



**Embedded  
France**

L'embarqué made in France

# Normes pour la Sûreté de fonctionnement Logiciel et système - Groupe de travail NSL Embedded France

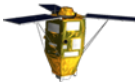
ETR 2017

Paris - 30 août 2017



# Safety Standards Working Group

→ Cross-domain analysis and comparison of safety standards

→ Better standards, practices, synergy...



Jean-Paul Blanquart, Airbus Defence and Space  
Philippe Baufreton, SAFRAN Electronics & Defense  
Jean-Louis Boulanger, CERTIFER  
Jean-Louis Camus, ANSYS ESTEREL  
Cyrille Comar, AdaCore  
Hervé Delseny, Airbus  
Jean Gassino, IRSN  
Abderrahmane Kettany (Alstom Transport)  
Emmanuel Ledinot, Dassault Aviation  
Philippe Quéré, Renault  
Bertrand Ricque, SAFRAN Electronics & Defense

Aeronautics		ARP 4754, 4761 DO 178, 254 
Processus industriels		IEC 61508, 61511
Automobile		ISO 26262 
Nucléaire		IEC 60880, 62138
Ferroviaire		EN CENELEC 50126, 8, 9, 50155, 50159-1, 50159-2
Espace		ECSS Q30, Q40, Q80
Editeurs de logiciel		
Medical		

1. Notion of categories for safety - An overview
2. System level considerations
3. Software level considerations

# 1. Notion of categories for safety

## An overview

DAL – ASIL (SIL / SSIL)

➔ It is possible to get very trustable systems (elements, ...)

- › Thanks to agreed means of assurance

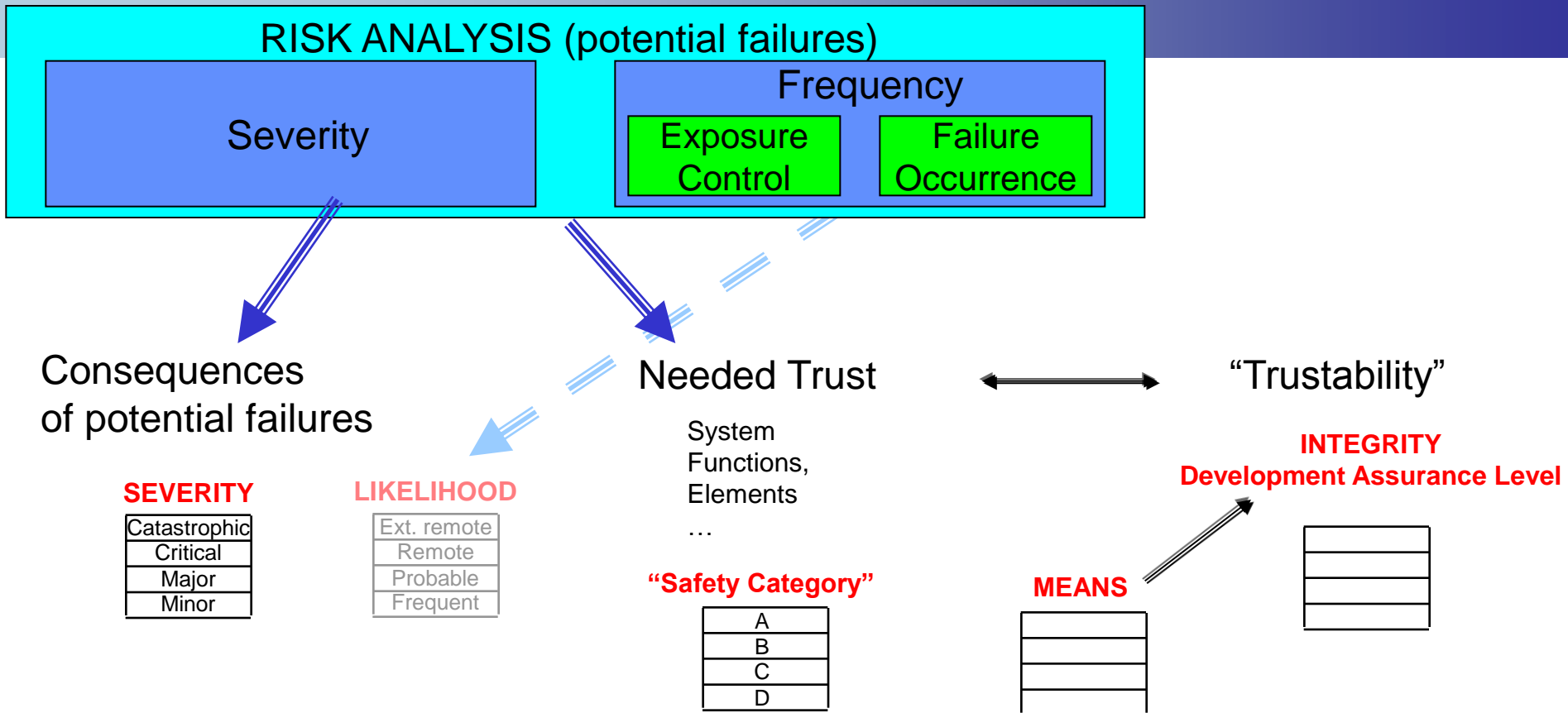
➔ Limitations:

