MAENAD

Grant Agreement 260057

Model-based Analysis & Engineering of Novel Architectures for Dependable Electric Vehicles

Report type — Deliverable D2.2.1

Report name — Design methodology

Methodology description for embedded systems development with EAST-ADL

Dissemination level	**PU**
Status	**Final**
Version number	**3.0**
Date of preparation	**2013-08-31**

Authors

Editor	E-mail
J. Fiedler	CON (jens.fiedler@continental-corporation.com)

Authors	E-mail
R. Librino	4SG (renato.librino@4sgroup.it)
H. Lönn	VTEC (henrik.lonn@volvo.com)
F. Stappert	CON (friedhelm.stappert@continental-corporation.com)
F. Tagliabò	CRF (fulvio.tagliabo@crf.it)
S. Torchiaro	CRF (sandra.torchiaro@)crf.it
S. Voget	CON (stefan.voget@continental-corporation.com)

Reviewers	E-mail
Henrik Lönn	VTEC (Henrik.lonn@volvo.com)
R. Librino	4SG (renato.librino@4sgroup.it)

Approval	Date
Henrik Lönn	2013-08-31

The Consortium

Volvo Technology Corporation (S) Centro Ricerche Fiat (I)

Continental Automotive (D) Delphi/Mecel (S) 4S Group (I)

ArcCore AB (S) MetaCase (Fi) Systemite (SE) CEA LIST (F)

Kungliga Tekniska Högskolan (S) Technische Universität Berlin (D) University of Hull (GB)

Revision Chart and History Log

Version	Date	Reason
0.1	2010-12-07	First internal release
0.1	2011-01-19	Edited by VTEC
0.5	2011-06-20	Second internal release
0.9	2011-12-14	Inclusion of the methodology
1.0	2012-01-30	Intermediate release
2.0	2012-11-30	2nd intermediate release
2.1	2013-05-24	Internal draft
2.9	2013-08-09	Final draft for review
3.0	2013-08-31	Final release

List of Abbreviations

Abbreviation	Description
FEV	Fully Electric Vehicle
V&V	Validation & Verification
EPF	Eclipse Process Framework
BPMN	Business Process Model and Notation

Table of Contents

1 Introduction

During the ATESST2 project the EAST-ADL methodology has been defined, to give guidance on the use of the language for the construction, validation and reuse of a well-connected set of development models for automotive systems.

The aim of the MAENAD project is to extend the EAST-ADL methodology for the engineering of FEV.

The following aspects related to methodology have been addressed:

- Specific requirements in FEV engineering and specific applicable standards (e.g. high voltage, flammability of batteries, high current switching);
- Application of safety concepts in FEV as defined in ISO 26262, supported by EAST-ADL and novel techniques for automated fault tree analysis and FMEA;
- Application of automated techniques for ASIL decomposition;
- Application of new concepts for V&V, e.g. using behavioral simulation, fault simulation and fault injection;
- Introduction of new concepts for overall safety assessment, providing sufficient evidence of application of ISO 26262 concerning the design process and the relevant work products, including requirements capturing and modeling, completeness of safety analysis, of the safety case, and of the V&V

The following steps were performed to define a detailed methodology based on EAST-ADL for engineering of FEV systems, using a seamless integrated approach compliant with ISO 26262 Functional Safety requirements:

- Review of the already existing EAST-ADL methodology in terms of compliance with the last version of ISO 26262. Moreover the ISO26262 activities and work products not yet included in the EAST-ADL methodology were identified.
- EV standards & regulations analysis: the requirements coming from EV standards and regulations will be analyzed in detail to identify the requirements to be considered relevant for MAENAD approach.
- Integration of ISO 26262 concepts and EV needs into the EAST-ADL methodology

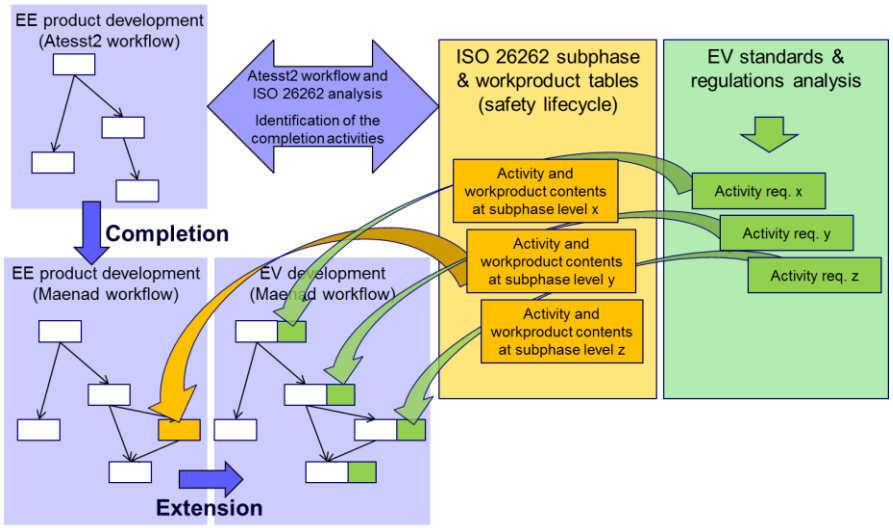


Figure 1 – Illustration of the methodology evolution process

The methodology is based on a set of elementary work tasks which produce output artifacts that serve as input for the next work task. These tasks are structured into disciplines which the system developer or process expert would synthesize to an appropriate work flow. This leads to a highly linked network of methodological activities in which an end user can easily navigate to get information and guidance on the use of the language for particular development tasks.

Technically, modeling of the methodology has been done using Business Process Model and Notation (www.omg.org/spec/BPMN/2.0/). The MAENAD methodology is intended to be a compoundable methodology where activities and work products related to different aspects of development are documented separately. Generic aspects are represented by safety and timing, a domain instantiation is represented by electrical vehicle development. This is manifest as “swimlanes” in the methodology model.

The tooling used for methodology modeling allows publishing an html export as main methodological artifact for the end user.

2 Overall Design Process

Given the complexity of the development activities in automotive embedded system development, it is mandatory to structure the methodology so as to enable a relatively fast and easy access to the EAST-ADL language for a small kernel of essential development activities. These can then be seamlessly extended to a comprehensive treatment of the language including more specialized development activities which may not necessarily be used in all development projects. Hence the methodology is structured into swimlanes representing different aspects of the language.

Figure 2 shows a typical V model, and how the EAST-ADL artifacts typically relate to such work-flow. The four model structures at the left side of the V correspond to phases in the EAST-ADL methodology. Focus is currently on the left side of the V, and virtual integration is used to verify and validate in each phase. Physical integration as represented in the right side of the V is not covered explicitly in the current methodology.

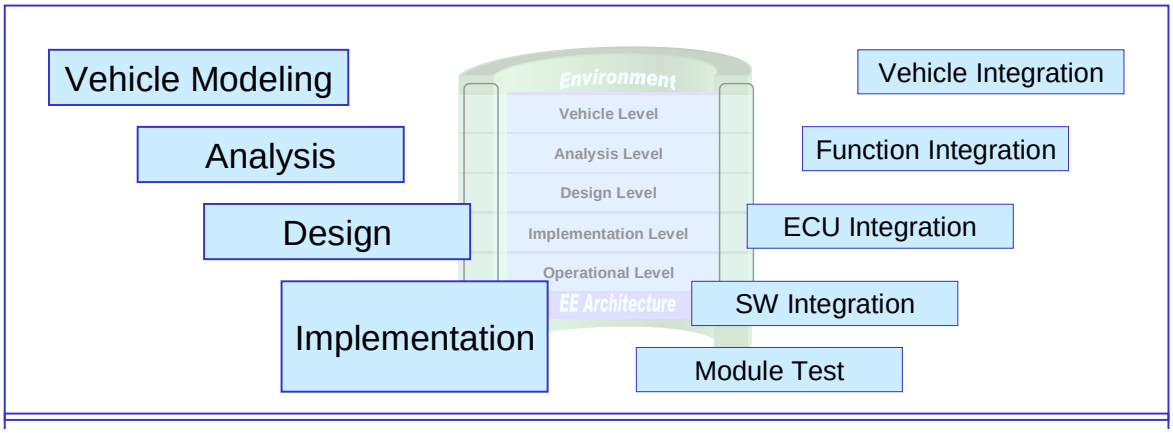


Figure 2 – EAST-ADL artifacts in a V-model context

The main component, the kernel development part, comprises a top-down description of the central constructive phases of automotive embedded software development:

- **Vehicle Modeling:** The analysis of external requirements resulting in the construction of a top-level vehicle feature model together with the definition of necessary or intended feature configurations. In addition, for each feature a set of requirements is specified at vehicle level.
- **Analysis:** The creation of a functional analysis model specifying a solution of the requirements without concern about implementation restrictions of automotive series development. The analysis model is a logical representation of the system to be developed and its environment, and the boundary of the system to its environment. All the modeling in this phase will be on a logical behavior level, i.e. it will make no distinction between HW and SW or about the implementation of communication. Behavior may be specified in detail by executable models.
- **Design:** The creation of a functional design model specifying a solution to the requirements in terms of efficient and reusable architectures, i.e. sets of (structured) HW/SW components and their interfaces, a hardware architecture, and a mapping from functional components to HW/SW components. The architecture must satisfy the constraints of a particular development project in automotive series production.

- **Implementation:** The HW/SW implementation and configuration of the final solution. This part is mainly a reference to the concepts of AUTOSAR which provides standardized specifications at this level of automotive software development. However, the use of AUTOSAR concepts is not mandated by the methodology. Other, in particular more traditional implementation concepts can be used in this phase while leaving the other phases unchanged.

3 MAENAD Methodology

The MAENAD methodology is modeled in BPMN2.0 using the open source Eclipse based tool ADONIS. The methodology is packaged as a HTML file set allowing end users to browse the methodology. This chapter describes the structure and basic modeling principles of the methodology.

3.1 Methodology Modeling Principles

The methodology is modeled in “swimlanes”. The core development methodology leading a developer through the EAST-ADL language is modeled in the “Core” lane. It is structured in 7 steps according to the Generic Method Pattern (GMP) identified together with the TIMMO-2-USE project.

The structuring of swimlanes follows a separation of concerns principle, in which core system design activities are separated from activities related to specific aspects, as safety, timing and FEV. This is shown in Figure 3, where core activities, safety activities, timing activities and FEV activities are organized in four separated lanes covering the all EAST-ADL phases.

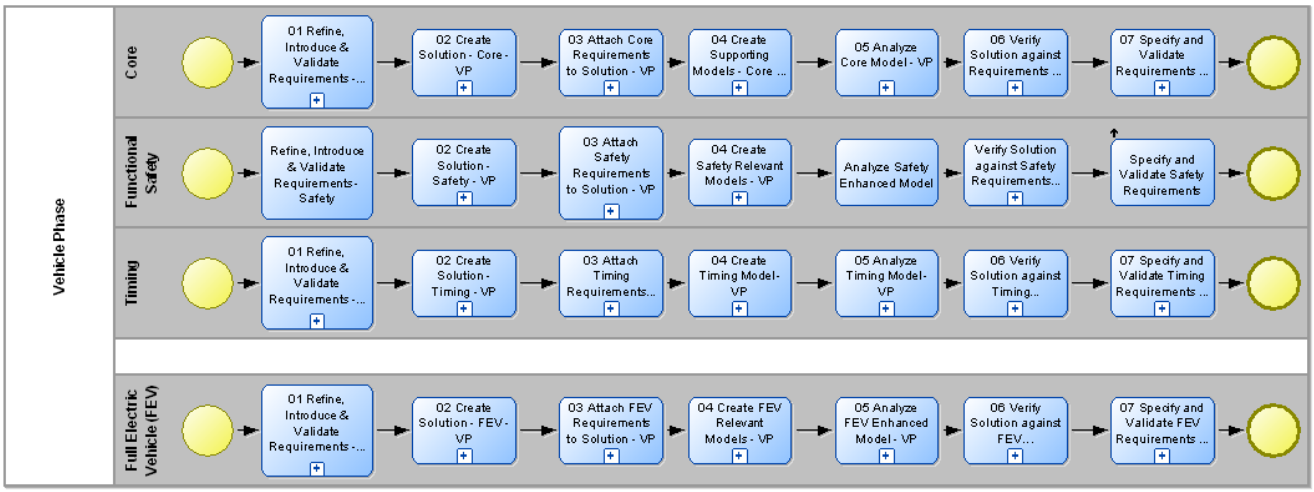


Figure 3: Example from methodology to illustrate the methodology modeling principles

The swimlanes can be included or excluded in a process depending on the needs of a specific project. E.g. in case of a FEV vehicle development, the FEV swimlane would be considered for the process.

Functional Safety swimlane:

The Functional Safety lane has been modeled by taking into account the ISO 26262 safety life-cycle, by giving support for the development of several of the ISO safety life-cycle phases.

Following a top-down approach, the functional safety analysis starts from the EAST-ADL Vehicle Phase, beginning from the item definition (in terms of “target feature”), and the malfunction definition (feature flaws), as anomalies of the item's outputs. On vehicle level, it is already possible to perform a hazard analysis and risk assessment, to estimate the level of risk associated to the Item, and to define a safety goal (and if possible the safe state(s)) for each hazardous event identified. The functional safety analysis continues on the EAST-ADL Analysis Phase, by defining the functional safety requirements, and by allocating them on a preliminary architecture. Once the

functional safety concept is specified, the item can be developed with a system perspective that includes detailed architectural solutions and hardware platform on the EAST-ADL Design Phase. It is then possible to define the technical safety requirements and allocate them on the architectural elements

Timing swimlane

The Timing swimlane gives guidelines on timing related activities that might be conducted on EAST-ADL phases. Timing activities that are applicable on a phase are subject to the system information available on that phase. As presented in Figure 4, for each phase, the timing swimlane consists of a modeling activity where the core model is enriched with timing model elements coming from the timing requirements and, timing verification activities of the timed model.

