's OPENCOSS survey

- First survey on standards and certification awareness done. (public, 85 respondents)
- In-depth interviews executed: Alstom, Thales Toulouse, Thales Valence, Renault, CRF
- Cross domain workshop to be organised

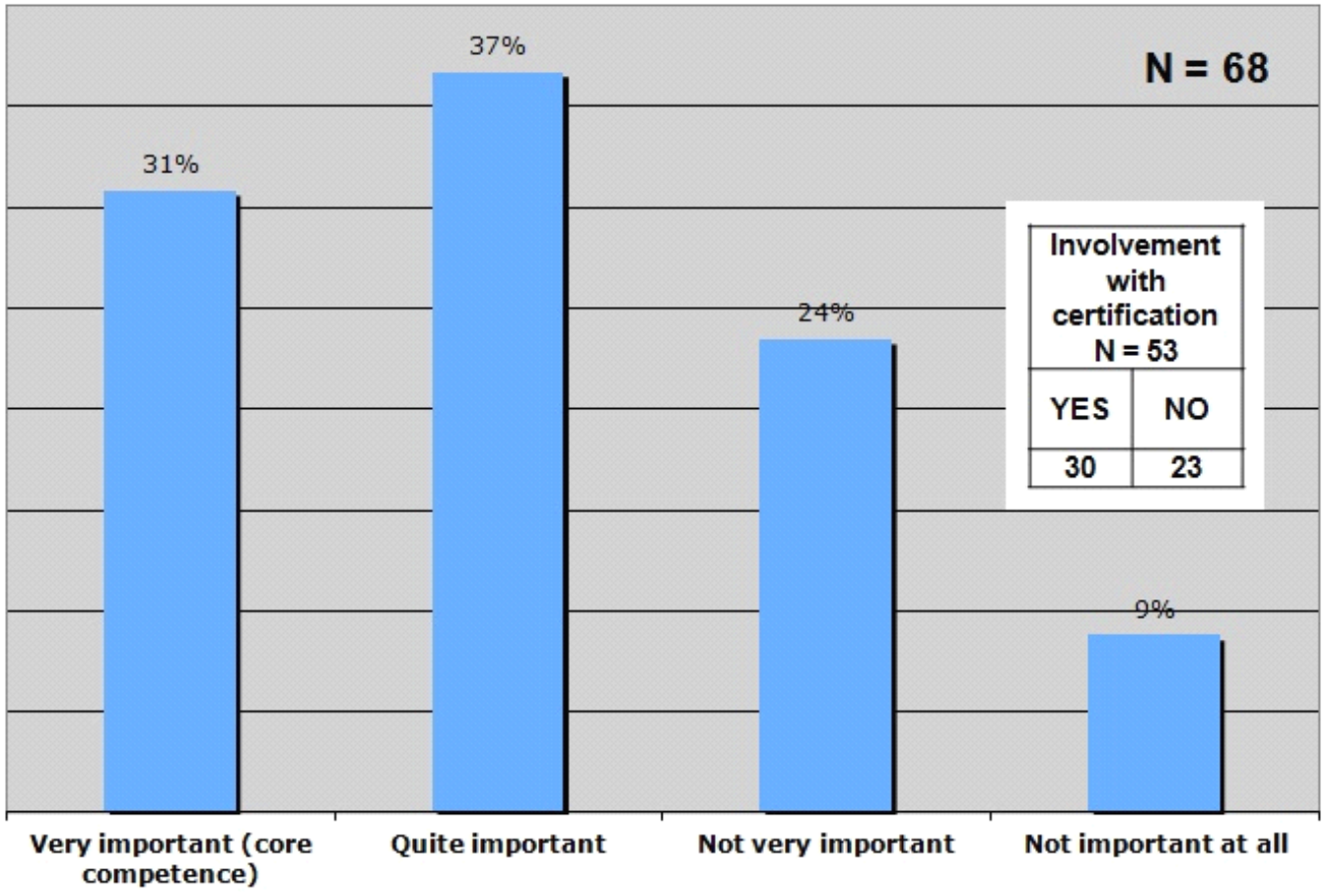


About 41% of the respondents indicate that organisation has been certified. About half for ISO-9001 and CMMI