- › Technical feasibility
- › Cost

➔ Basic principles:

- › Identify “how much trust is needed”
- › Define accordingly the appropriate assurance means
- › Seek for a generic approach
  - Not everything is addressed
  - What is addressed is based on a rationale established only once

# The approach at a glance



The "safety category"  
Is related to the severity category of the most severe consequences of potential failures...

... so as to meet the required level of safety and dependability thanks to development and validation means appropriate with respect to the identified safety category

## → Aeronautics

- › “Failure Conditions”

## → Automotive

- › Vehicle level hazards

## → Nuclear

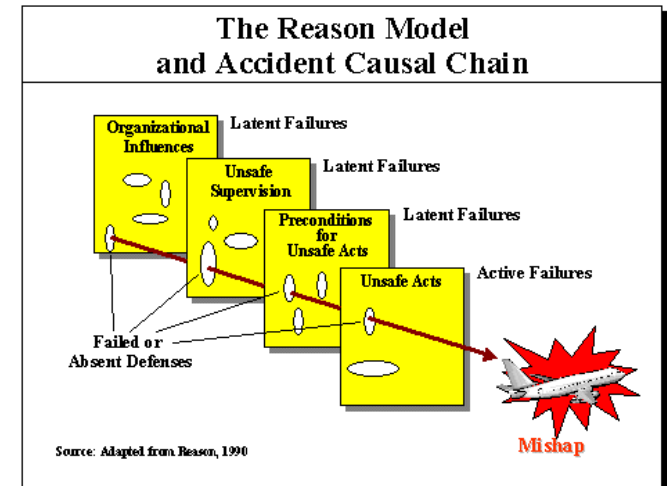
- › Accident conditions, considering reactor type, plant states

## → Railway

- › Hazard (dangerous event and capability to develop as an accident)