FEV swimlane

The Fully Electric Vehicle swimlane is a guideline to develop FEVs by addressing the systems that are specific of this kind of vehicles, so as to provide designers with a fundamental complement of the general methodology, which deals with other concerns in the parallel swimlanes.

In particular, the functions and the systems considered have been grouped as follows:

- Electric propulsion
- Regenerative Energy Storage
- Regenerative Braking
- Recharging
- Energy conversion
- Insulation and Protection
- Anti-theft system
- Human Machine Interface (HMI)

In order to provide an effective support to FEV development, for most of the activities of the process, the reference to the applicable standards and regulations, and some synthetic requirements of the norms are given. In fact, most of the FEV specific issues are well addressed by norms, and only some issues are left as additional and usually necessary activities to be included in the development process. It has to be mentioned that the anti-theft system has been explicitly listed because some EV standards require such a system and prescribe its operation.

The references include the relevant ISO, IEC, EN, SAE standards, and UNECE and FMVSS regulations.

3.2 Methodology Model Overview

Figure 4 shows the top level view of the methodology. The methodology follows one to one the abstraction level principle of the EAST-ADL language, starting from the most abstract level, the vehicle phase, to the most concrete level, the implementation phase.

The methodology shows an idealized forward process oriented view only. It is possible to work with the EAST-ADL model top-down, bottom-up and middle-out; iterations are typically needed in most activities. This is, however, implicit and not encoded in the methodology model. Rather, it can be used during practical application of the methodology and for process instantiation in a concrete project.

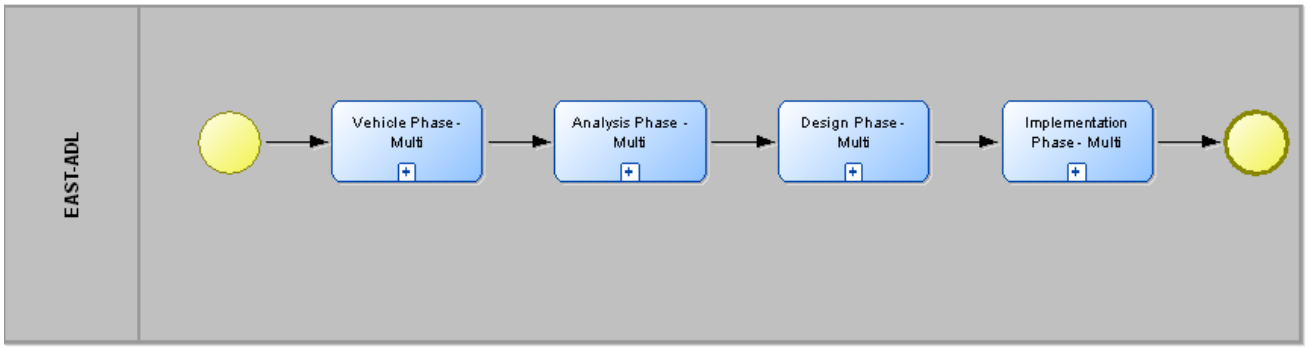


Figure 4 – Top level of the MAENAD methodology

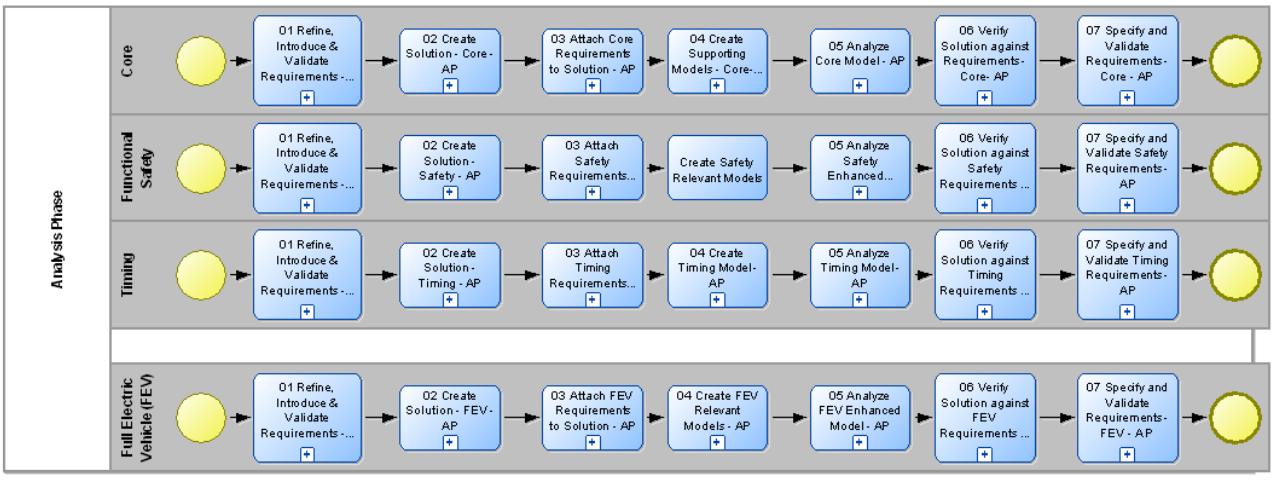


Figure 5 – Analysis phase of the MAENAD methodology

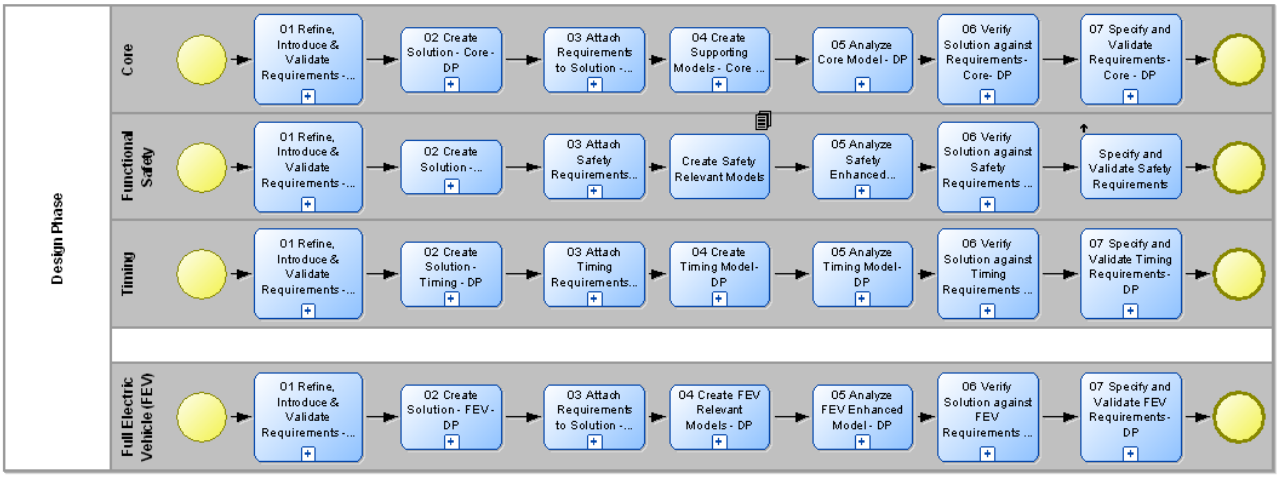


Figure 6 – Design phase of the MAENAD methodology

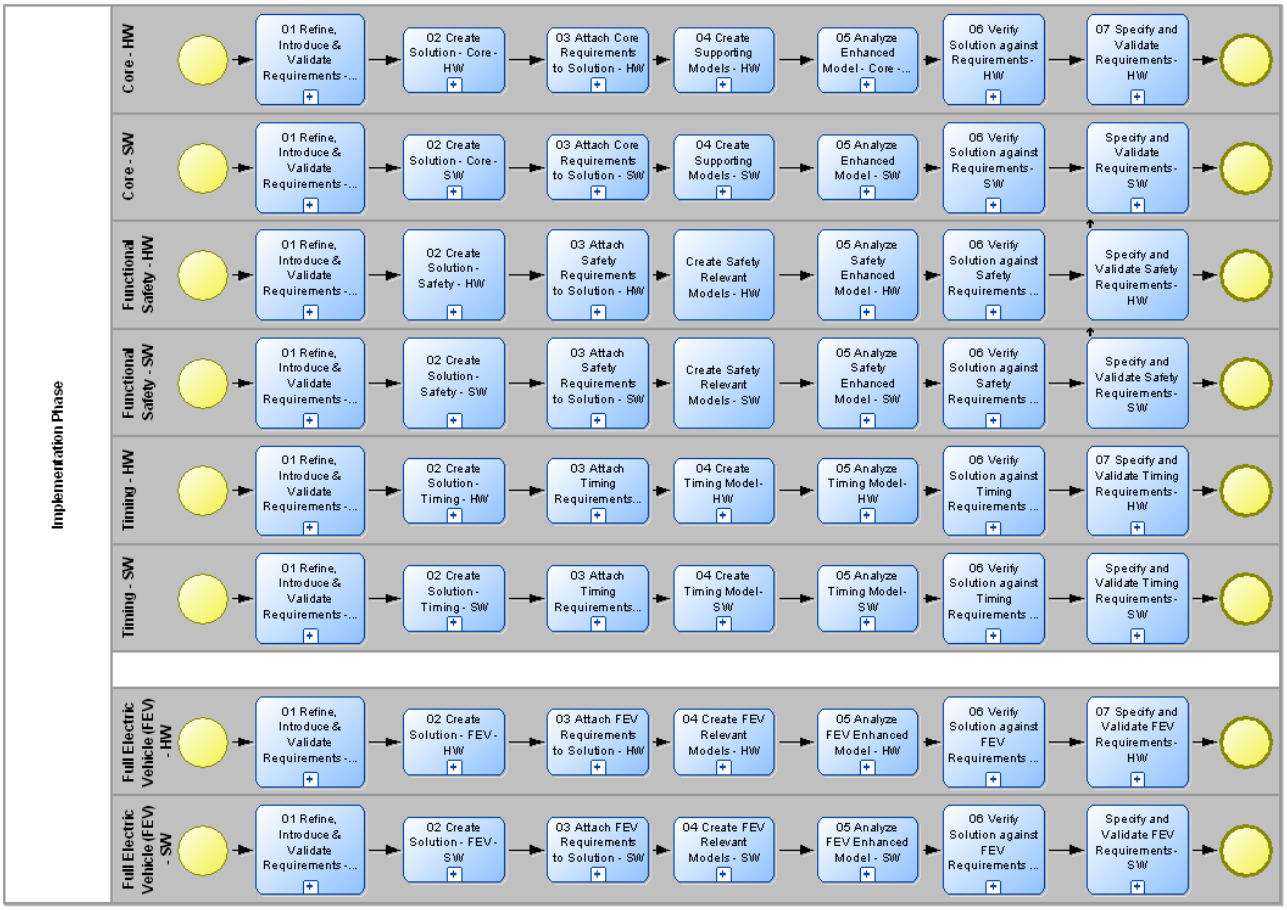


Figure 7 – Implementation phase of the MAENAD methodology

Standards managing functional safety aspects (e.g. ISO26262) usually are structured in a process-oriented manner. On the other hand standards which describe FEV specific features often refer to a concrete work product or to a specified vehicle part.

Following these strategies the safety aspects given in the tables in chapter 4 (Methodology Analysis) and in chapter 8 (Appendix 1: ISO26262 Requirements) are grouped according the development process. The FEV aspects described in chapter 5 (Design Methodology Checklist for FEVs) are assigned to a specific subject.

4 Methodology Analysis and Refinement for Safety and Electrical Vehicle Design

To define the MAENAD EAST-ADL methodology, the ATESST2 methodology was used as a basis. The normative regulation for functional safety in the automotive domain, ISO 26262, has phases with sub-phases comprised by work products, tools and the responsible role. Both process models were compared regarding functional safety to identify needs on methodology content. The following tables show an excerpt of this analysis.