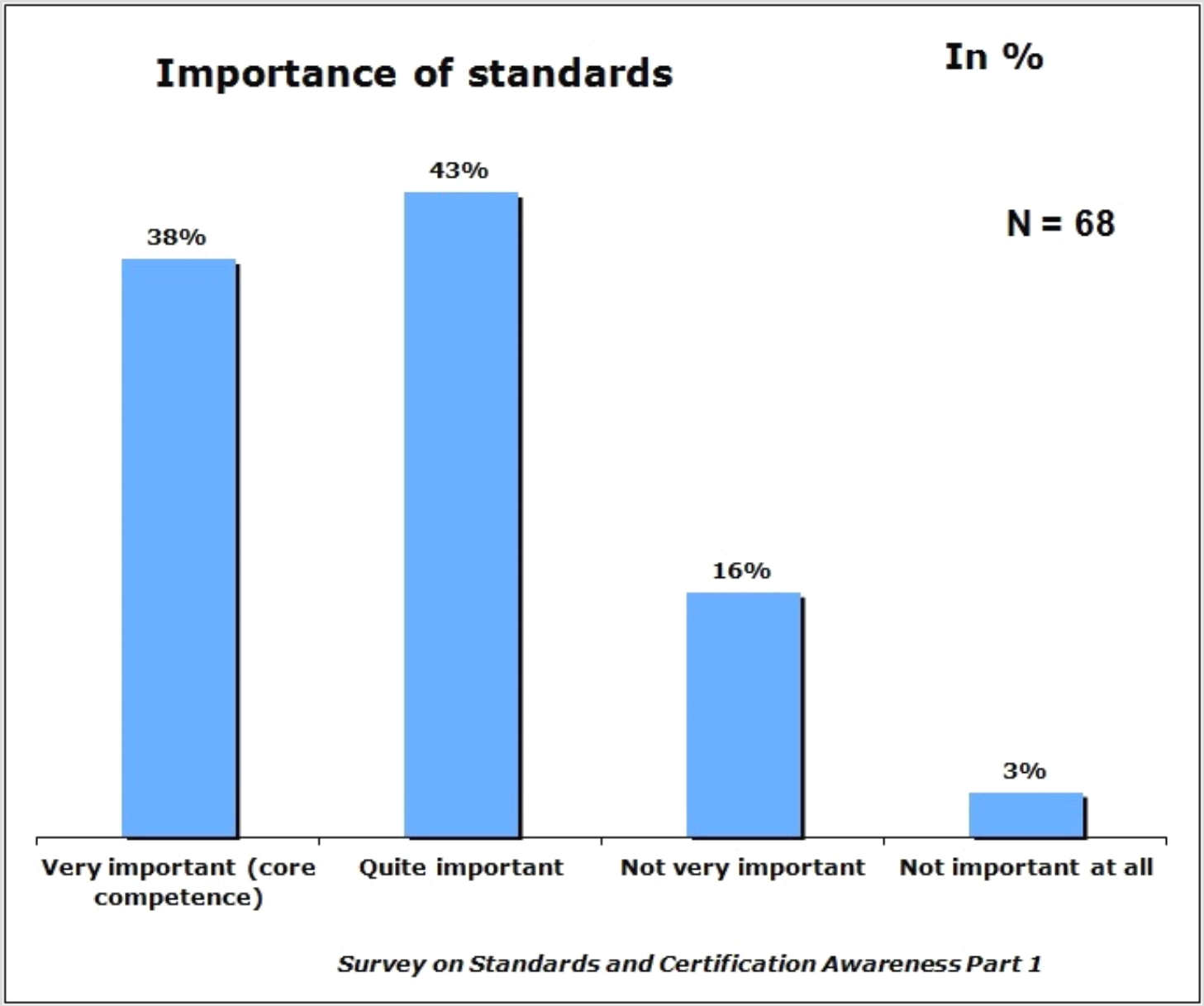

Application domains in %
N = 81
40%
35%
31%
30%
22%
20%
17%
15%
15%
14%
14%
5%
4%
IT
Automotive
Avionics
Public Transport
Space
Defence
Telecom
Infrastructure
Medical
Production
Health
Consumer Products
Academic
Survey on Standards and Certification Awareness Part 1

Importance of certification

In %



Survey on Standards and Certification Awareness Part 1


Importance of standards
In %
N = 68
38%
43%
16%
3%
Very important (core competence)
Quite important
Not very important
Not important at all
Survey on Standards and Certification Awareness Part 1

Standards: pick your own (total: 266, out of more)