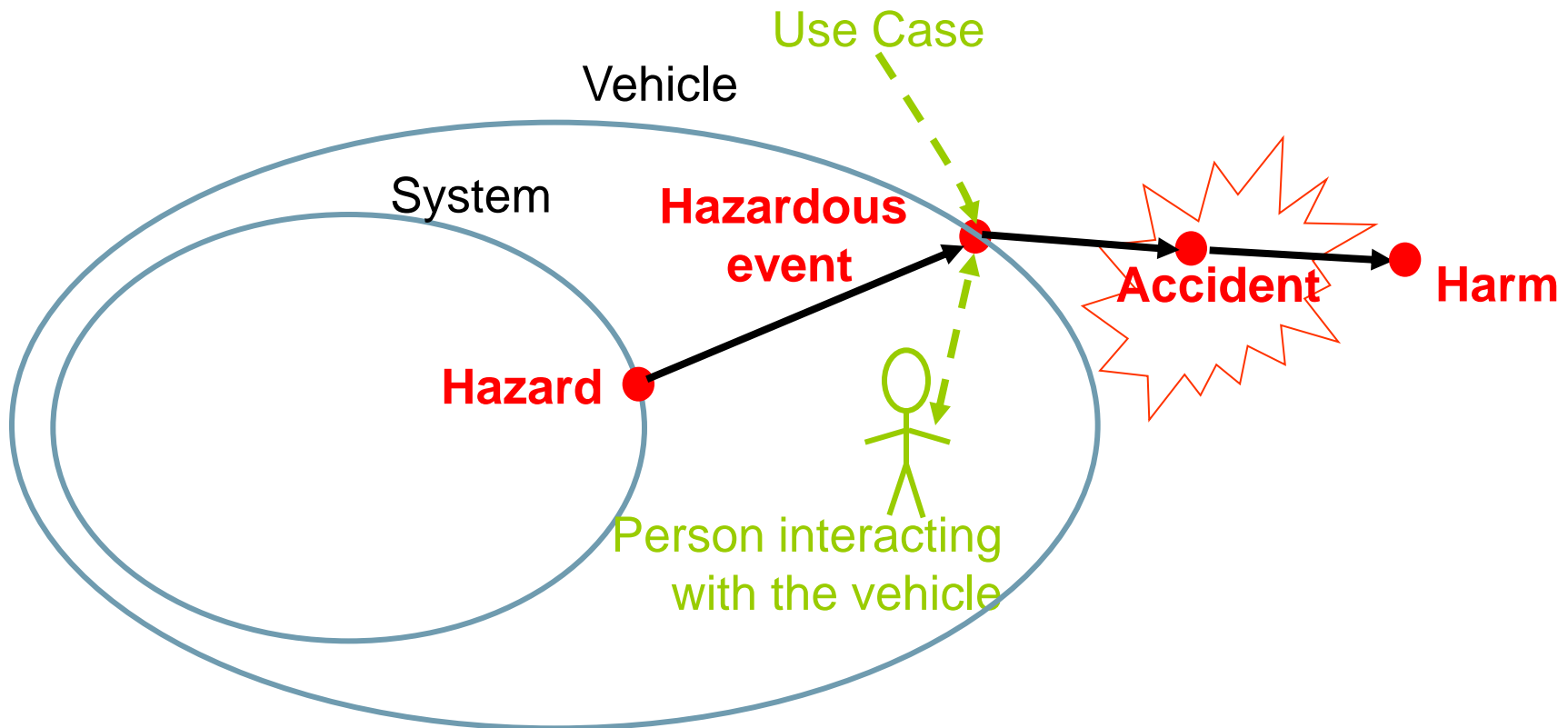
## → Space

- › End-effect consequences of potential failures

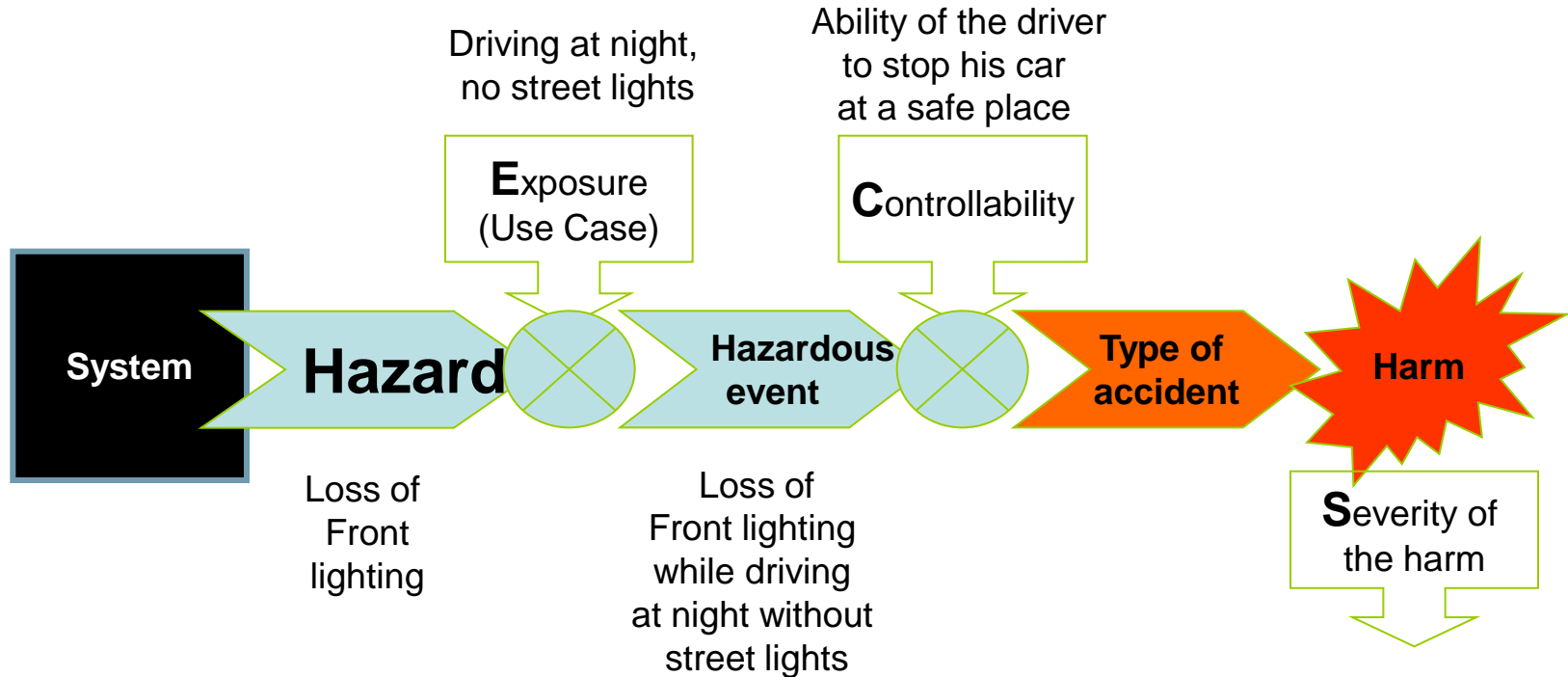




## ASIL: characterizes a Hazard



**Hazard**: system failure mode or unintended behaviour that may lead to harm



**ASIL:**

- "criticality" of a Hazard =  $F(E, C, S)$   
(given the hazard, how [severe x frequent] will be the effect)
- level of "development effort" required for a Hazard

# First categorised element and allocation along design

## → Aeronautics

- › Functions (FDAL) → Items (IDAL)

## → Automotive

- › Safety Goal → Safety requirements and “architectural components”

## → Nuclear

- › Nuclear facility (functions) → Systems, sub-systems

## → Railway

- › Function → Systems, sub-systems

## → Space

- › Function → “products” implementing them

## → Guidance provided

- › Aeronautics (only once, at “system” level)
- › Automotive (safety requirements, any level)
- › Independency to validate at initial category