ISO Part	Phase	Sub-phase	Work Products	Tools	Activity Responsible	LINK to Product Development Work Flow (EAST-ADL Based)	EAST-ADL artifacts
Part 2	Safety Management	Overall safety management	Organization-specific rules and processes for functional safety	Not applicable	*Product Liability Manager*	Vehicle Level Analysis Level Design Level	N/A
			Evidence of competence	Not applicable	*Product Liability Manager*	Vehicle Level Analysis Level Design Level	N/A
			Evidence of quality management	Not applicable	*Product Liability Manager*	Vehicle Level Analysis Level Design Level	N/A
		Safety management during the concept phase and the product development	Safety plan	Compatible with traceability requirements, change mangement.	Safety Manager	Vehicle Level Analysis Level Design Level	VVCase for detailed activities, SafetyCase structure for overall information structure, Requirements with Satisfy relation to Ground to detail the evidence required.
			Project plan (refined)	Not applicable	Project Manager		N/A
			Safety case	Compatible with traceability requirements, change mangement, configuration…	Safety Manager	EAST-ADL Quality+Safety Process > Analysis Phase > Functional Safety Requirements; EAST-ADL Quality+Safety Process > Design Phase > Functional Safety Requirements; EAST-ADL Quality+Safety Process > Analysis Phase > Safety Goals; EAST-ADL Quality+Safety Process > Analysis Phase > Perform Risk Assessment Validation > Safety Goals; EAST-ADL Quality+Safety Process > Vehicle Phase > Perform Safety Analysis > Safety Goals; EAST-ADL Quality+Safety Process > Design Phase > Safety Goals	SafetyCase Functional Safety Requirements Safety Goals
			Functional safety assessment plan	Compatible with traceability requirements, change mangement.	Safety Manager		N/A
			Confirmation measure reports	Compatible with traceability requirements, change mangement.	Safety Manager		Warrant.Evidence
		Safety management after the item's release for production	Evidence of field monitoring	Compatible with traceability requirements, change mangement, configuration…	*Persons appointed to maintain functional safety after release for production*		Warrant.Evidence

Table 1: Example EAST-ADL methodology elements for ISO26262 part 2

ISO Part	Phase	Sub-phase	Work Products	Tools	Activity Responsible	LINK to Product Development Work Flow (EAST-ADL Based)	EAST-ADL artifacts
Part 3	Concept Phase	Item Definition	Item Definition	MS Visio (Function modelling and visualisation tools) / Dimension RM / MKS / DOORS (Requirement management tools) / Enterprise Architect (UML) / Papyrus 'EAST ADL' (Structure/Architecture modelling tools)	Safety Manager	Safety … feature model /	Item references Features Realizing Artifacts define SystemBoundary Features describe purpose and functionality, including operating modes and states on user level Features (its use cases, requirements, refined requirements) describe interactions with other items or elements on user level, Realizing Artifacts describe interactions with other items or elements of solution Features (its requirements) define Applicable laws and regulations, national and international standards OperatingScenario on Item describes operating scenarios which impact the functionality of the item. Requirements on Features define expected or required environmental conditions that are independent of solution, Requirements on Artifacts define expected or required environmental conditions that are dependent of solution ErrorModels linked o artifacts identify known
		Initiation of the safety lifecycle	Impact Analysis	MS Visio (Function modelling and visualisation tools) / Dimension RM / MKS / DOORS (Requirement management tools) / Enterprise Architect (UML) / Papyrus 'EAST ADL' (Structure/Architecture modelling tools)	Safety Manager	Vehicle Level Analysis Level Design Level	Documentation with references to relevant elements in existing and modified model. Documentation element and its references to elements TBD. Symbolic Expression?
			Safety plan (refined)	Compatible with traceability requirements, change mangement.	Safety Manager	Vehicle Level Analysis Level Design Level	See Safety Plan
		Hazard analysis and risk assessment	Hazard analysis and risk assessment	Relex / IQ-FMEA / SAM 2000 / Dimension RM / MKS / DOORS (Requirement management tools)	Safety Manager	Vehicle Level, ASIL for each Hazardous Event EAST-ADL Quality+Safety Process > Design Phase > Complete Risk Assessment; Analysis Level, ASIL for each Hazardous Event EAST-ADL Quality+Safety Process > Analysis Phase > Perform Risk Assessment Validation > Complete Risk Assessment; EAST-ADL Quality+Safety Process > Vehicle Phase > Perform Safety Analysis > Complete Risk Assessment; EAST-ADL Quality+Safety Process > Design Phase > DecomposedASIL; EAST-ADL Quality+Safety Process > Vehicle Phase > Perform Safety Analysis > Hazard List; EAST-ADL Quality+Safety Process > Analysis Phase > Perform Risk Assessment Validation > Hazardous Events; EAST-ADL Quality+Safety Process > Vehicle Phase > Perform Safety Analysis > Hazardous Events;	FeatureFlaw, Hazard and HazardousEvent ASIL for each Hazardous Event Complete Risk Assessment DecomposedASIL Hazard List Hazardous Events
			Safety goals	Not applicable	Safety Manager		SafetyGoal
			Verification review report of the hazard analysis and risk assessment and the safety goals	Not applicable	Safety Manager		Report generated from Item's FeatureFlaw, Hazard and HazardousEvent combined withh any comments and Rationale.
		Functional safety concept	Functional safety concept	BlockSim 'FTA' / APIS IQ-FMEA / Relex (Reliability analysis tools) / Dimension RM / MKS / DOORS (Requirement management tools) / Enterprise architect / MS Visio / Papyrus 'EAST ADL' (tools for architecture development)	Safety Manager	EAST-ADL Quality+Safety Process > Vehicle Phase > Perform Safety Analysis > Operational Conditions	Requirements in a FunctionalSafetyConcept related with Satisfy links to FAA without redundancy or safety measures OR Requirements in a FunctionalSafetyConcept related with Satisfy links to FAA with redundancy and safety measures (In case safety solutions are modelled as a modified FAA, this structure may also linked with a refine relation to a Functional Safety Requirement. The original FAA stays non-redundant in that case) ("Preliminary physical architecture, in which functionality is allocated" appears too be too early. ISO26262 does not mention preliminary hardware, it only mentions architectural elements
			Verification report of the functional safety	Not applicable	Safety Manager		Requirements and related VVCase structure

Table 2: Example EAST-ADL methodology elements for ISO26262 part 3

FEV specific standards

Further on, standards concerning FEV are analyzed in order to identify the requirements that should be considered relevant to MAENAD, especially those regarding E/E addressing functionality, safety, communication, thus excluding mechanics, environmental conditions, EMC, operational procedures not related to the design phase.

The following normative standards concerning FEV are used:

- SAE – J2289 Electric-Drive Battery Pack System: Functional Guidelines.
- ISO 6469-1 Electrically propelled road vehicles – Specific requirements for safety – Part 1: On board energy storage
- ISO 6469-2 Electric road vehicles – Safety specifications – Part 2: Vehicle operational safety means and protection against failures
- ISO 6469-3 Electric road vehicles – Safety specifications – Part 3: Protection of persons against electric hazards
- R.116 and subsequent amendments

The identified requirements are evaluated to define further requirements, which should be captured in MAENAD, in terms of:

- system description and modeling requirements
- methodological requirements for system design

The SAE – J2289 collects a set of requirements for the Electric-Drive Battery Pack System. These requirements are mapped in MAENAD to system description and modeling. Further requirements for the design methodology are derived.

In detail there are requirements for

- Modes and associated electrical modes
- Key on – Discharge
- Key on – Regen Operation
- Key on – Charge
- Key-Off Parked Off Plug Operating
- Parked Off Plug IDLE/Storage Operation
- Traction Wiring and Connectors Sensor Wiring
- Contactors/Disconnects
- Electrical Isolation
- Discharge Management – Performance Limits
- Charge Management
- Key-On Startup Diagnostics and Warning
- Key-On Running Diagnostics and Warning
- Service Diagnostics
- Multiplex Communication Interface

- Toxic Emissions
- Flammable Gasses

The ISO 6469-1, Part 1, collects requirements for “On board energy storage”. A detailed list is given in the following enumeration:

- The measurement of the isolation resistance of the RESS shall include auxiliary components located inside the RESS housing, e.g. monitoring or temperature-conditioning devices and liquid fluids (if any).
- Heat generation under any first-failure condition, which could form a hazard to persons, shall be prevented by appropriate measures, e.g. based on monitoring of current, voltage or temperature.
- RESS over-current interruption: If a RESS system is not short-circuit proof in itself, a RESS over-current interruption device shall open the RESS circuit under conditions specified by the vehicle and/or RESS manufacturer, to prevent dangerous effects for persons, the vehicle and the environment.

The ISO 6469-2, Part 2, collects “Vehicle operational safety means and protection against failures”. The following enumeration is given:

- Electric road vehicles - Safety specifications - Part 2: Functional safety means and protection against failures
- Operational safety -Connection of the vehicle to an off-board electric power supply
- Operational safety – Driving - Indication of low energy content of RESS
- Operational safety - Driving backwards
- Operational safety – Parking
- Protection against failures

The ISO 6469-3, Part 3 focuses “Protection of persons against electric hazards”. Also Safety requirements are described regarding:

- Measures and requirements for protection of persons against electric shock - Protection under first failure conditions
- Measures and requirements for protection of persons against electric shock - Alternative approach for protection against electric shock
- Measures and requirements for protection of persons against electric shock - Isolation resistance requirements
- Measures and requirements for protection of persons against electric shock - Requirements of potential equalization
- Requirements for vehicle charging inlet - Voltage decrease requirement
- Requirements for vehicle charging inlet - Grounding and isolation resistance requirement for charging inlet

In order to define some FEV design processes addressing the different subjects covered by the standards and regulations, the design methodology requirements have been analyzed, so as to identify the design activities that shall be performed according to the standards and regulations. The design phases considered are related only to E/E systems, but include also the planning of test activities, whenever the planning should be performed during the design phase, also according to ISO 26262.

The following processes have been defined:

- Design of an insulation monitoring system
- Design the Regenerative Energy Storage System
- Design of the Regenerative Braking System
- Design of conductive charge coupling
- Design of the vehicle operation modes
- Design of theft protection system

The figures hereafter show the highest level representation of the design processes, while the detailed description at lower level in terms of subprocesses and activities is reported using the tool Adonis. In order to provide a useful guideline to FEV designers, the textual description of the subprocesses and of the activities include the reference to the standards and regulations. It should be pointed out that some activities refer to more than one standards or regulations. Designers should identify the applicable standards and regulations according to the specific system under development or the legislative constraints.

It has to be pointed out that all the above design processes are FEV specific. The last one (Design of theft protection system) is also EV specific, because it is intended to ensure the safety of EVs by preventing the unauthorized use of FEVs, which can be dangerous.

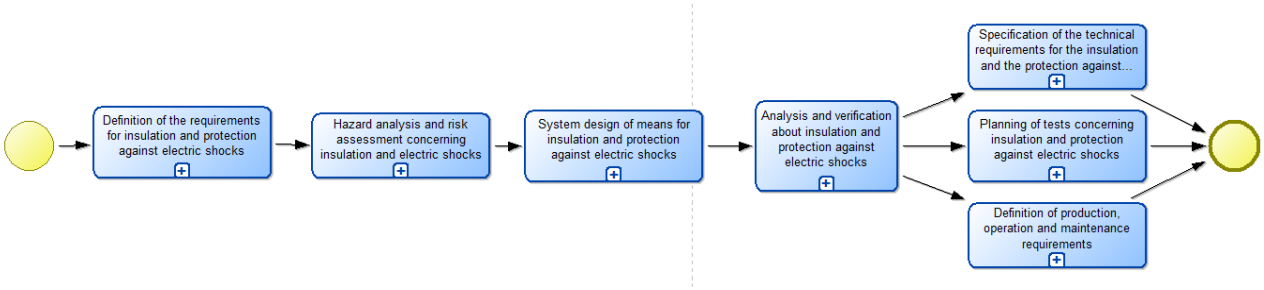


Figure 8 – Design of an insulation monitoring system

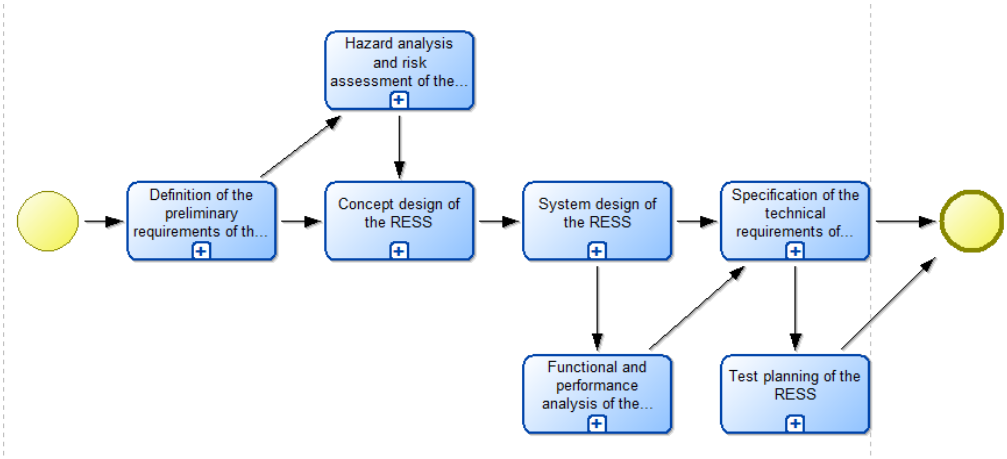


Figure 9 – Design the regenerative energy storage system

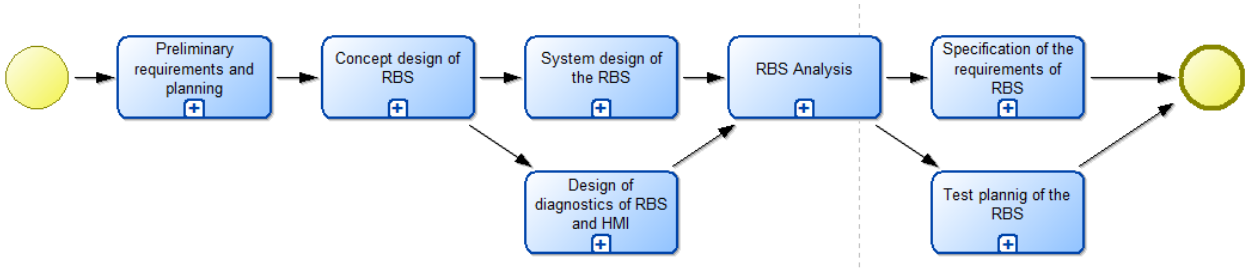


Figure 10 – Design of the regenerative braking system

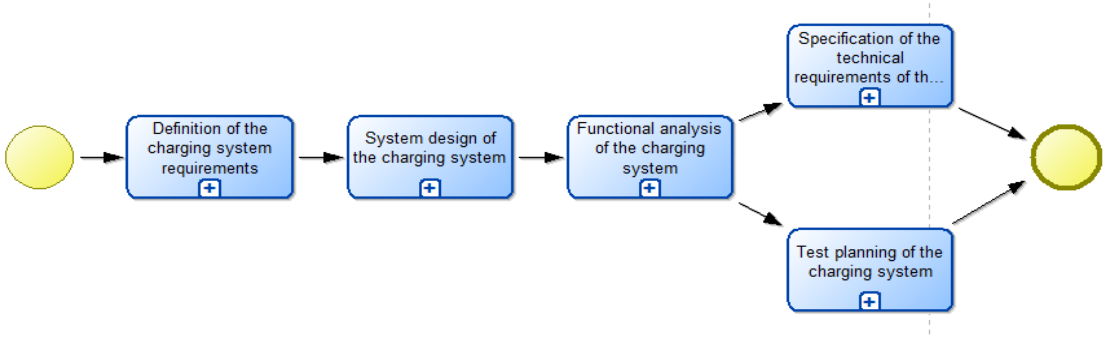


Figure 11 – Design of conductive charge coupling

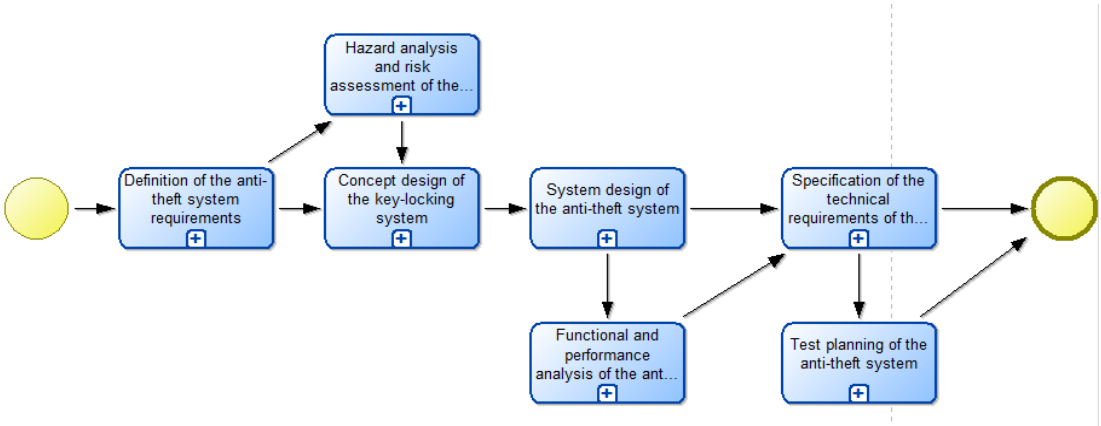


Figure 12 – Design of theft protection system

The identified methodology steps were organized according to the generic methodology pattern and represented in the FEV Swimlane, as described in chapter 3.