EN ISO 10993, ISO 26262, D0-178,IEC 60118, ANY, IEC 60601, 21 CFR, EN ISO 389, IEC 61000, IEC 61508, EN 50126, EN 60601, ISO 9000, AUTOSAR, IEC 60068, EN ISO 14644, IEC 60068, SYSML, CENELEC, EN 45502, EN 50128, EN 868, UML, CMMI, DVB, EN 10088, EN 17025, EN 50129, EN ISO 11138, IEC 62304, ISO 8253, ISO 965, 93/42/EC, EN 17020, EN 556, EN ISO 11137, EN ISO 11607, EN ISO 12100, EN ISO 14155, EN ISO 14698, ETSI EN 301 489, IEC 60086, IEC 61511, IEEE 1149.1, ISO 13485, ISO 14971, ISO 15223, ISO 2768, ISO 9126, UNISIG, W3C, 2000/70/EC, 2004/108/EC, 2006/42/EC, 2007/47/EC, 89/392/EC, 90/385/EC, 95/46/EC, AADL, AAMI HE74, AAMI HE75, AAMI ST241, AAMI ST81, AAMI TIR 32, AAMI TIR 36, AAMI TIR18, AAMI TIR40, AAMI TIR80002, ABET, Ada, AEC-Q100, ANSI S3.21, ANSI S3.22, AQAP, ARINC 629, AS 9100, ASN.1, ASQ D1160, ASQ Z1.4, ASQ Z1.9, ASTM B265, ASTM D3078, ASTM D4169, ASTM F 1140, ASTM F 1608, ASTM F 17, ASTM F 1886, ASTM F 1929, ASTM F 1980, ASTM F 2052, ASTM F 2054, ASTM F 2095, ASTM F 2096, ASTM F 2097, ASTM F 2119, ASTM F 2182, ASTM F 2213, ASTM F 2503, ASTM F 2504, ASTM F 382, ASTM F 67, ASTM F 88, ASTM F136, ATEX, CAN, CEPT/ERC Report 25, CISPR 11, CISPR 14, CISPR 15, CR CCS TSI, DICOM, DIN, DO-254, EASA, ECSS, ECTS, EN 1040, EN 1041, EN 1080, EN 12353, EN 1275, EN 1276, EN 13060, EN 13427, EN 13428, EN 13429, EN 13430, EN 13697, EN 13824, EN 1422, EN 14682, EN 1499, EN 1500, EN 15038, EN 1650, EN 1656, EN 1657, EN 285, EN 475, EN 50155, EN 60825, EN 60850, EN 62304, EN 62366, EN 867, EN 980, EN ISO 11135, EN ISO 11140, EN ISO 13485, EN ISO 14121, EN ISO 14161, EN ISO 14937, EN ISO 14971, EN ISO 15882, EN ISO 16061, EN ISO 17664, EN ISO 17665, ESD S20.20, ETSI EN 300 330, ETSI TS 105175, FFFIS, FlexRay, HL7, HS CCS TSI, I2C, IEC 12207, IEC 60318, IEC 60512, IEC 60645, IEC 60812, IEC 60950, IEC 61131, IEC 61158, IEC 61438, IEC 61499, IEC 61951, IEC 61959, IEC 61960, IEC 62061, IEC 62133, IEC 62281, IEC 60384, IEC 60529, IEC 60884, IEC 61058, IEC/TR 62354, IEC61508, IEEE 1074, IEEE 11073, IEEE 12207, IEEE 1588, IEEE 1625, IEEE 1725, IETF, IHE, IPC 7351, IPC-A-600, IPC-STD-001E, IPC-STD-033, IPV6, IRIS, ISA 100, ISO 10012, ISO 11137, ISO 13606, ISO 13715, ISO 13781, ISO 13940, ISO 14000, ISO 15225, ISO 15504, ISO 16022, ISO 22600, ISO 250xx, ISO 26000, ISO 2700, ISO 2859, ISO 5832, ISO 5834, ISO 5838, ISO 60118, ISO 6474, ITIL, ITU-R 27, Java RT, JIS C 8711, LN, MDD, MED DEV 2.7.1, MED DEV 2.7.2, MIL B 49030, MIL standards (all), MIL-M 38510, MIL-PRF 38534, MIL-PRF 38535, MIL-PRF-49471B, MIL-STD-883H, MISRA-C, MOF, MOST1000, MPEG, NATO Stanags, NE 2575, NEN 1010, NEN 2575, NEN 2654, OCL, Pascal (ISO), Privacy, Python, QVT, RFID, Safety, SAFETY SIL, SDL, SDL-RT, SOA, SOX, SQUARE, Subset026, TTCN-3, UIC 544-1 , UL 2054, USB, WirelessHART, Zigbee,

Importance and obstacles for standards

- Importance:
  - It is a legal or market requirement 31.1%
  - It enhances quality, reduces cost and increase lifetime of products/systems developed. 32.8%
  - Interoperability 36.1%
- Obstacles:
  - Standards are complex, difficult, costly, change constantly, vague and difficult to obtain 60.3%
  - Acceptance by the organisation or culture is lacking 22.4%
  - Proprietary solutions work better, standards lag technology and hamper innovation 17.2%

Demotivating factors for certification

- Effort, cost, complexity, inconsistency, bureaucratic (paperwork) 60.7%
- Change management (evolving standards, evolving products), differences national/ international 21.4%
- Rigidity, lagging market and technology 17.9%

Conclusion from the in-depth interviews

- Certification and safety engineering is complex
- The safety domain is still in an early phase (diversity of practices and safety standards across the different domains)
- The maturity is greatest for avionics, followed by railway whereas automotive still in an emerging phase.
- From paper-driven waterfall to agile, lean processes.
  - Word, excel, etc. dominate
  - Lots of human/manual work
- Lowering certification costs today:
  - use a better process, not so much more tools
  - Lean/agile: -30% cost, integration from 12m to 3w

Safety as a goal = "Safety Integrity Levels"

Domain					
General (IEC-61508) Programmable elecronics	SIL0	SIL1	SIL2	SIL3	SIL4
Automotive (26262)	ASIL-A	ASIL-B	ASIL-C	ASIL-D	-
Avionics (DO-178/254)	DAL-E	DAL-D	DAL-C	DAL-B	DAL-A
Railway (CENELEC 50126/128/129)	SIL0	SIL1	SIL2	SIL3	SIL4

SIL targets risk reduction

SIL	PFD (Probability of Failure per Hour)	PFD (power)	RRF (Risk Reduction Factor)
1	0.1-0.01	10^{-1} - 10^{-2}	10-100
2	0.01-0.001	10^{-2} - 10^{-3}	100-1000
3	0.001-0.0001	10^{-3} - 10^{-4}	1000-10,000
4	0.0001-0.00001	10^{-4} - 10^{-5}	10,000-100,000

For continuous operation, these change to the following.