## → No guidance provided, general rule applies

- › Railway
- › Space

## → Not considered

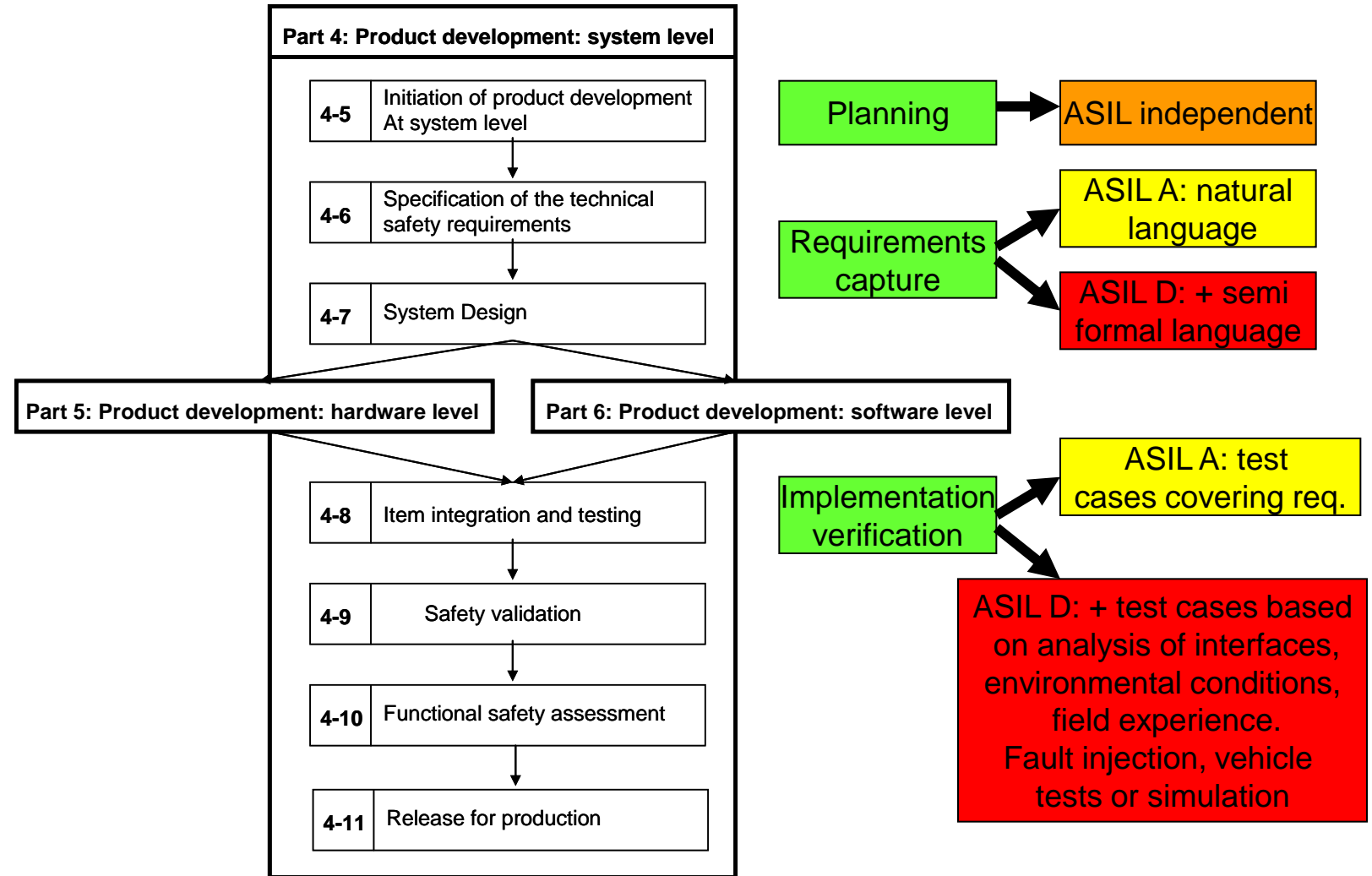
- › Nuclear

- ➔ A very similar global approach
  - › Applied with more or less specialisation to a domain
    - E.g., categories of severity, first categorised element
- ➔ Some variations
  - › “Presentation” issues (likelihood)
  - › But not only
    - Exposure
    - Granularity of the allocation
    - Dependability architecture
- ➔ Must be completed
  - › Cross-domain comparison of assurance means