5 Design Methodology Checklist for FEVs

As reported in D2.1.1, a process was followed to define the requirements related to FEV development, in order:

- to verify the capability of the current version of EAST-ADL2 to cover the needs related to specific characteristics of FEVs, and to extend its features if necessary; and similarly;
- to verify the capability of the analysis tools and to give inputs to adapt or, possibly, create specific tools to perform the necessary analyses;
- to define an extension of the basic E/E system development methodology resulted from ATTEST2, in order to help designers to perform the development activities required by the standards and the regulations, or those compliant to best practices or engineering needs for EV development.

Therefore, through a sequence of activities according to a bottom-up approach, three categories of requirements have been defined: language requirements, analysis requirements, and methodology requirements.

The requirements defined have been reported in an Excel sheet and, subsequently, in a UML model in the tool Enterprise Architect, to comply with the method followed for the collection of MAENAD requirements, thus allowing better traceability, uniform categorization and assignment to WPs.

The following table is an excerpt of the Excel file and includes only the methodology requirements. The table can be seen as a checklist for development projects in consideration of specific FEV aspects within the MAENAD methodology.

Reference are given to the requirement codes used in EA; the field “subject” has been introduced to better identify the related engineering topic and to establish a link with the language and analysis requirements related to the same topic.

It has to be pointed out that in the following table some language requirements are referred to a specific standard or regulation. However, the requirements, in some cases, can be referred to similar standards (not mentioned here, but only in the Excel sheet, which gives a more global view of the analysis conducted to define the requirements).

First level user requirements		Second level user requirements Design methodology requirements		
Code	**Title**	**Subject**	**Requirement description**	**Code**
4SG 7	EV safety standards/ ISO 6469-1	Insulation	- Deployment of insulation resistance - Addressing insulation monitoring system - Hazard analysis and risk assessment concerning insulation monitoring - Design issues concerning recharging (grounding, communication) - Test planning concerning insulation - Production, operation and maintenance requirements during design phase (ISO 26262-4)	4SG78
		Heat generation	Designing a monitoring system to prevent dangerous effects to persons, in the case of failures producing heat generation	4SG79
		RESS over-current interruption	- Designing an over-current interruption device - Hazard analysis in the case of short circuit of RESS - Planning of short circuit test	4SG82
4SG 8	EV safety standards/ ISO 6469-2	Connection of the vehicle to an off-board electric power supply	Designing a means to make impossible to move the vehicle when connected to off-board electric power supply and charged by the user	4SG83
		Indication of reduced power	Designing a warning to signal to the driver that the propulsion power is reduced, in the case this is done	4SG84
		Driving backwards	Designing means to prevent unintentional switching in reverse when the vehicle is in motion (two options are available)	4SG85
		Parking	Designing a warning to indicate whether propulsion is in the driving–enable mode, when user leaves the vehicle. Designing a safety mechanism to prevent unexpected movements.	4SG86
		Protection against failures	In functional safety development, include unintended acceleration, deceleration and reverse motion as	4SG87

First level user requirements		Second level user requirements Design methodology requirements		
Code	**Title**	**Subject**	**Requirement description**	**Code**
			hazards to be prevented or minimized.	
4SG 9	EV safety standards/ ISO 6469-3 Protection of persons against electric hazards	Protection of persons against electric shock	Designing mechanical and electronics means according to the standard. Verification planning for measures protection (design verification, test plan)	4SG88
		Alternative approach for protection against electric shock	Conduct an appropriate hazard analysis with respect to electric shock and establish a set of measures which give sufficient protection against electric shock	4SG89
		Isolation resistance requirements	Assignment of insulation resistance to high voltage components as to achieve the overall insulation resistance (dc, ac cases).	4SG90
		Requirements of potential equalization	Designing insulation barriers and bonded conductive equalization barriers. Planning verification of barriers, including bond testing.	4SG92
		Charging inlet disconnection	Designing charge system, as to ensure voltage decrease of inlet according to time requirements. Verification by simulation, analysis and testing.	4SG94
		Grounding and isolation resistance requirement for charging inlet	Designing charging system as to meet insulation requirements in the case of ac and ac inlet.	4SG95
4SG 16	EV safety standards/ EN 61851	Types of EV connection	- Define the charging system according to one of the 4 charging modes. - Define the control pilot mandatory and optional functions (modes 2-4), including charging operation states.	4SG96
		Protection against electric shock	Define and provide measures to prevent electric shock both in normal service and in case of fault.	4SG97
		Stored energy – discharge of capacitors	Design the EV voltage input in such a way to control the voltage decay after EV disconnection	4SG99
		Detection of the electrical continuity of the protective	Design a monitoring system to detect the electrical continuity of the protective conductor during charg-	4SG100

First level user requirements		Second level user requirements Design methodology requirements		
Code	**Title**	**Subject**	**Requirement description**	**Code**
		conductor	ing modes 2, 3 and 4.	
		Dielectric withstand voltage	Design the on board charging equipment as to withstand the test voltage at any input connection (2U +1000 V, min. 1500 V ac). Design all vehicle equipment as to withstand a test voltage of 4kV between ac or dc input and low voltage inputs (if any).	4SG101
		Electric vehicle insulation resistance	Verify the insulation resistance (by analysis and testing). Minimum required: 1 MΩ.	4SG102
		Drive train interlock	Design a system to detect the connection of the mobile connector or that the plug and the cable have been stored in the vehicle. The system shall also inhibit the drive train	4SG103
4SG 18	EV safety standards/ J2289	Vehicle operational modes	- Defining the vehicle operational modes - Justify possible discrepancies	4SG104
		Key-on discharge	- Assessment of battery capability to match the vehicle demand (range, supply of auxiliary equipment) - Designing means to detect and limit the overdischarge of individual cells - Providing fault protection devices (fuses, fast contactors)	4SG107
		Key-on Regen operation	- Assessing the compliance of the voltage with the limits during regeneration - Providing design means to avoid drive component overvoltage occurrence during regeneration - Verifying the compliance with current and voltage profiles - Providing design means to limit battery current and voltage during regeneration according to the specified profiles	4SG110
		Key on – Charge	- Verifying that all charge system components match w.r.t. electrical characteristics - Designing charge algorithm with	4SG113

First level user requirements		Second level user requirements Design methodology requirements		
Code	**Title**	**Subject**	**Requirement description**	**Code**
			the battery supplier	
		Key-Off Parked Off Plug Operating	- Providing energy management to prevent excessive discharge due to vehicle equipment operating in key-off mode - Verify energy behavior in key-off mode by simulation/calculation - Designing charge algorithm with the battery supplier	4SG116
		Parked Off Plug IDLE/Storage Operation	Designing a battery disconnect system for operation during storage or maintenance	4SG118
			- Designing contactor operation as to be deactivated in the case of crash or isolation fault - Designing disconnect system for added safety during service or by first responders during accidents.	4SG119
		Discharge management - Performance limits	Designing BMS to protect for over-temperature, under-temperature, over-current	4SG121
		Charge management	Design communication in compliance with SAE J1772, SAE J1773, and SAE J2293	4SG122
		Key-on startup diagnostics and warning	Design key-on running diagnostics and warning procedures	4SG124
		Service diagnostics	Design service diagnostics	4SG125
		Toxic emissions Flammable gasses	Consider toxic emissions and flammable gasses caused by battery damages	4SG126
4SG 72	FMVSS No. 114 Theft protection	Key-locking device	Design the key-locking system to prevent the activation of the motor and steering or self-mobility (or both)	4SG128
		Parking function	- Design the operation of key-locking system according to the standard (see interaction with park command). - Verify (by calculation and testing) that the maximum movement of the vehicle when locked is less than the max. allowable limit.	4SG129

First level user requirements		Second level user requirements Design methodology requirements		
Code	**Title**	**Subject**	**Requirement description**	**Code**
4SG 73	FMVSS No. 102 Transmission shift lever sequence, starter interlock, and transmission braking effect		Designing the shift lever according to the sequence position and rotation requirements	4SG130
4SG 75	R 116 Theft protection	Locking device	Designing devices to prevent unauthorized use (deactivation of engine in combination with a system to lock other vehicle functions, see regulation)	4SG131
		Locking function	Conduct functional safety analyses to cover the devices intended to prevents unauthorized use	4SG132
4SG 71	FMVSS No. 135 Passenger car brake systems	Regenerative braking system	- Plan the analysis and the development of braking system according to the operation mode of the RBS: control of RBS by ABS (if RBS is always active, also in neutral without any means to disconnect it by the driver, RBS is part of the service braking system). - Item definition: consider the interactions between RBS and ABS (w.r.t. interfacing and system definition in ISO 26262)	4SG133
		Diagnostics and warning	- Include diagnostics task related to RBS, in order to transmit information to the visual warning indicator - Design proper warning in the case of failure of brake power supply, reduced SoC, RBS failure	4SG135
		Braking performance	Plan a braking test in depleted battery state-of-charge condition	4SG137
4SG 19	EV performance standards/ ISO 8715	Performance testing - Test conditions and procedures	Include the simulation of vehicle performance according to test conditions and test procedure requirements Include vehicle performance testing according to test condition and test procedure requirements	4SG141
4SG 20	EV performance standards/ ISO 8714	Energy and range testing - Test conditions and procedures	Include the simulation of vehicle performance according to test conditions and test procedure requirements Include vehicle performance testing according to test condition and	4SG145

First level user requirements		Second level user requirements Design methodology requirements		
Code	**Title**	**Subject**	**Requirement description**	**Code**
			test procedure requirements	
4SG 23	EV performance standards/ ISO 12405-2	Test sequence - Test conditions	- Simulate vehicle performance according to test conditions requirements(when applicable) - Test vehicle performance according to test conditions requirements	4SG148
4SG 74	SAE J2777 Conductive charge coupler	Control pilot	Model communication protocol based on PWM and signal amplitude (by switching a resistor)	4SG150
		Communication	Design the communication according to the standard (charging station status, power level, fault conditions)	4SG151
		Proximity detection	Design the management of the connector detection signal: to start charge control, to engage drive train interlock, to reduce charge load during disconnection	4SG152
		Charge management	Design the charging state machine according to the standard, including safe states in case of fault.	4SG153
		Charge status indicator	Define the charge status indicator, including diagnostic functions.	4SG154
4SG 70	R 13H Braking	Phasing of braking sources (B category)	If the RBS is part of service brake, design the braking inputs, compensating the variations of the regenerative braking and ensuring breaking action in all wheels.	4SG156
		Integration with ABS	Include a development task to define and manage the interaction between ABS and RBS.	4SG157
4SG 13	EV safety standards/ J2344 Electric Vehicle Safety	Electric isolation	- Design the high voltage insulation (100 Ω/V DC, 500 Ω/V AC) - Design barriers between AC and DC, if the DC limit is applied - Plan testing to demonstrate high voltage withstand capability - Design an isolation loss monitoring system	4SG160
		High voltage automatic disconnect	• Design an automatic disconnect system actuated - by a crash sensor - in the case of loss of isolation, only in non-motoring mode - in the case of overcurrent con-	4SG161

First level user requirements		Second level user requirements Design methodology requirements		
Code	**Title**	**Subject**	**Requirement description**	**Code**
			dition, as a primary or secondary protection - according to the guidelines given by SAE J2344 • Design a crash sensor, properly qualified to operate in the crash tests. • Design the disconnect system to be activated by the crash sensor and to maintain disconnection after crash.	
		High voltage manual disconnect	Design a manual disconnect system actuated by an interlock loop	4SG162
		Grounding	Design grounding of the conductive cases containing high voltage systems, also by means of indirect connection.	4SG163
		Fault monitoring	- Design a fault monitoring system. - Design vehicle operation in such a way that the vehicle operator is not allowed to persist in unsafe condition.	4SG164
		Rechargeable energy storage	Design the operation in low state-of-charge in such a way that - the performance of the critical safety systems is not degraded - the state is indicated in a separate indicator if the vehicle performance is reduced	4SG165
		Mechanical safety	Design a lock system activated when the shift mechanism is in P position or the key is in “off” position.	4SG166

Table 3: FEV methodology checklist

6 Summary and Conclusion

This document has provided an overview of the MAENAD EAST-ADL methodology and a description of how FEV standards and requirements and the ISO26262 standard were analysed to provide input to the methodology definition. There is also a list of FEV system requirements to be used as a checklist for FEV development. The Appendix, finally, contains a list of relevant requirements from the ISO 26262 and how they were met by the EAST-ADL constructs.