SIL	PFH	PFH (power)	RRF
1	0.00001-0.000001	10^{-5} - 10^{-6}	100,000-1,000,000
2	0.000001-0.0000001	10^{-6} - 10^{-7}	1,000,000-10,000,000
3	0.0000001-0.00000001	10^{-7} - 10^{-8}	10,000,000-100,000,000
4	0.00000001-0.000000001	10^{-8} - 10^{-9}	100,000,000-1,000,000,000

Note: risk reduction depends on domain!

Problems with SIL definition

- Poor harmonization of definition across the different standards bodies which utilize SIL
- Process-oriented metrics for derivation of SIL
- Estimation of SIL based on reliability estimates
- System complexity, particularly in software systems, makes SIL estimation difficult if not impossible
- => based on probabilities that are very hard if not impossible to measure and estimate
- Risk figures are different for each domain => reuse?
- The law of Murphy still applies:
  - The next instant can be catastrophic

Problems with SIL as a design goal

- Safety goal is put first => architecture is derived
- Clearly effort in standards to minimise effort and costs => complex and hard to use tables
- First principles in safety engineering:
  - Safety culture: is also keep it Simple and Smart
  - Quality is a pre-condition
    - Traceability
    - Configuration management

How to achieve safety?

- It's a requirement
- What are the derived specifications?
- How to derive these safety specifications?
- How to fulfill these safety specifications?
  - Develop a solution that is safe by design
  - Follow a process that leads to safe(r) products

The real issue with SIL

- SIL is a system level concept
  - But we design using components and reuse
- SIL cannot be reused
  - But components can!
- SIL is domain specific
  - Components are domain independent
- Composibility still unclear issue. Why?
- => we must start at the component level

Safety composibility

- Although this is the practice in systems engineering, it is poorly addressed in the standards
- Example from ISO-DIS-25119 (derived ISO 13849)
  - "The safety-related parts of control systems shall be designed in accordance with the requirements of one or more of the 5 categories specified in ISO-DIS-25119-2:2008-Annex A. When a safety function is realized by an integrated combination of multiple hardware categories, the resulting safety function AgPL is limited by the overall hardware category, including MTTFdc, DC, SRL, CCF, etc."
  - If the embedded software has to implement software components with different SRLs or safety-related and non-safety related software components, then the overall SRL is limited to the component with the lowest SRL, unless adequate independence between the software components can be demonstrated

Redefining SIL to be domain independent ?

- **S** = Safety
  - (includes security)
- **I**ntegrity **L**evel
- Attempt at one SIL definition for all domains
- **If we would follow the standards in spirit**

SIL 0	no impact
SIL 1	material damage
SIL2	harmful impact on people
SIL3	one person dead
SIL4	many persons dead

What's wrong with this definition?

- Even a perfectly designed and proven system composed of perfect components can fail catastrophically
  - Concorde: 100% safe until the first one crashed due to an improbable external cause
- The law of Murphy has priority over probability over a life time
- SIL is a lifetime indicator for a system, not a criterium for selecting components

Why is ASIL-D ≠ SIL4 ?

- More or less: ASIL-C = SIL3
  - Switch to fail-safe mode
- More or less: ASIL-D = SIL 3.5
  - Supervised operation
  - Whereas SIL4 implies fault tolerance
- According to generic SIL table =>
  - car has upto 5 passengers => SIL 3.5
  - but 35000 people get killed per year (EU) => SIL4
    - System is not the car but car based transport
    - Car is component in the larger system

New definition: we start from the component

- **ARRL: Assured Reliability and Resilience Level**

ARRL 0	it might work (use as is)
ARRL 1	works as tested, but no guarantee
ARRL 2	works correctly, IF no fault occurs, guaranteed no errors in implementation) => formal evidence
ARRL 3	ARRL 2 + goes to fail-safe or reduced operational mode upon fault (requires monitoring + redundancy) - fault behavior is predictable as well as next state
ARRL 4	ARRL 3 + tolerates one major failure and is fault tolerant (fault behavior is predictable and transparent for the external world)