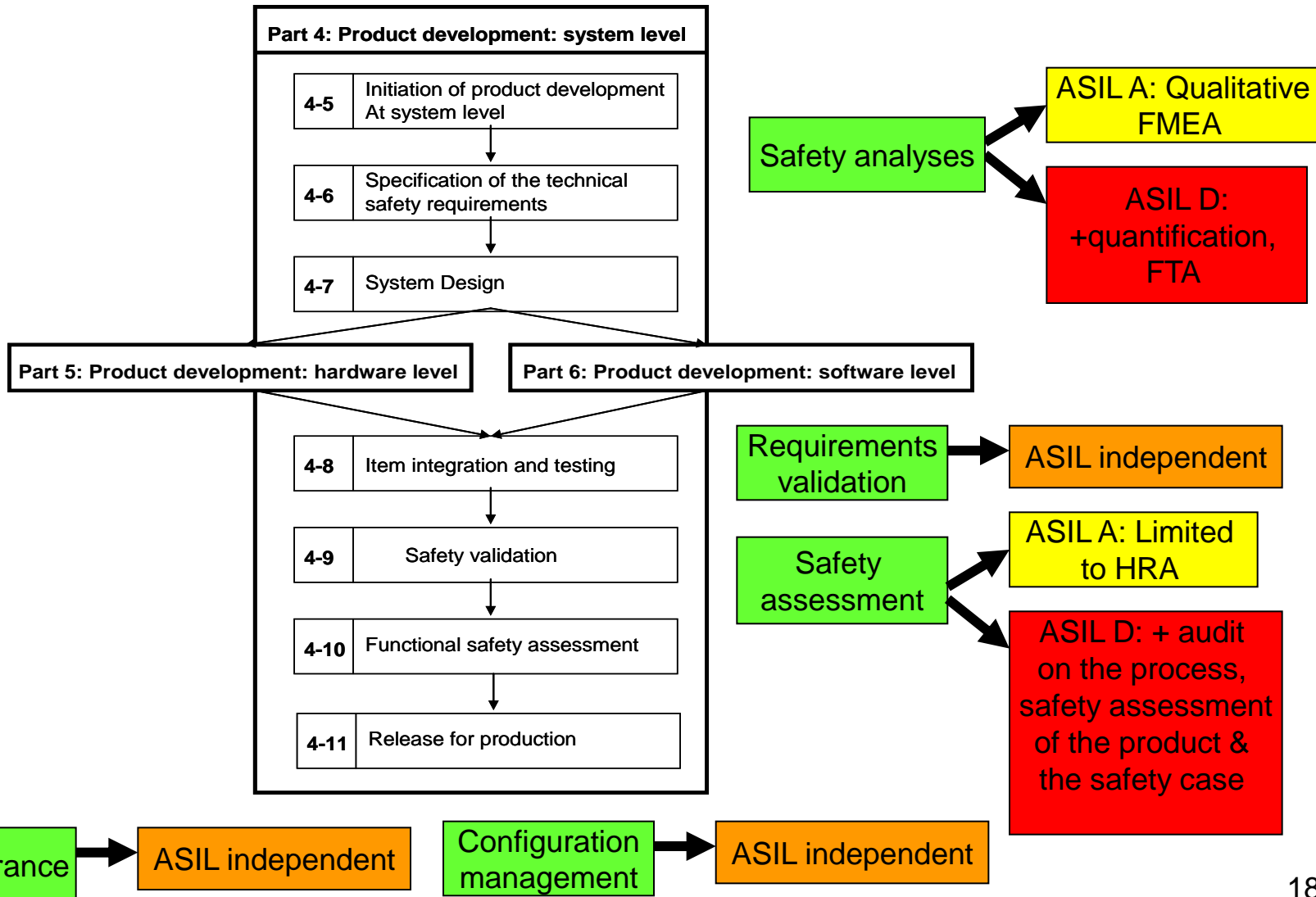
## 2. System level considerations

- ➔ Standards ED-79A/ARP-4754A (Systems) and ED-135/ARP-4761 (Safety)
- ➔ 5-level scale based on severity of consequences of failures
  - › System, functions
  - › Products implementing the functions: architecture, hardware, software
- ➔ Product safety requirements
  - › Functional safety requirement expressed in quantitative terms
  - › Minimum number of independent faults per category of consequences
- ➔ Process safety requirements
  - › A development assurance level for the system which implements the function.
  - › Qualitative and quantitative analysis, verification

➔ Reference phased model for the development of a safety related system (ISO 26262).







## → Commonalities

- › Every domain defines requirements for systems that are applicable to each step of the whole lifecycle.
- › Each domain attributes a “safety level” (SIL, ASIL, DAL ...) to each system, based at least on the worst case consequences (e.g. severity) of its potential failure.
- › Those safety levels imply requirements applicable to the system itself (e.g. architecture) and to its development process.
- › To cope with residual design errors, fault tolerance is introduced at system level for the highest safety levels.
- › Emphasize sufficient independence (e.g. isolation) against common cause failure.

## → Dissimilarities

- › Difference in the nature of the requirements prescribed by the standards (objectives vs means).
- › Modulation of rigour / activities is domain dependent.
- › When a safe state can be reached, safety systems may be segregated from control command systems.

- All standards studied in this group have a common view of the fundamental principles (Integrity level) but have many variations due to specificities of each application domain.
- Opportunities of product reuse from domain to domain are not expected at system level on a large scale.
- Conversely, the use of common tools is foreseeable e.g. safety analysis.

## 3. Software level considerations

# Software Development Assurance

## A few Questions to be Answered

- ➔ Does the DAL-dependent progressive construction of software safety rely on the same principles ?