The MAENAD methodology is provided as a model complementing this document. The model is represented in HTML to provide easy access through a web browser.

7 References

[1] ATESST2 Deliverable D5.1.1 Methodology guideline when using EAST-ADL2, June 2010.

[2] ISO 26262, First edition (15.11.2011): Road vehicles – Functional safety standard

[3] Maenad_Deliverable_D2.2.1_Appendix.zip: Integrated MAENAD Methodology

8 Appendix 1: ISO26262 Requirements

8.1 Vehicle Level Modeling

8.1.1 Part 3: Concept phase, Clause 5.4.1

„*The functional and non-functional requirements of the item as well as the dependencies between the item and its environment shall be made available. This information includes:*

a) the functional concept, describing the purpose and functionality, including the operating modes and states of the item;

b) the operational and environmental constraints;

c) legal requirements (especially laws and regulations), national and international standards;

d) behaviour achieved by similar functions, items or elements, if any;

e) assumptions on behaviour expected from the item; and

f) potential consequences of behaviour shortfalls including known failure modes and hazards."

Recommendation: This ISO clause is concerned with an early development phase in which the starting point for the functional safety work is defined in the form of an item definition. The topics addressed in the clause should mainly be included in the modeling at the vehicle level, although some specific aspects might be more appropriately addressed at lower modeling levels (i.e. analysis, design or implementation). Checklists for the different levels can be defined where a) – f) above are explicitly included in the respective checklist.

Derived Requirements:

The purpose and functionality of Item shall be defined by means of an Item's Feature(s) and its requirements.

8.1.2 Part 3: Concept phase, Clause 5.4.2

„*The boundary of the item, its interfaces, and the assumptions concerning its interaction with other items and elements, shall be defined considering:*

a) the elements of the item;

b) the assumptions concerning the effects of the item's behaviour on other items or elements, that is the environment of the item;

c) interactions of the item with other items or elements;

d) functionality required by other items, elements and the environment;

e) functionality required from other items, elements and the environment;

f) the allocation and distribution of functions among the involved systems and elements; and

g) the operating scenarios which impact the functionality of the item."

Recommendation: This ISO clause is concerned with an early development phase in which the starting point for the functional safety work is defined in the form of an item definition. The topics addressed in the ISO 26262 requirement above should be included in the modeling at the vehicle level. To support this modeling, a checklist can be defined where a) – f) above are explicitly included in the checklist. However, it does not seem to be appropriate to consider item-internal elements (as indicated in a) and partly in e) above) at this stage.

Derived Requirements:

The elements of the item shall be defined in terms of functional elements on Analysis Level which realize the Item's Features.

The elements of the item shall be defined in terms of functional and resource elements on Design Level which realize the Item's Features.

The elements of the item shall be defined in terms of software and hardware elements on Implementation Level which realize the Item's Features.

The effects of the item's behavior on other items or elements shall be defined through the interface definitions of the elements of the item on Analysis, Design and Implementation level.

Requirements of the item on other items, elements and environments shall be defined through the output interface definition of the Item's elements on Analysis, Design and Implementation level.

Requirements from other items, elements and environment on the item shall be defined through the input interface definition of the Item's elements on Analysis, Design and Implementation level.

The item's functionality shall be realized by elements on Analysis, Design and Implementation level.

Elements on Analysis, Design and Implementation level which realize an item shall be linked to the Item's features with a Realize relation.

Operating scenarios of the item shall be defined in terms of traffic and environment (operating situations) and operational situation (use cases).

8.1.3 Part 3: Concept phase, Clause 7.4.1.1

„*The hazard analysis and risk assessment shall be based on the item definition.*"

Recommendation: The requirement itself is not particularly applicable to model-based development (although it could be included verbatim in a checklist for how to perform the hazard analysis). More importantly however, the requirement implies that traceability should exist between the item definition and the hazard analysis. It is therefore highly desirable that the modeling incorporates such traceability, preferably in both directions. The applicable checklists could support this by explicitly requiring traceability.

Derived Requirements:

Hazard analysis shall be performed for each Item and represented through Hazards, Hazardous Events and related elements.

8.1.4 Part 3: Concept phase, Clauses in chapter 7.4.2

In these clauses, ISO 26262 gives several requirements on how to perform the hazard analysis.

Recommendation: We assume that the hazard analysis itself is performed outside the tool environment for model-based development. Thus, the requirements in ISO 26262 on how to perform this analysis is out-of-scope for these guidelines. However, the results of the hazard analysis should be represented in the models by being linked to the corresponding systems. These results include:

- the identified hazards, preferably expressed as inabilities of the considered system to operate as intended
- the operational situations and operating modes for which the hazards could lead to harm
- the ASIL associated with each identified hazard

It is important that hazards are defined in an appropriate way. They should be defined so that they are fully within the scope of the considered system. A hazard should not be defined so that it can only occur when certain environmental conditions are fulfilled. For example, "*the airbag will not be activated if an airbag-relevant collision occurs*" is a good example of a hazard. The hazard itself can exist independently of the driving situation even though it is only in an airbag-relevant collision that the hazard would really have an effect. In other words, the hazard can exist even if there is no collision. A less appropriate hazard formulation would be "*the vehicle is involved in an airbag relevant collision but the airbag is not activated*". This situation can only occur when there is a collision so it is not independent of the driving situation. In fact, this second example is a 'hazardous event' rather than a hazard in the ISO 26262 terminology.

Derived Requirements:

All identified Hazards shall be represented as Hazards and linked to the Item.
Hazardous Events shall be defined and its corresponding operational situation.

8.1.5 Part 3: Concept phase, Clauses 7.4.4.3 – 7.4.4.6

Clause 7.4.4.3: „*A safety goal shall be determined for each hazardous event with an ASIL evaluated in the hazard analysis. If similar safety goals are determined, these may be combined into one safety goal.*"

Clause 7.4.4.4: „*The ASIL determined for the hazardous event shall be assigned to the corresponding safety goal. If similar safety goals are combined into a single one, in accordance with 7.4.4.3, the highest ASIL shall be assigned to the combined safety goal.*"

Clause 7.4.4.5: „*If a safety goal can be achieved by transitioning to, or by maintaining, one or more safe states, then the corresponding safe state(s) shall be specified.*"

Clause 7.4.4.6: „*The safety goals together with their attributes (ASIL) shall be specified in accordance with ISO 26262-8:2011, Clause 6.*"

Recommendation: Although the ISO requirement states that a safety goal shall be formulated for each hazardous event, in most (and possibly all) cases it makes more sense to formulate one safety goal for each hazard. In fact, a typical safety goal is simply a statement that a given hazard shall not occur. Together with the ASIL determined for the corresponding hazard, a safety goal constitutes a top-level requirement in the functional safety hierarchy. Thus, each safety goal and its associated ASIL should be represented in the requirements model if such a model is indeed

created. This could be further supported by a requirements modeling checklist that explicitly states that safety goals and associated ASILs shall be represented in the requirements model and that these shall be identifiable as safety goals in this model.

Regarding the details of the ISO requirement, the following can be noted:

- Clause 7.4.4.5 is quite unnecessary here from a strictly logical viewpoint. It should be considered in the functional safety concept and not in the safety goal formulation. However, if compliance with ISO 26262 is an absolute requirement associating a safe state with each safety goal (when applicable) is not a difficult task.
- Clause 7.4.4.6 deals with requirements management and is addressed elsewhere in these guidelines.

Derived Requirements:

Each Hazardous Event shall have one associated Safety Goal.
There shall be one safe state defined using the safe state attribute of the Safety Goal.

8.2 Analysis Level Modeling

8.2.1 Part 3: Concept phase, Clause 8.4.2.2

„*At least one functional safety requirement shall be specified for each safety goal.*

NOTE: One functional safety requirement can be valid for several safety goals."

Recommendation: This ISO requirement does not directly concern modeling but it could be addressed in a checklist for the requirements model if such a model is indeed created in a given project. The checklist would then include a formulation along the lines of "is every safety goal linked to at least one functional safety requirement?"

Furthermore, the functional safety requirements (as defined in ISO 26262) should be identifiable as functional safety requirements in the requirements model.

Derived Requirements:

One or several requirements shall be defined for each Safety Goal and be associated to a FunctionalSafetyConcept requirements container with role functional safety requirement.

8.2.2 Part 3: Concept phase, Clause 8.4.2.3

„*Each functional safety requirement shall be specified by considering the following, if applicable:*

a) operating modes;

b) fault tolerant time interval;

c) safe states,

d) emergency operation interval, and

e) functional redundancies (e.g. fault tolerance)."

Recommendation: This ISO requirement does not directly concern modeling but it could be addressed in a checklist for the requirements model if such a model is indeed created in a given project. The checklist would then include a formulation along the lines of "has the following issues a-e been considered in the specification of each functional safety requirement?"

Derived Requirements:

For each functional safety requirement, the following information shall be defined, where applicable:

a) Operating modes - defined as associated modes indicating when the functional safety requirement is valid,

b) Fault tolerant time interval - defined in the requirement text or as a derived requirement,

c) Safe states - defined in the requirement text or as a derived requirement,

d) Emergency operation interval - defined in the requirement text or as a derived requirement,

e) Functional redundancies (e.g. fault tolerance) - defined in the requirement text or as a derived requirement.

8.2.3 Part 3: Concept phase, Clauses 8.4.2.4 – 8.4.2.6

These ISO 26262 requirements specify some aspects that should be covered in the technical safety requirements: emergency operation, warning and degradation concept, assumptions on the actions of the driver or other involved people.

Recommendation: The ISO requirements can easily be translated into specific questions in a requirements management checklist: "Has the warning and degradation concept been specified?", etc. This checklist can be applied to the requirements model in a project to check whether the ISO requirements are met or not.

Derived Requirements:

A warning and back-up concept shall be specified using architectural elements on Analysis Level.

An emergency operation shall be specified using architectural elements on Analysis Level, unless a safe state can be reached by immediate switching off.

Assumptions made on the necessary actions of the driver or other endangered persons in order to comply with the safety goals shall be represented as requirements on the Environment model, and possibly also behavioral models.

8.2.4 Part 3: Concept phase, Clause 8.4.3.1

„*The functional safety requirements shall be allocated to the elements of the preliminary architectural assumptions:*

NOTE Redundancy and independence issues can be checked by an analysis of dependent failures (see ISO 26262-9:2011, Clause 7).

a) During the course of allocation, the ASIL and information given in 8.4.2.3 shall be inherited from the associated safety goal or, if ASIL decomposition is applied, from the level above.

b) If several functional safety requirements are allocated to the same architectural element, then the architectural element shall be developed in accordance with the highest ASIL for those safety requirements if independence or freedom from interference cannot be argued in the preliminary architecture.

c) If the item comprises more than one system, then the functional safety requirements for the individual systems and their interfaces shall be specified, considering the preliminary architectural assumptions. These functional safety requirements shall be allocated to the systems.

d) *If ASIL decomposition is applied during the allocation of the functional safety requirements, then it shall be applied in accordance with ISO 26262-9:2011, Clause 5.*"

Recommendation: The allocation of functional safety requirements should be visible in the modeling at the analysis level. This could be highlighted in a checklist for the analysis modeling. ASIL issues should be handled as indicated in the ISO requirement and this could also be highlighted in a modeling checklist.

(A detailed guideline for how to address ASIL issues as described above could be defined, but this is not done in this report since such a guideline would depend on)

Derived Requirements:

Functional Safety Requirements, i.e. Requirements in the Functional Safety Concept shall be associated to elements on Analysis level through the Satisfy association.

The ASIL of a Functional Safety Requirement shall be defined using the ASIL attribute of a SafetyConstraint associated to the requirement with a Refine relationship.

Each SafetyConstraint shall be associated to a FaultFailure. The FaultFailure defines the failure mode which is to be avoided at the integrity level according to the SafetyConstraint's ASIL attribute.

8.2.5 Part 3: Concept phase, Clause 8.4.3.2

„*If the functional safety concept is to rely on elements of other technologies, then the following shall apply:*

a) The functional safety requirements implemented by elements of other technologies shall be derived and allocated to the corresponding elements of the architecture.

b) The functional safety requirements relating to the interfaces with elements of other technologies shall be specified.

c) The implementation of functional safety requirements by elements of other technologies shall

be ensured through specific measures that are outside the scope of ISO 26262.

d) No ASIL should be assigned to these elements."

Recommendation: The safety architecture concept represents the conceptual architecture of the system in terms of architectural provisions to ensure functional safety. Thus, the safety architecture concept includes redundancy principles such as replication of components (for example more than one sensor to measure a physical quantity), monitoring of a system element by another system element, activation of an actuator only when two system elements agree that such an activation shall be made, etc. The safety architecture concept can be represented by one or more block diagrams that show the redundancy principles. For the modeling, a checklist can be defined that includes the simple question "is the safety architecture concept represented in a model?"

Derived Requirements:

A safety architecture concept shall be specified using architectural elements on Analysis Level.

8.2.6 Part 3: Concept phase, Clause 8.4.4.1

„*The acceptance criteria for safety validation of the item shall be specified based on the functional safety requirements.*"

Recommendation: Assuming that validation is somehow represented by models, the acceptance criteria should be represented in such models. A checklist for such modeling can include this issue to aid the modeler.

Derived Requirements:

Each Functional Safety Requirement shall be linked to a VVProcedure with a Verify association.

Each Functional Safety Requirement shall have an acceptance criteria specified as a VvIntendedOutcome of the Requirement's VVProcedure.

8.2.7 Part 3: Concept phase, Clause 8.4.5.1

„*The functional safety concept shall be verified in accordance with ISO 26262-8:2011, Clause 9, to show*

a) its consistency and compliance with the safety goals; and

b) its ability to mitigate or avoid the hazardous events."