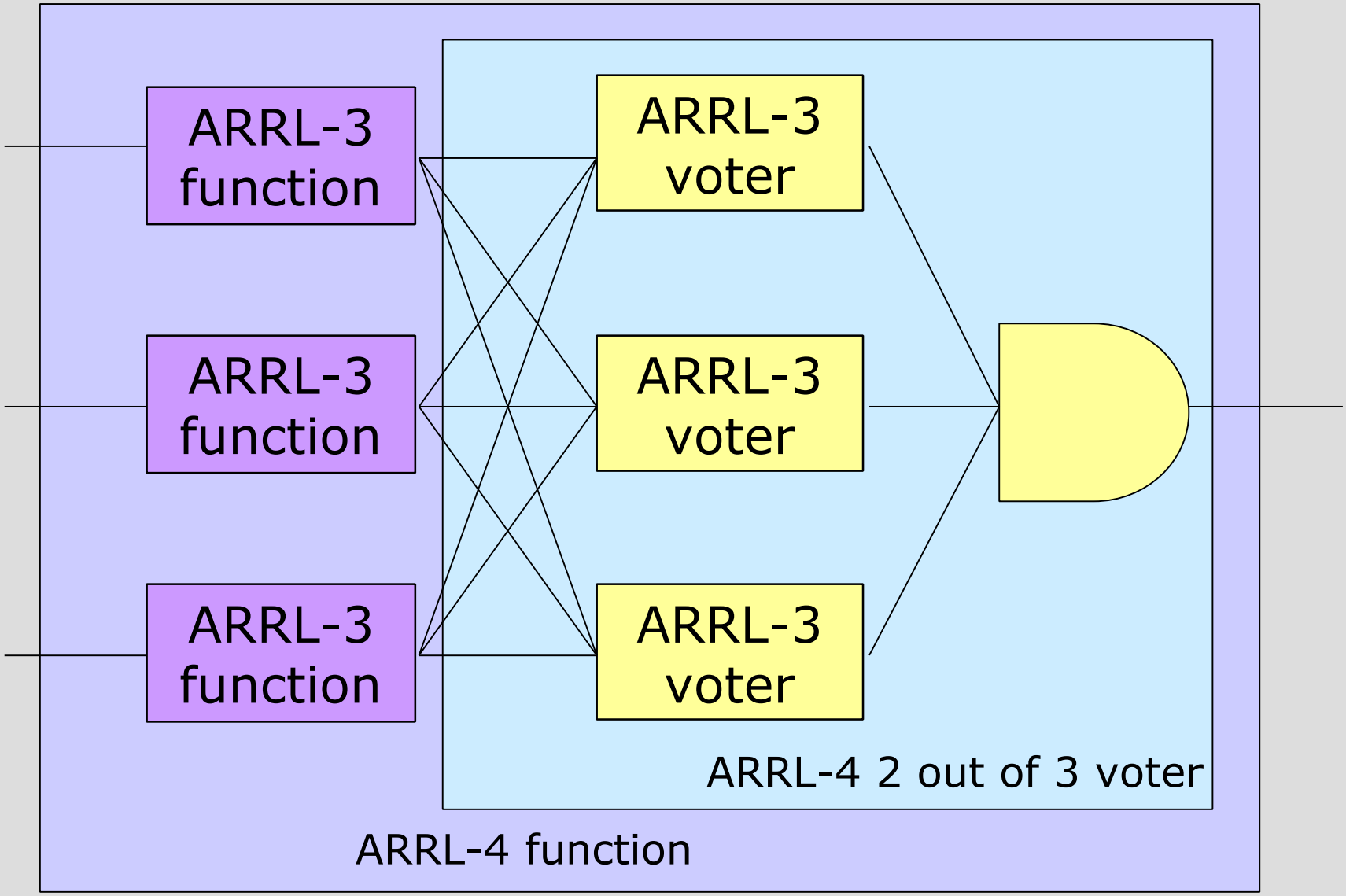
Consequences

- If a system/component has a fault, it drops into a degraded mode => lower ARRL
  - ARRL3 is the operational mode after an ARRL4 failure
    - Functionality is preserved
    - Assurance level is lowered
- SIL not affected and domain independent
  - System + environment + operator defines SIL

Composition rule:

- A system can only reach a certain SIL level if all it components are at least of the same ARRL level.
  - This is a pre-condition, not a sufficient condition
  - Redundancy can compose ARRL 4 components out of ARRL 3 components (needs an ARRL 4 voter)
  - ARRL3 component can use ARRL 2 components (>2)
- Consequences:
  - Interfaces and interactions also need ARRL level!
  - Error propagation is to be prevented => partitioning architecture (e.g. distributed, concurrent)

Generic example



Common mode failures => ARRL-5

- ARRL-4 assumes independence of faults in each redundant channels
- Covers only a subset of the common mode failures
- Less visible are e.g. common misunderstanding of requirements, translation tool errors, time dependent faults => require asynchronous operation and diversity/heterogenous solutions
- Hence we can define an ARRL-5 as well