- ➔ Could the highest development assurance levels of :
  - › ED-12/DO-178,
  - › IEC 61508,
  - › ISO 26262
  - › IEC 60880
  - › EN 50128
  - › ECSS-Q-ST-80C,be claimed equivalent ? If so, on what basis ?

- ➔ Analysis of the quantified system level safety objective the highest DAL is claimed to be "*compatible with*" ...
  - › Are the standards aiming at the same rareness of residual systematic faults ?
  
- ➔ Analysis of the DAL-dependent requirements
  - › What is incremental rigor ?
  - › on supporting processes
  - › on development processes
  - › on verification processes

- Process-based software development assurance standard
- 5 DALs : from E (no requirement) to A (most critical "~"  $\lambda = 10^{-9}/h$ )
- The DALs drive sets of activities & quality objectives, verification independence, lists of work products under CM control, test coverage, ..  
**Quality** : Compliance, consistency, accuracy, completeness, HW compatibility,..
- Revision C : no change in the core principles
  - › the 3 technical supplements (OO, MBDV, FM) are *not* mean prescriptive
  - › few quality objectives added
- Context dependency of SW qualification but no focus on application dependent safety requirements
- Significant influence of the DALs on supporting processes (planning, CM,..)
- From D to A : compliance of executable object code with HLRs
- Effort required : A ~ B >> C >> D



- ➔ Specific : the DALs are allocated to safety requirements
  - › ... leads to the possibility of ASIL mix within a ECU
- ➔ 4 levels : from ASIL A to ASIL D (most critical)
- ➔ ASIL-dependent variability applies to the methods and tools
  - › the work products to be supplied for confirmation measures are the same from A to D
  - › ASILs modulate the level of formality\* for requirement capture & software verification, for applicable coding standards,
  - › ASILs modulate testing methods and testing coverage criteria
- ➔ ASIL-dependent variability also applies to **product-based** requirements :
  - › error detection and handling mechanisms, isolation mechanisms, etc.