Recommendation: A checklist for the requirements modeling should include the need for verification of the compliance between functional safety requirements and safety goals, with explicit mentioning of applicable verification techniques like inspection, walkthrough and formal methods.

Derived Requirements:

Checklist.

8.3 Design Level Modeling

8.3.1 Part 4: Product development at the system level, Clause 6.4.1.4

„*The technical safety requirements shall specify safety-related dependencies between systems or item elements and between the item and other system.*"

Recommendation: The requirement seems to be associated with design level modeling and may possibly be relevant for modeling.

Derived Requirements:

Dependencies between different parts of the functional design architecture shall be represented by the interface definitions.

8.3.2 Part 4: Product development at the system level, Clause 6.4.2.2

„*The technical safety requirements shall specify the necessary safety mechanisms (see ISO 26262-8:2011, Clause 6) including:*

a) the measures relating to the detection, indication and control of faults in the system itself;

b) the measures relating to the detection, indication and control of faults in external devices that interact with the system;

c) the measures that enable the system to achieve or maintain a safe state;

d) the measures to detail and implement the warning and degradation concept; and

e) the measures which prevent faults from being latent [see 6.4.4 (Avoidance of latent faults)]."

Recommendation: This ISO requirement does not directly concern modeling but it could be addressed in a checklist for the requirements model if such a model is indeed created in a given project. The checklist would then include formulations like "Have technical safety requirements concerning measures related to... been specified?" (See a-e in the ISO requirement.)

Furthermore, the technical safety requirements (as defined in ISO 26262) should be identifiable as technical safety requirements in the requirements model.

Derived Requirements:

Technical Safety Requirements shall be defined as requirements that are associated to a TechnicalSafetyConcept requirements container with role technical safety requirement.

8.3.3 Part 4: Product development at the system level, Clause 6.4.2.3

„*For each safety mechanism that enables an item to achieve or maintain a safe state the following shall be specified:*

a) the transition to the safe state;

b) the fault tolerant time interval;

c) the emergency operation interval, if the safe state cannot be reached immediately; and

d) the measures to maintain the safe state."

Recommendation: This requirement should be represented in a checklist to be used in the design level modeling ("For each safety mechanism represented in a design model, have the following been specified?...").

Derived Requirements:

Checklist.

8.3.4 Part 4: Product development at the system level, Clause 6.4.4.1

„*This requirement applies to ASILs (A), (B), C, and D, in accordance with 4.3: if applicable, safety mechanisms shall be specified to prevent faults from being latent.*"

Recommendation: This requirement should be represented in a checklist to be used in the design level modeling ("Are safety mechanisms for prevention of latent faults part of the design? ")

Derived Requirements:

Checklist.

8.3.5 Part 4: Product development at the system level, Clause 6.4.6.1

„*The technical safety requirements shall be verified in accordance with ISO 26262-8:2011, Clause 9, to provide evidence for their:*

a) compliance and consistency with the functional safety concept; and

b) compliance with the preliminary architectural design assumptions."

Recommendation: A checklist for the requirements modeling should include the need for verification of the technical safety requirements with respect to consistency with the functional safety concept and the preliminary architectural design, with explicit mentioning of applicable verification techniques like inspection, walkthrough and formal methods.

Derived Requirements:

Requirements in a TechnicalSafetyConcept shall be derived from Requirements in a FunctionalSafetyConcept and linked with a Derived association.

A TechnicalSafetyConcept shall be defined in the FunctionalDesignArchitecture to realize the

FunctionalSafetyConcept on Analysis Level.

Architectural elements in a TechnicalSafetyConcept shall be linked to elements in the corresponding FunctionalSafetyConcept with a realize relation.

8.3.6 Part 4: Product development at the system level, Clause 6.4.6.2

„*The criteria for safety validation of the item shall be refined based on the technical safety requirements.*"

Recommendation: Assuming that validation is somehow represented by models, the acceptance criteria should be represented in such models. A checklist for such modeling can include this issue to aid the modeler.

Derived Requirements:

Each Technical Safety Requirement shall be linked to a VVProcedure with a Verify association.

Each Technical Safety Requirement shall have an acceptance criteria specified as a VvIntendedOutcome of the Requirement's VVProcedure.

8.3.7 Part 4: Product development at the system level, Clause 7.4.2.3

„*If an element is comprised of sub-elements with different ASILs assigned, or of non-safety-related sub-elements and safety-related sub-elements, then each of these shall be treated in accordance with the highest ASIL, unless the criteria for coexistence, in accordance with ISO 26262-9:2011, Clause 6, are met.*"

Recommendation: The ASIL assigned to a certain requirement shall propagate to the architectural elements to which this requirement applies in such a way that each element is assigned the highest ASIL of all the requirements that apply to the element. In the modeling, it shall be possible to associate ASILs with system elements and the modeler should check that the ASILs are inherited in the way defined in the standard. The ASIL inheritance rules of ISO 26262 can be represented in a checklist for the modeling. (Note that in some cases a lower ASIL can be assigned to a sub-element in accordance with the "criteria for coexistence of elements" section in Part 9 of ISO 26262).

Derived Requirements:

Each Technical Safety Requirement shall be associated to a SafetyConstraint using the Refine relation. Each SafetyConstraint shall define the ASIL level and define the exact failure mode to avoid using the FaultFailure element.

A Technical Safety Requirement derived from a Functional Safety Requirement shall have the same or higher ASIL as the Functional Safety Requirement. Alternatively, ASIL decomposition can be applied such that the Technical Safety Concept meets the Functional Safety Requirement at the required ASIL using redundancy.

8.3.8 Part 4: Product development at the system level, Clause 7.4.2.4

„*Internal and external interfaces of safety-related elements shall be defined, in order to avoid other elements having adverse safety-related effects on the safety-related elements.*"

Recommendation: All interfaces of all design shall be precisely defined in the design models. This can be explicitly addressed in a design model checklist.

Derived Requirements:

Interfaces of safety-related elements shall be defined using ports and datatypes.

8.3.9 Part 4: Product development at the system level, Clause 7.4.4.1

„*Measures for detection and control, or mitigation of random hardware failures shall be specified with respect to the system design given in 7.4.1 (System design specification and technical safety concept).*"

Recommendation: Mechanisms for error detection and error handling should be represented in the models. This can be explicitly addressed in a design model checklist.

Derived Requirements:

Functions in the FunctionalDesignArchitecture shall be allocated to Nodes in the HardwareArchitecture using the Allocation association.

Hardware-dependent error detection and control functions shall be defined as BasicSoftwareFunctionType or DesignFunctionType allocated to the concerned Node.

Checklist.

8.3.10 Part 4: Product development at the system level, Clause 7.4.4.3

„*This requirement applies to ASILs (B), C, and D, in accordance with 4.3: one of the alternative procedures of evaluation of violation of the safety goal due to random hardware failures (see ISO 26262-5:2011, Clause 9) shall be chosen and the target values shall be specified for final evaluation at item level (see requirement 9.4.3.3).*"

Recommendation: This ISO requirement does not directly concern modeling but it could be addressed in a checklist for the requirements model if such a model is indeed created in a given project. The checklist would then include a formulation like "Have target values for the probability of safety goal violations been defined in the requirements model?"

Derived Requirements:

Each SafetyGoal shall be associated using Verify relation to a VVProcedure establishing the

probability of violation of the safety goal. The VvIntendedOutcome of the VVProcedure shall define the assessment criteria for the probability of violation of the safety goal

8.3.11 Part 4: Product development at the system level, Clause 7.4.6.1

„*The HSI specification shall specify the hardware and software interaction and be consistent with the technical safety concept. The HSI specification shall include the component's hardware devices that are controlled by software and hardware resources that support the execution of software.*"

Recommendation: For hardware that is controlled by software and hardware that supports the execution of software, the hardware-software interface shall be represented in the models at design level and/or possibly at the implementation level. This can be explicitly addressed in a design model checklist.

Derived Requirements:

(Detailed HSI aspects are the concern of Implementation level.)

The functionality of hardware components in a technical safety concept shall be defined using HardwareFunctionType.

(duplex) Functions in the FunctionalDesignArchitecture shall be allocated to Nodes in the HardwareArchitecture using the Allocation association.

The functional hardware-software interface shall be defined using the ports of HardwareFunctionTypes.

The non-functional hardware-software interface aspects shall be defined using requirements on the Hardware Architecture elements.

8.3.12 Part 4: Product development at the system level, Clause 7.4.6.2

„*The HSI specification shall include the following characteristics:*

a) the relevant operating modes of hardware devices and the relevant configuration parameters;

b) the hardware features that ensure the independence between elements and that support software partitioning;

c) shared and exclusive use of hardware resources;

d) the access mechanism to hardware devices; and

e) the timing constraints defined for each service involved in the technical safety concept."

Recommendation: This ISO requirement can be represented in a checklist for the design level modeling ("Have the following characteristics been considered in the hardware/software interface specification?...").

When applicable, timing aspects should be accounted for in the modeling. These aspects include the timing constraints related to the performance of hardware parts. The timing constraints shall be checked for compliance with respect to the technical safety requirements. This recommenda-

tion could be implemented in a checklist to be used during implementation-level modeling.

Note: The results of the TIMMO project (http://www.timmo.org) are expected to be relevant for this issue. However, this has not been investigated in the creation of this guidelines document.

Derived Requirements:

(Duplex) The non-functional hardware-software interface aspects shall be defined using requirements on the Hardware Architecture elements.

Safety-relevant Timing Requirements shall be defined using a Requirement with both a Timing Constraint and a SafetyConstraint associated using a Refine relation.

8.3.13 Part 4: Product development at the system level, Clause 7.4.6.3

„*The relevant diagnostic capabilities of the hardware and their use by the software shall be specified in the HSI specification:*

a) the hardware diagnostic features shall be defined; and

b) the diagnostic features concerning the hardware, to be implemented in software, shall be defined."

Recommendation: This ISO requirement can be represented in a checklist for the design level modeling ("Have any inbuilt diagnostic features within the hardware components been addressed in the design level modeling?").

Derived Requirements:

(Duplex) The non-functional hardware-software interface aspects shall be defined using requirements on the Hardware Architecture elements.

(Duplex) The functional hardware-software interface shall be defined using the ports of HardwareFunctionTypes.

8.3.14 Part 4: Product development at the system level, Clause 7.4.8.1

„*The system design shall be verified for compliance and completeness with regard to the technical safety concept using the verification methods listed in Table 3.*"

Recommendation: A checklist for the design level modeling should include the need to verify that design is compliant with the technical safety concept. Appropriate methods should be given in the checklist, such as inspection, walkthrough, simulation, prototyping and analysis.

Derived Requirements:

Checklist.

Each technical safety requirement shall have a VVProcedure which shall be used to verify the requirement.

8.4 Implementation Level Modeling

8.4.1 Part 5: Product development at the hardware level, Clause 6.4.2

„*The hardware safety requirements specification shall include each hardware requirement that relates to safety, including the following:*

NOTE 1 The hardware safety requirements described in bullets a), b), c), or d) include the attributes needed to ensure the effectiveness of the above safety mechanisms.

a) the hardware safety requirements and relevant attributes of safety mechanisms to control internal failures of the hardware of the element, this includes internal safety mechanisms to cover transient faults when shown to be relevant due, for instance, to the technology used;

b) the hardware safety requirements and relevant attributes of safety mechanisms to ensure the element is tolerant to failures external to the element;

c) the hardware safety requirements and relevant attributes of safety mechanisms to comply with the safety requirements of other elements;

d) the hardware safety requirements and relevant attributes of safety mechanisms to detect and signal internal or external failures; and

e) the hardware safety requirements not specifying safety mechanisms."

Recommendation: This ISO requirement does not directly concern modeling but it could be addressed in a checklist for the requirements model if such a model is indeed created in a given project. The checklist would then include formulations like "Have hardware safety requirements related to... been specified?" (See a-e in the ISO requirement.)

Furthermore, the hardware safety requirements (as defined in ISO 26262) should be identifiable as hardware safety requirements in the requirements model and each hardware safety requirement should be assigned an ASIL in the requirement model.

Derived Requirements:

Hardware safety requirements shall be defined as a Requirement with an associated Safety Constraint and associated to AUTOSAR hardware elements with a Satisfy relation.

8.4.2 Part 5: Product development at the hardware level, Clause 6.4.6

„*The criteria for design verification of the hardware of the item or element shall be specified, including environmental conditions (temperature, vibration, EMI, etc.), specific operational environment (supply voltage, mission profile, etc.) and component specific requirements:*

a) for verification by qualification for hardware components or part of intermediate complexity, the criteria shall meet the needs of ISO 26262-8:2011, Clause 13, and

b) for verification by testing, the criteria shall meet the needs of Clause 10."

Recommendation: Assuming that verification of the hardware design is somehow represented by models, the acceptance criteria for such verification should be represented in these models. A checklist for such modeling can include this issue to aid the modeler.

Derived Requirements:

Each hardware safety requirement shall be linked to a VVProcedure with a Verify association.

Each hardware safety requirement shall have an acceptance criteria specified as a VvIntendedOutcome of the Requirement's VVProcedure.

8.4.3 Part 5: Product development at the hardware level, Clause 6.4.10

„*The HSI specification initiated in ISO 26262-4:2011, Clause 7, shall be refined sufficiently to allow for the correct control and usage of the hardware by the software, and shall describe each safety-related dependency between hardware and software.*"

Recommendation: For hardware that is controlled by software and hardware that supports the execution of software, the hardware-software interface should be represented in the models at the implementation level. This can be explicitly addressed in a checklist for the implementation level.

Derived Requirements:

(duplicate) Hardware safety requirements shall be defined as a Requirement with an associated SafetyConstraint and associated to AUTOSAR hardware elements with a Satisfy relation.

Checklist.