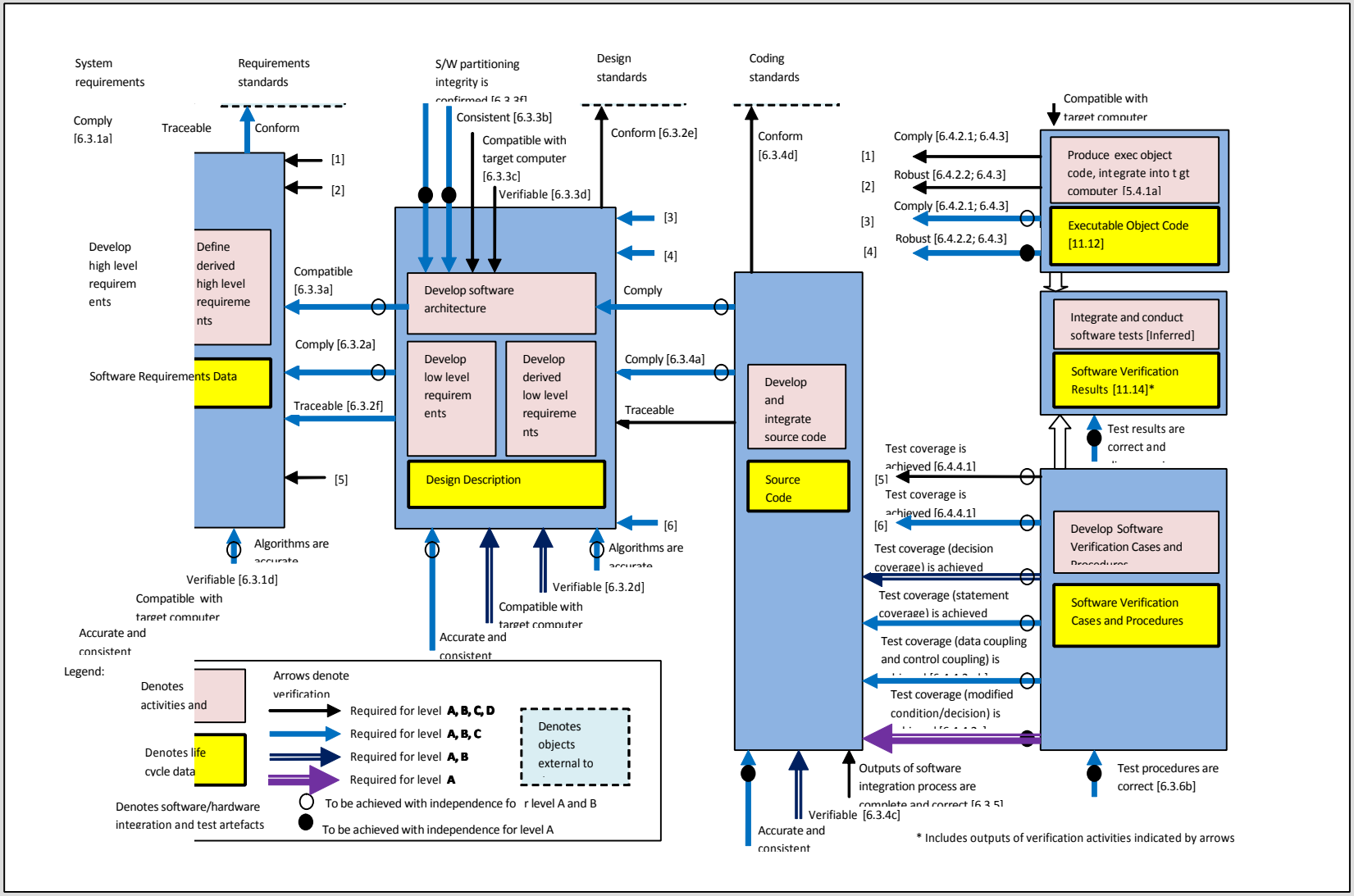
Dependency analysis

- SIL is a top level Requirement, decomposable in
  - Normal Case requirements (ARRP 1 &2)
  - Fault Case Requirements (ARRL 3 & 4 or even 5)
- Requirements become Specified properties to be met and verified by implementation
  - Each ARRP level has same "normal case" architecture but different "ARRP" level architecture and hence also different non-functional properties safety (measures have a resource cost)