- ➔ The different software assurance standards modulate :
- › quality objectives (product & process)
  - › process activities,
  - › lifecycle work products,
  - › means : methods, tools, rules, patterns, software engineering standards, etc.
  - › the independence of some verification activities
    - on software
    - on process conformity to the standard (e.g process quality assurance)
  - › the content of the software product
    - defensive programming, error detection mechanisms, self supervision

- ➔ Focus on safety related requirements .vs. focus on software as a whole
- ➔ DAL-independent process with DAL- dependent means .vs. DAL- dependent process with DAL- independent means
- ➔ Space : meta-level nature of the standard
- ➔ Variability of the criticality grain :
  - › at safety requirement level for Automotive
  - › grain = ECU-software for the other domains
- ➔ Great variability on formal methods
  - › Railway highly recommends for SSIL 4
  - › Aeronautics does not recommend, even for level A

- ➔ "Compatibility principle" with quantified system level safety objectives
- ➔ All standards modulate on support processes (planning, CM, etc.)
- ➔ Structural coverage : uniformly used and uniformly DAL-dependent
- ➔ Verification independence : uniformly used at highest level, but :
  - › of the SW product and/or
  - › of the process conformance to the standard
- ➔ All the mean-oriented standards (Automotive, Automation, Nuclear, Railway) modulate :
  - › design and programming rules ("standards")
  - › methods of verification by analysis
  - › methods of testing
  - › testing environments

- ➔ Standards for software in safety-critical applications vary but exhibit commonalities.
  - › Typical standards comprise a number of *assurance requirements*, (*objectives* in DO-178C) most of which specify activities that developers must perform or qualities that development artefacts must have.
  - › Most standards for software in safety-critical applications are sometimes called *prescriptive* or *process-based*.
  - › Most standards' assurance requirements are only indirectly related to the property of interest (e.g., system safety).
    - No direct measure of software contributions to system safety (or even of software correctness); instead, each relates to assessable actions or properties such as having reviewed the software requirements or achieved specified structural test coverage.
- ➔ The issues of whether standards for software in safety-critical systems work and how to improve them have been a topic of discussion for many years.



	DAL-dependency		
	None	Medium	High
<b>Product/content</b>	Aero	Automation Automotive, Space Railway Nuclear	
<b>Process/ Quality objectives</b>		Automation Automotive, Railway	Aero Nuclear Space*
<b>Process/ activities</b>	Automation Automotive, Railway	Nuclear	Aero Space*
<b>Process/ means</b>	Aero	Nuclear Space*	Automation Automotive, Railway
<b>Process/ Independence/ V&amp;V</b>		Aero Automation Automotive Nuclear	Space Railway
<b>Process/ Independence/ Conformity</b>		Automation Automotive Nuclear	Aero Space Railway

- ➔ Homogeneity of principles for the group of 4 mean-oriented standards :
  - › Automation
  - › Automotive
  - › Nuclear
  - › Railway
    - Dealing the with a qualitative assessment of the equivalence of the highest DALs seems tractable.
    - Quantified equivalence ?
  
- ➔ Comparing the confidence level ensured by DO-178 level A with that ensured by SIL 4, ASIL D, level 3 or SSIL 4 ?

- ➔ Participation to creation / revision of standards
  - › IEC 61508 (industry)
  - › Merging IEC/EN 62061 "Safety of machinery" and ISO 13849 (industrial control systems)
  - › ARP 4761, ED202, ED203, ED204 (aeronautics)
  - › ISO 26262 (automotive)
  - › ECSS Q30, Q40, Q80 (space)
  
- ➔ Software Safety and Software Safety Analysis Across Domains and Safety Standards (to appear at ERTSS 2017)
  
- ➔ Development of an example aside RESSAC\* to exhibit: joint use of test and formal methods, coverage measures, unintended functions

\* Re-Engineering and Streamlining Standards for Avionics Certification



Thank you for your attention