8.4.4 Part 5: Product development at the hardware level, Clause 7.4.1.1

„*The hardware architecture shall implement the hardware safety requirements defined in Clause 6.*"

Recommendation: The hardware architecture model shall be consistent with the hardware safety requirements. This (obvious) requirement could be highlighted in a checklist for the implementation level modeling.

Derived Requirements:

The hardware architecture on Implementation Level shall be defined using AUTOSAR hardware elements that are linked to their corresponding elements on Design Level with a Realize association.

8.4.5 Part 5: Product development at the hardware level, Clause 7.4.1.4

„*If a hardware element is made of sub-elements that have different ASILs assigned, or sub-elements that have no ASIL assigned and safety-related sub-elements, then each of these shall be treated in accordance with the highest ASIL, unless the criteria for coexistence in accordance*

with ISO 26262-9 are met."

Recommendation: In the hardware architecture model, ASILs should be associated to elements and sub-elements in accordance with the ISO 26262 requirements. The basic rule is that en element shall be assigned the highest ASIL of all the hardware safety requirements assigned to the element. However, if the "criteria for coexistence" in Part 9 of ISO 26262 are fulfilled, some sub-elements within the element can sometimes be assigned lower ASILs.

Derived Requirements:

(duplicate) Hardware safety requirements shall be defined as a Requirement with an associated SafetyConstraint and associated to AUTOSAR hardware elements with a Satisfy relation.

Checklist.

8.4.6 Part 5: Product development at the hardware level, Clause 7.4.1.5

„*The traceability between the hardware safety requirements and their implementation shall be maintained down to the lowest level of hardware components.*"

Recommendation: The hardware architecture models should contain traceability-related information so that tracing between hardware safety requirements and corresponding architectural elements and solutions is possible. A checklist to be used in the modeling could highlight this: "Are traceability links established between requirements and implementation?"

Derived Requirements:

(duplicate) Hardware safety requirements shall be defined as a Requirement with an associated SafetyConstraint and associated to AUTOSAR hardware elements with a Satisfy relation.

Checklist.

8.4.7 Part 5: Product development at the hardware level, Clause 9.4.1

„*This requirement applies to ASIL (B), C and D of the safety goal. The item shall comply with either 9.4.2 or 9.4.3. "*

Recommendation: If requirements are modeled, the probability of a violation of each safety goal due to random hardware faults should be addressed in the requirements model. A choice should then be made about whether these requirements shall be in the form of required quantitative probabilities at the item level or in the form of (semi-qualitative) probabilities of each potential cause of an item-level safety goal violation.

8.4.8 Part 5: Product development at the hardware level, Clause 9.4.2.2

„*This requirement applies to ASIL (B), C, and D of the safety goal. Quantitative target values of requirement 9.4.2.1 shall be expressed in terms of average probability per hour over the operational lifetime of the item.*"

Recommendation: If target values for the probability of violation of a safety goal due to random hardware faults are specified, they should be expressed as average probability per hour over the operational lifetime of the item.

8.4.9 Part 5: Product development at the hardware level, Clause 9.4.3.3

„*This requirement applies to ASIL (B), C, and D of the safety goal. The failure rate class ranking for a hardware part failure rate shall be determined as follows:*

a) the failure rate corresponding to failure rate class 1 shall be less than the target for ASIL D divided by 100; unless 9.4.3.4 is applied;

b) the failure rate corresponding to failure rate class 2 shall be less than or equal to 10 times the failure rate corresponding to failure rate class 1;

c) the failure rate corresponding to failure rate class 3 shall be less than or equal to 100 times the failure rate corresponding to failure rate class 1; and

d) the failure rate corresponding to failure rate class i, i > 3 shall be less than or equal to $10^{(i-1)}$ times the failure rate corresponding to failure rate class 1."

Recommendation: If violations of safety goals due to random hardware faults are addressed in the form of (semi-qualitative) probabilities of each potential cause of such violations, failure rate classes as defined in the ISO 26262 clause above should be used.

8.4.10 Part 6: Product development at the software level, Clause 6.4.1

„*The software safety requirements shall address each software-based function whose failure could lead to a violation of a technical safety requirement allocated to software...*"

Recommendation: This ISO requirement does not directly concern modeling but it could be addressed in a checklist for the requirements model if such a model is indeed created in a given project: "Do the software safety requirements address each software-based function whose failure could lead to a violation of a technical safety requirement allocated to software?"

8.4.11 Part 6: Product development at the software level, Clause 6.4.2

„*The specification of the software safety requirements shall be derived from the technical safety concept and the system design in accordance with ISO 26262-4:2011, 7.4.1 and 7.4.5, and shall consider:*

a) the specification and management of safety requirements in accordance with ISO 26262-8:2011, Clause 6;

b) the specified system and hardware configurations;

c) the hardware-software interface specification;

d) the relevant requirements of the hardware design specification;

e) the timing constraints;

f) the external interfaces; and

g) each operating mode of the vehicle, the system, or the hardware, having an impact on the software."

Recommendation: This ISO requirement does not directly concern modeling but it could be addressed in a checklist for the requirements model if such a model is indeed created in a given project. The checklist would then include formulations like

- "Have software safety requirements been derived from the system design specification?"
- "Are the software safety requirements complete and consistent"?
- "Have the following been considered in the specification of software safety requirements?..." (see b – g in the ISO requirement above)

Furthermore, the software safety requirements should be identifiable as software safety requirements in the requirements model and each software safety requirement should be assigned an ASIL in the requirement model.

8.4.12 Part 6: Product development at the software level, Clause 6.4.8

„*The software safety requirements and the refined hardware-software interface requirements shall be verified in accordance with ISO 26262-8:2011, Clauses 6 and 9, to show their:*

a) compliance and consistency with the technical safety requirements;

b) compliance with the system design; and

c) consistency with the hardware-software interface."

Recommendation: A checklist for the requirements modeling should include the need for verification of the software safety requirements with respect to compliance with the functional safety concept and the system design specification, consistency with hardware safety requirements, correct allocation of ASIL, and completeness with regard to the technical safety requirements allocated to software. The checklist could explicitly mention suitable verification techniques (as given in the tables referenced by the ISO requirement above): Inspection, Walkthrough, Semi-formal verification, Formal verification.

8.4.13 Part 6: Product development at the software level, Clause 7.4.3

„*In order to avoid failures resulting from high complexity, the software architectural design shall exhibit the following properties by use of the principles listed in Table 3:*

a) modularity;

b) encapsulation; and

c) simplicity."

Recommendation: Depending on the ASIL, the software architecture should be described using an informal or semi-formal (or formal) notation. For the lower ASILs (ASIL A and ASIL B), informal notation is considered sufficient but for the higher ASILs (ASIL C and ASIL D), at least a semi-formal notation should be used. This requirement could be highlighted in a checklist for the software architecture modeling.

8.4.14 Part 6: Product development at the software level, Clause 7.4.6

„*Every safety-related software component shall be categorized as one of the following:*

a) newly developed;

b) reused with modifications; or

c) reused without modifications."

Recommendation: For each software component, the software architecture model should include information about the component's origin: newly developed, reused with modification, reused without modification, or COTS (Commercial Off-The-Shelf). Like most of the recommendations in this document, this recommendation can be represented by an entry in a checklist for the modeling: "Has every software been categorised as...?"

8.4.15 Part 6: Product development at the software level, Clause 7.4.9

„*The software safety requirements shall be allocated to the software components…*"

Recommendation: The allocation of software safety requirements to software components shall be represented in the software architecture model and every defined software safety requirement shall be allocated to at least one software component. A checklist for the software architecture modeling could include these issues.

8.4.16 Part 6: Product development at the software level, Clause 7.4.11c

„*…the part of the software that implements the software partitioning is developed in compliance with the same or an ASIL higher than the highest ASIL assigned to the requirements of the software partitions…*"

Recommendation: An ASIL shall be associated with that part of the software that implements software partitioning. This ASIL shall equal the highest ASIL among the software partitions that are protected by this partitioning software. This recommendation can be highlighted by including it in a checklist for the implementation level modeling.

8.4.17 Part 6: Product development at the software level, Clause 7.4.17

„*An upper estimation of required resources for the embedded software shall be made, including:*

a) the execution time;

b) the storage space; and

c) the communication resources."

Recommendation: When appropriate, information about required resources (execution time, storage space, communication resources, etc) shall be represented in the software architecture model. This requirement may be highlighted in a checklist for the software architecture modeling.

8.4.18 Part 6: Product development at the software level, Clause 7.4.18

„*The software architectural design shall be verified in accordance with ISO 26262-8:2011, Clause 9, and by using the software architectural design verification methods listed in Table 6 to demonstrate the following properties:*

a) compliance with the software safety requirements;

b) compatibility with the target hardware; and

c) adherence to design guidelines."

Recommendation: The software architecture model should be verified with respect to compliance with software safety requirements, compatibility with target hardware and adherence to any applicable design guidelines. Possible techniques for this are walkthrough, inspection, simulation, prototype generation/animation, formal verification, control flow analysis and data flow analysis. See the corresponding Tables in ISO 26262 for which technique, or combination of techniques, to use for a particular ASIL.

This recommendation could be represented in a checklist for the software architecture modeling.

8.4.19 Part 6: Product development at the software level, Clause 8.4.2

„*To ensure that the software unit design captures the information necessary to allow the subsequent development activities to be performed correctly and effectively, the software unit design shall be described using the notations listed in Table 7.*"

Recommendation: Depending on the ASIL, the software unit design should be described in natural language and also in an informal or semi-formal (or formal) notation. For ASIL A, informal notation is considered sufficient but for the higher ASILs (ASIL C and ASIL D), at least a semi-formal notation should be used. See the corresponding Table in ISO 26262 for more detailed information about which technique - or combination of techniques - to use for a particular ASIL. This requirement could be highlighted in a checklist for the software architecture modeling.

8.4.20 Part 6: Product development at the software level, Clause 8.4.5

„*The software unit design and implementation shall be verified in accordance with ISO 26262-8:2011 Clause 9, and by applying the verification methods listed in Table 9, to demonstrate:*

a) the compliance with the hardware-software interface specification (in accordance with ISO 26262-5:2011, 6.4.10);

b) the fulfillment of the software safety requirements as allocated to the software units (in accordance with 7.4.9) through traceability;

c) the compliance of the source code with its design specification;

d) the compliance of the source code with the coding guidelines (see 5.5.3); and

e) the compatibility of the software unit implementations with the target hardware."

Recommendation: The software unit design should be verified with respect to compliance with any applicable requirements concerning the software unit design process (for example as defined in ISO 26262), compliance with the hardware/software interface, and completeness regarding both software safety requirements and the software architecture.

Possible techniques for this are inspection and walkthrough of a model of a software unit, semi-formal verification, formal verification, control flow analysis and data flow analysis. See the corresponding Tables in ISO 26262 for which technique, or combination of techniques, to use for a particular ASIL.

This recommendation could be represented in a checklist for the software unit modeling.

8.4.21 Part 6: Product development at the software level, Clause C.4.1

„*The configuration data shall be specified to ensure the correct usage of the configurable software during the safety lifecycle. This shall include:*

a) the valid values of the configuration data;

b) the intent and usage of the configuration data;

c) the range, scaling, units; and

d) the interdependencies between different elements of the configuration data."

Recommendation: Representation of configuration data in implementation models should be such that it supports the deployment of the configurable software. The following aspects should be represented within the model, when applicable:

- Valid values of the configuration data
- Intent and usage of the configuration data
- Range, scaling, units
- Interdependencies between different elements of the configuration data

This recommendation could be included in a checklist to be used in the implementation-level modeling.

8.4.22 Part 6: Product development at the software level, Clause C.4.2

„*Verification of the configuration data shall be performed to ensure:*

a) the use of values within their specified range; and

b) the compatibility with the other configuration data."

Recommendation: The specific values of the configuration data for an intended use should be verified with respect to being in the valid range and being compatible with other configuration data. This recommendation could be included in a checklist to be used in the implementation-level modeling.

8.4.23 Part 6: Product development at the software level, Clause C.4.3

„*The ASIL of the configuration data shall equal the highest ASIL of the configurable software by which it is used.*"

Recommendation: There shall be provisions for associating an ASIL with the configuration data of any configurable piece of software. This ASIL shall equal the maximum ASIL of those safety requirements that might be violated by the configuration data if this data is incorrect. This recommendation can be highlighted by including it in a checklist for the implementation level modeling.

8.4.24 Part 6: Product development at the software level, Clause C.4.6

„*The calibration data associated with software components shall be specified to ensure the correct operation and expected performance of the configured software. This shall include:*

a) the valid values of the calibration data;

b) the intent and usage of the calibration data;

c) the range, scaling and units, if applicable, with their dependence on the operating state;

d) the known interdependencies between different calibration data; and

e) the known interdependencies between configuration data and calibration data."

Recommendation: Representation of calibration data in implementation models should be such that it supports the achievement of correct operation of the software. The following aspects should be represented within the model, when applicable:

- Valid values of the calibration data
- Intent and usage of the calibration data
- Range, scaling, units
- Interdependencies between different calibration data within one calibration set
- Known interdependencies between configuration data and calibration data

This recommendation could be included in a checklist to be used in the implementation-level modeling.

8.4.25 Part 6: Product development at the software level, Clause C.4.7

„*The verification of the calibration data shall be planned, specified and executed in accordance with ISO 26262-8:2011, Clause 9. The verification of calibration data shall examine whether the calibration data is within its specified boundaries.*"

Recommendation: The specific values of the calibration data for an intended use should be verified with respect to being in the valid range. This recommendation could be included in a checklist to be used in the implementation-level modeling.

8.4.26 Part 6: Product development at the software level, Clause C.4.8

„*The ASIL of the calibration data shall equal the highest ASIL of the software safety requirements it can violate.*"

Recommendation: There shall be provisions for associating an ASIL with any set of calibration data. This ASIL shall equal the maximum ASIL of those safety requirements that might be violated by the calibration data if this data is incorrect. This recommendation can be highlighted by including it in a checklist for the implementation level modeling.