What about a unified SE process?

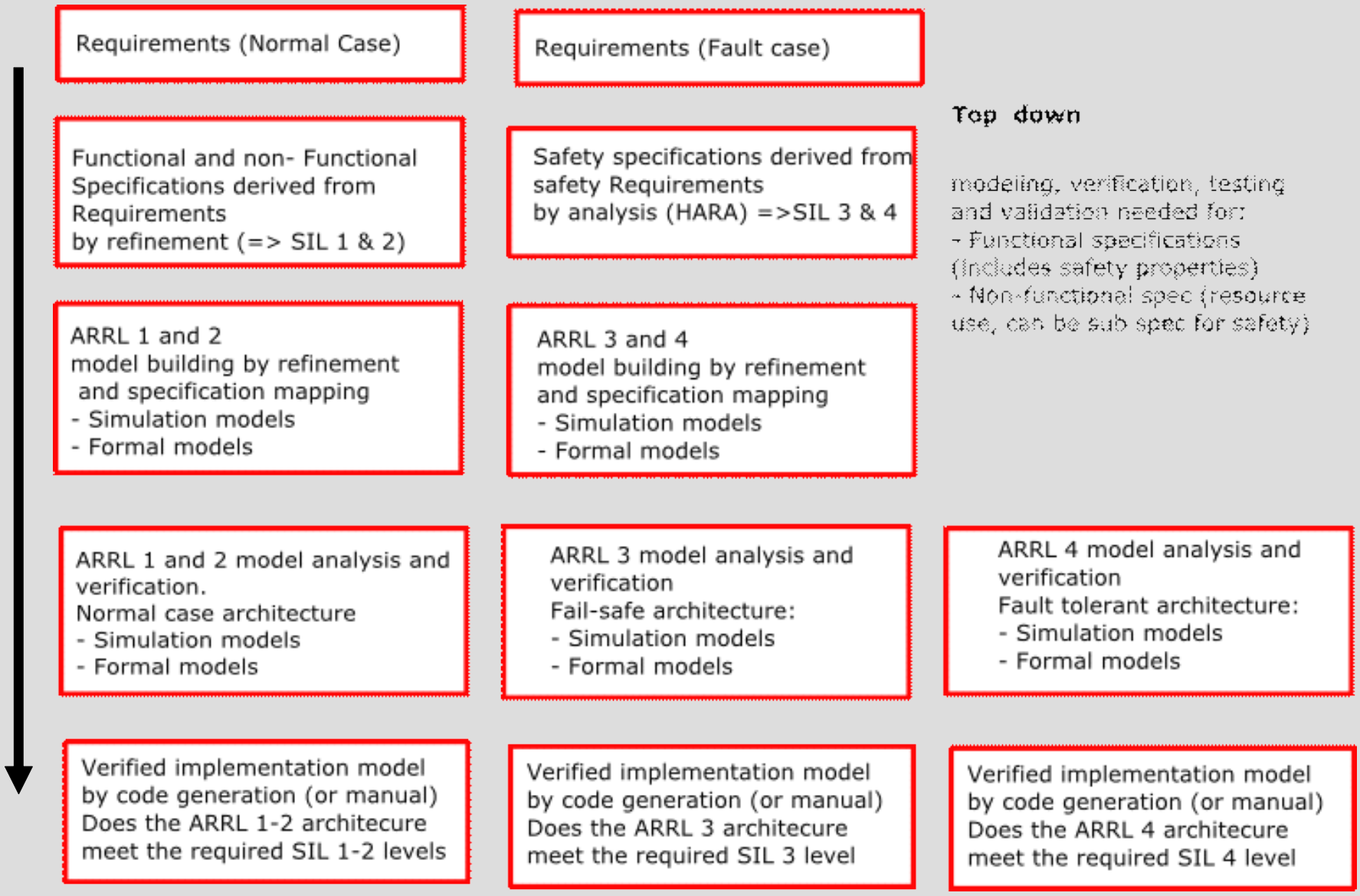
- Dependency chain + iteration
- Requirements => Specifications
  - By decomposition and refinements ("testable")
- Input for Work Package
  - also needs Resources
  - Composed of Development, Verification, Test and Validation/integration Tasks
  - Produces Work Products:
    - Process artefacts (evidence, ...)
    - Entities and Models (incl. implementation)

Current practice (example from DO-178B - SW)

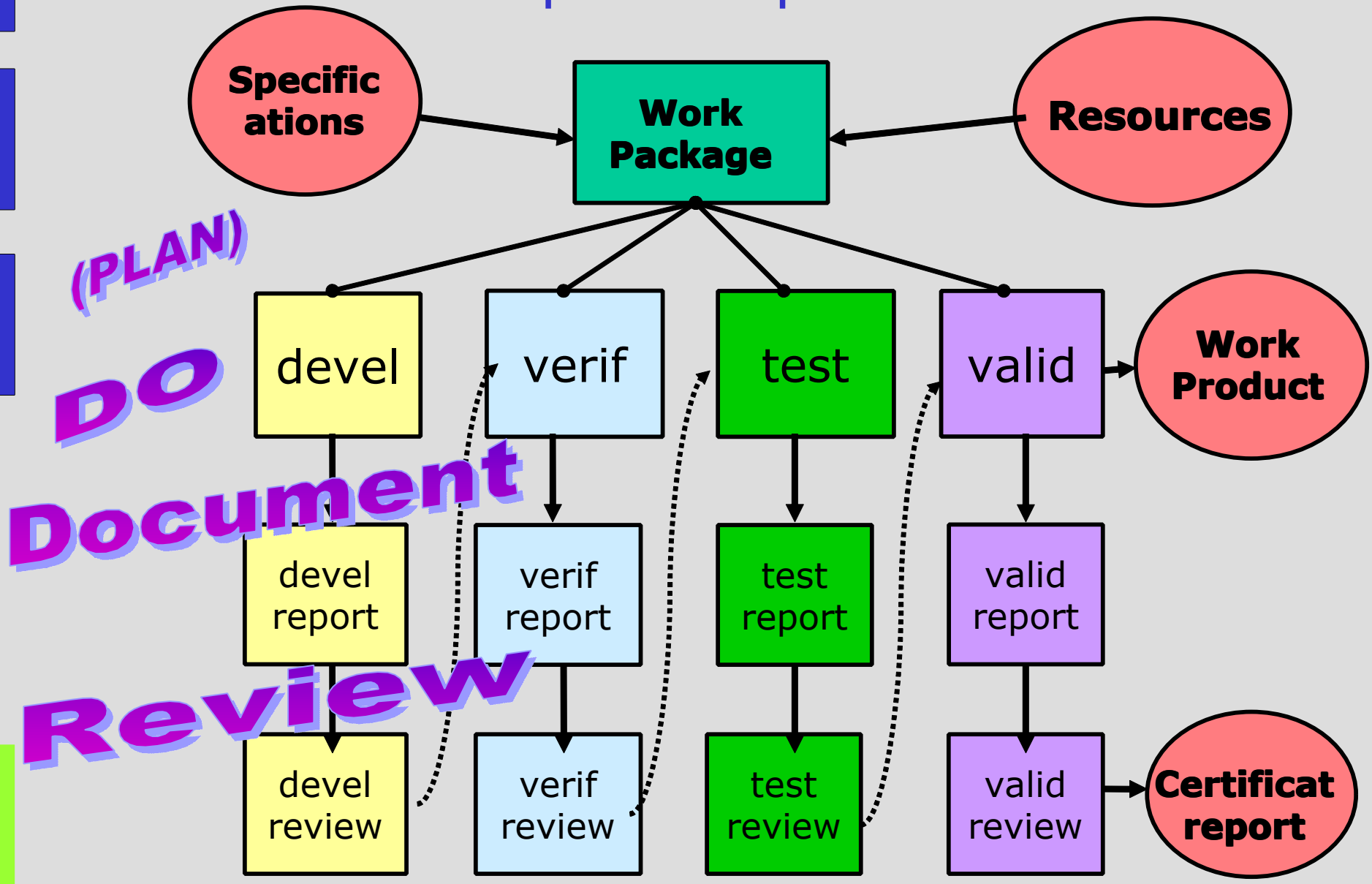


RTCA-DO/178B Software Development and Verification Processes

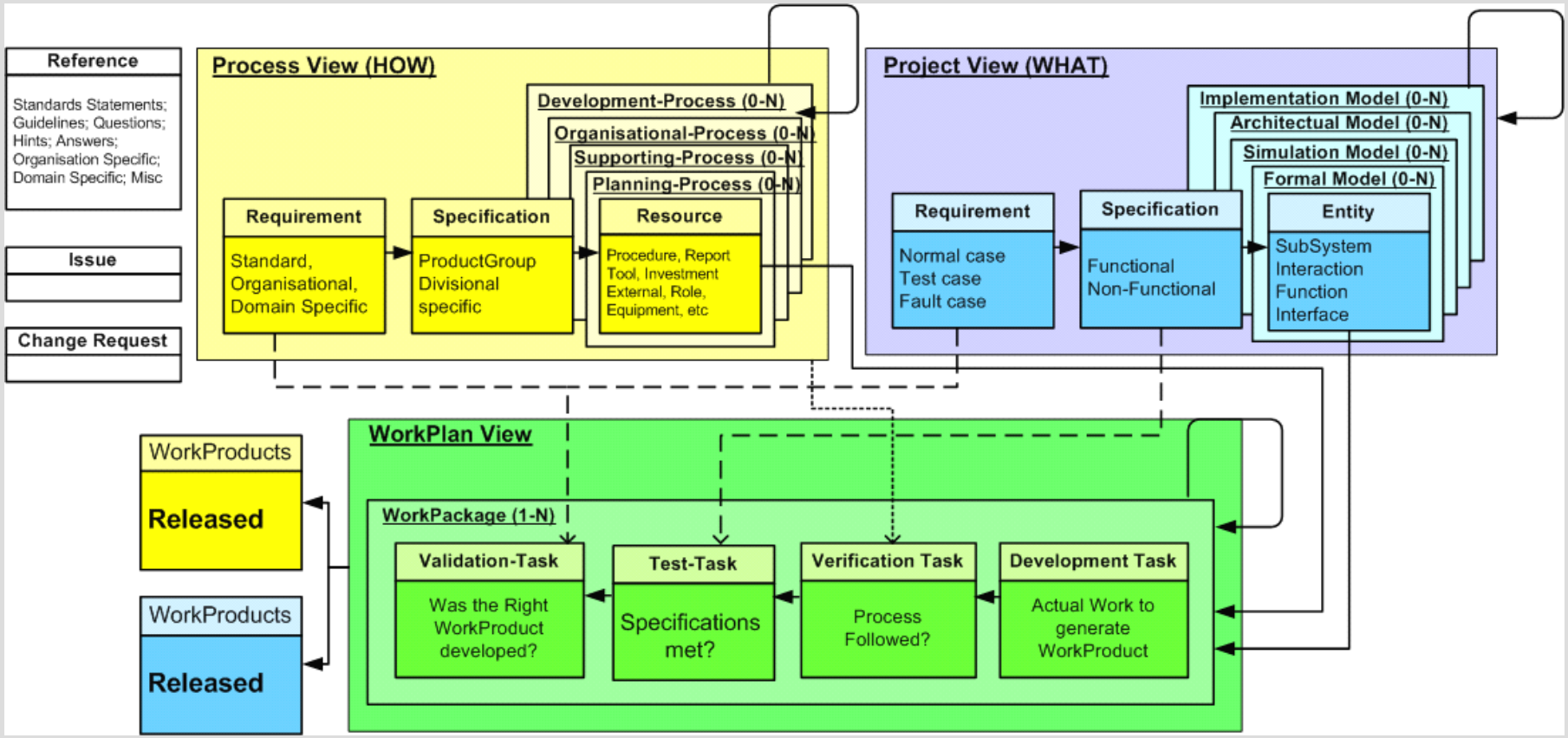
ARRL process flow



What we need: a process pattern for reuse



Interplay of view (as in GoedelWorks)



Conclusions

- Unified system and safety engineering is feasible
- Unified safety certification is not yet feasible (standards and SIL differ too much)
- ARRL concept allows composable safety engineering with reuse of components
- A unified process pattern can unify systems and safety engineering standards:

DO - DOCUMENT - REVIEW

More info:

www.altreonic.com/content/altreonic-releases-new-approach - systems-engineering-goedelworks