8.4.27 Part 6: Product development at the software level, Clause C.4.9

„*To detect unintended changes of safety-related calibration data, mechanisms for the detection of unintended changes of data as listed in Table C.1 shall be applied.*"

Recommendation: Calibration data should be checked at run-time (continuously or only during power-up) by an appropriate mechanism or set of mechanisms. Examples of mechanisms are plausibility checks, redundant storage and error detection codes. Of these three, plausibility checks should be the primary mechanism (according to ISO 26262 at least, but this could be debated.) This recommendation can be highlighted by including it in a checklist for the implementation level modeling.

8.5 Orthogonal Issues, applicable to all Modeling Levels

8.5.1 Part 2: Management of functional safety, Clause 6.4.6.2

„*The safety case should progressively compile the work products that are generated during the safety lifecycle.*"

Recommendation: A safety case is an argumentation of why a system is adequately safe. If this safety case is represented in a model, for example a GSN (Goal Structuring Notation) model, it should be ensured that all work products generated during the safety lifecycle (e.g. confirmation measures) are included in the model. This can be addressed in a checklist for safety case modeling, with the ISO 26262 requirement as stated above included in the checklist. (The confirmation measures are the audits, reviews and functional safety assessments described in Part 2 of ISO 26262.)

8.5.2 Part 8: Supporting processes, Clause 6.4.1.1

„*To achieve the characteristics of safety requirements listed in 6.4.2.4, safety requirements shall be specified by an appropriate combination of:*

a) natural language, and

b) methods listed in Table 1."

Recommendation: Safety requirements should be specified using natural language and an appropriate combination of informal and semi-formal (or even fully formal) notations. Informal notation is considered sufficient, together with natural language, for ASILs A-B. For higher ASILs, a combination of natural language and semi-formal notation is considered sufficient. This recommendation can be highlighted by including it in a checklist for the requirements modeling.

8.5.3 Part 8: Supporting processes, Clause 6.4.2.1

„*Safety requirements shall be unambiguously identifiable as safety requirements.*"

Recommendation: Safety requirements should be clearly identifiable as being safety requirements. Thus, in a requirements model, the safety requirements should have some specific "tag" or other special characteristic that differentiates them from other requirements. To aid the modeler, a checklist for the requirements modeling could include the following entry: "Have all safety requirements been labeled as safety-critical in the model?"

8.5.4 Part 8: Supporting processes, Clause 6.4.2.4

„*Safety requirements shall have the following characteristics:*

a) unambiguous and comprehensible,

b) atomic,

c) internally consistent,

d) feasible, and

e) verifiable."

Recommendation: Every safety requirement should be unambiguous, comprehensible, atomic, internally consistent (i.e. the requirement should not contradict itself), feasible and verifiable. These characteristics of the safety requirements can be listed in a checklist.

8.5.5 Part 8: Supporting processes, Clause 6.4.2.5

„*Safety requirements shall have the following attributes:*

a) a unique identification remaining unchanged throughout the safety lifecycle,

b) a status, and

c) an ASIL."

Recommendation: Every safety requirement should have a unique and constant identification, a status and an ASIL. If requirements models are used, these characteristics of the safety requirements should be possible to represent in the models. Furthermore, the characteristics can be listed in a checklist.

8.5.6 Part 8: Supporting processes, Clause 6.4.3.1

„*The set of safety requirements shall have the following properties:*

a) hierarchical structure,

b) organizational structure according to an appropriate grouping scheme,

c) completeness,

d) external consistency,

e) no duplication of information within any level of the hierarchical structure, and

f) maintainability."

Recommendation: The complete set of safety requirements should be hierarchical, organized, complete and consistent (i.e. requirements should not contradict each other). These characteristics of the safety requirements can be listed in a checklist for the safety requirements.

8.5.7 Part 8: Supporting processes, Clause 6.4.3.2

„*Safety requirements shall be traceable with a reference being made to:*

a) each source of a safety requirement at the upper hierarchical level,

b) each derived safety requirement at a lower hierarchical level, or to its realisation in the design, and

c) the specification of verification in accordance with 9.4.2."

Recommendation: Every safety requirement should be associated with traceability information concerning the sources at the next higher hierarchical level from which the requirement has been derived. For example, a technical safety requirement shall be linked to the functional safety requirements from which it has been derived. Similarly, links to the next lower hierarchical level (derived safety requirements or implementation) shall be established. Furthermore, traceability links shall be established to the verification procedures where the fulfillment of the requirement is checked.

This need for traceability from any requirement to related higher and lower requirements and to the verification procedures can be addressed in a checklist for the requirements modeling.

8.5.8 Part 9: ASIL-oriented and safety-oriented analyses, Clause 5.4.7

„*If ASIL decomposition of an initial safety requirement results in the allocation of decomposed requirements to the intended functionality and an associated safety mechanism, then:*

a) the associated safety mechanism should be assigned the highest decomposed ASIL;

b) a safety requirement shall be allocated to the intended functionality and implemented applying the corresponding decomposed ASIL."

Recommendation: ASIL decomposition can made by decomposing a safety requirement into two equivalent safety requirements, one of which is allocated to an intended functionality (i.e. a nominal function of the considered system) and the other is allocated to an associated safety mechanism. The idea is then that the nominal function, which is typically quite complex, can be assigned a relatively low ASIL while the safety mechanism, which is typically relatively simple, can be assigned a relatively high ASIL. Thus, the need for extremely stringent development of the (complex) nominal functionality is alleviated.

ASIL decompositioning rules in general and this recommendation in particular, can be addressed in a checklist for the system development.

8.5.9 Part 9: ASIL-oriented and safety-oriented analyses, Clause 5.4.9

„*When applying ASIL decomposition to a safety requirement, then:*

a) ASIL decomposition shall be applied in accordance with 5.4.10;

b) ASIL decomposition may be applied more than once;

c) each decomposed ASIL shall be marked by giving the ASIL of the safety goal in parenthesis."

Recommendation: ASIL decomposition may be applied more than once, for example an ASIL D requirement may be decomposed into one ASIL C(D) requirement and one ASIL A(D) requirement. The ASIL C(D) requirement may be further decomposed into an ASIL B(D) requirement

and an ASIL A(D) requirement.

8.5.10 Part 9: ASIL-oriented and safety-oriented analyses, Clause 5.4.10

„*One of the following decomposition schemes outlined below shall be chosen in accordance with the ASIL before decomposition (as shown in Figure 2), or a scheme resulting in higher ASILs may be used.*

a) An ASIL D requirement shall be decomposed as one of the following:

1) one ASIL C(D) requirement and one ASIL A(D) requirement; or

2) one ASIL B(D) requirement and one ASIL B(D) requirement; or

3) one ASIL D(D) requirement and one QM(D) requirement.

b) An ASIL C requirement shall be decomposed as one of the following:

1) one ASIL B(C) requirement and one ASIL A(C) requirement; or

2) one ASIL C(C) requirement and one QM(C) requirement.

c) An ASIL B requirement shall be decomposed as one of the following:

1) one ASIL A(B) requirement and one ASIL A(B) requirement; or

2) one ASIL B(B) requirement and one QM(B) requirement.

d) An ASIL A shall not be further decomposed, except, if needed, as one ASIL A(A) requirement and one QM(A) requirement."

Recommendation: If ASIL decomposition is performed, it shall be performed in accordance with the prescribed decomposition schemes in part 9 of ISO 26262.

9 Appendix 2: Methodology description for SEooC development with EAST-ADL

An opportunity of the MAENAD project is to extend the EAST-ADL methodology for the development of generic elements, named in ISO26262 Safety Element out of Context (SEooC).

The following aspects will be addressed by the methodology to develop a SEooC:

- Definition of the assumptions
- Safety lifecycle constraints
- Development strategy definition
- SEooC integration into the item

9.1 SEooC Overview

In the following chapters an overview on the SEooC will be given.

9.1.1 Automotive Industry approach on developing generic elements

The Car makers develop **generic systems or sub-systems** for different vehicles platforms, for different applications, before the finalising a specific vehicle. These generic elements are "*Systems on the shelf"* available for a future specific application. The Tier 1 suppliers develop **generic sub-systems or electronic control unit** for different customers or for different applications. Moreover the Tier 2 and 3 suppliers develop **generic components, modules** (HW, SW) for different ECU, different systems or for different applications.

In general the automotive industry develops **generic elements** for different applications and for different customers.

These **generic elements** can be developed in respect to the **functional safety approach** as **S**afety **E**lements **o**ut **o**f **C**ontext **SEooC** (ref. ISO 26262 - Road vehicles — Functional safety — Part 2: Management of functional safety - Clause 6.4.5.6 and Part 10: Guideline - Clause 9)

The SEooC is the more convenient way to develop a system applying the ISO26262 usable in multi –vehicles models and multi-platforms:

- The car-maker can apply the SEooC in different vehicles, models, application
- The suppliers can supply many customers

The SEooC is the way to deal the right responsibilities between

- OEM → Supplier(s)
- Tier 1 supplier →own suppliers

The SEooC is the way to reduce the "time to market" using "system on the shelf" already developed and verified

9.1.2 SEooC description

To better define the significant of the SEooC in respect of specific vehicle architecture the following definitions are useful:

- SEooC is a **generic element(s)** developed independently by different organizations.
- SEooC is a **safety-related element** which is **not developed** in the context of a **specific vehicle**.
- SEooC **isn't an Item** and **isn't developed for a specific item.**

SEooC differs from *qualified HW and SW components*, in fact it is developed in accordance with ISO 26262, it is based on assumptions, it is re-usable in multiple different items when the validity of its assumptions can be established during integration of the SEooC into the item.

The Qualified software / hardware components are not necessarily designed for reusability nor developed under ISO 26262. The Qualified software / hardware components are **pre-existing elements** available for an item developed under ISO 26262.

9.1.3 The assumptions

To develop a SEooC it is necessary to define a set of **assumptions** to which the SEooC aims.

The assumption can be categorized in two mainly classes: the *External,* related to the reference vehicle target (es E/E architecture, system(s), environment, ..) and the *Internal*, requirements and safety requirements, related to the application, that are placed on the element by higher levels of design. The assumptions allow the correct integration of the SEooC into a specific Item, this is allowed **checking** the **consistency of the assumptions** in respect with the specific interfaces of the *item*.

In case of the **SEooC assumptions do not fulfill the item** requirements is necessary a change(s) to the SEooC or a change(s) to the Item.

The **assumptions** give consistency to the application of the ISO26262 on developing of the generic element(s) during the item integration phase.

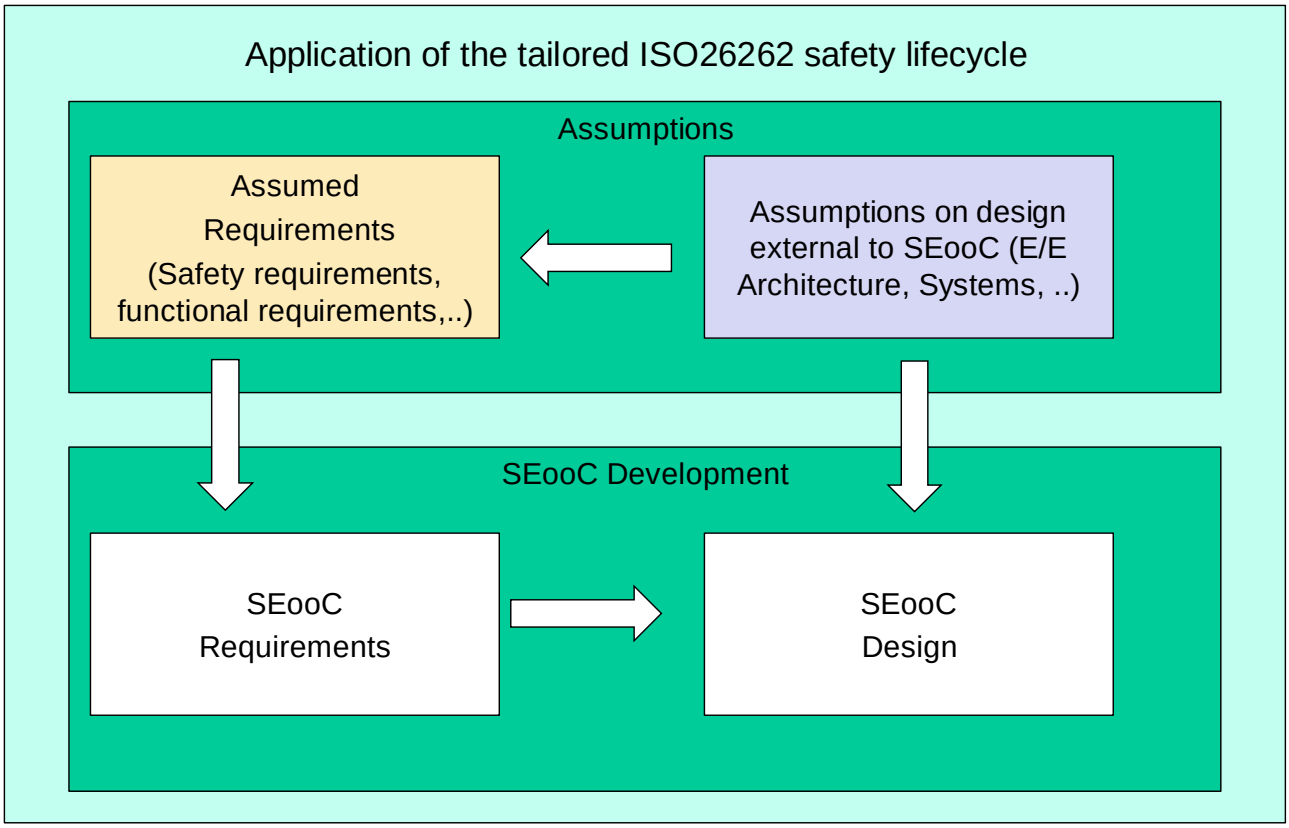


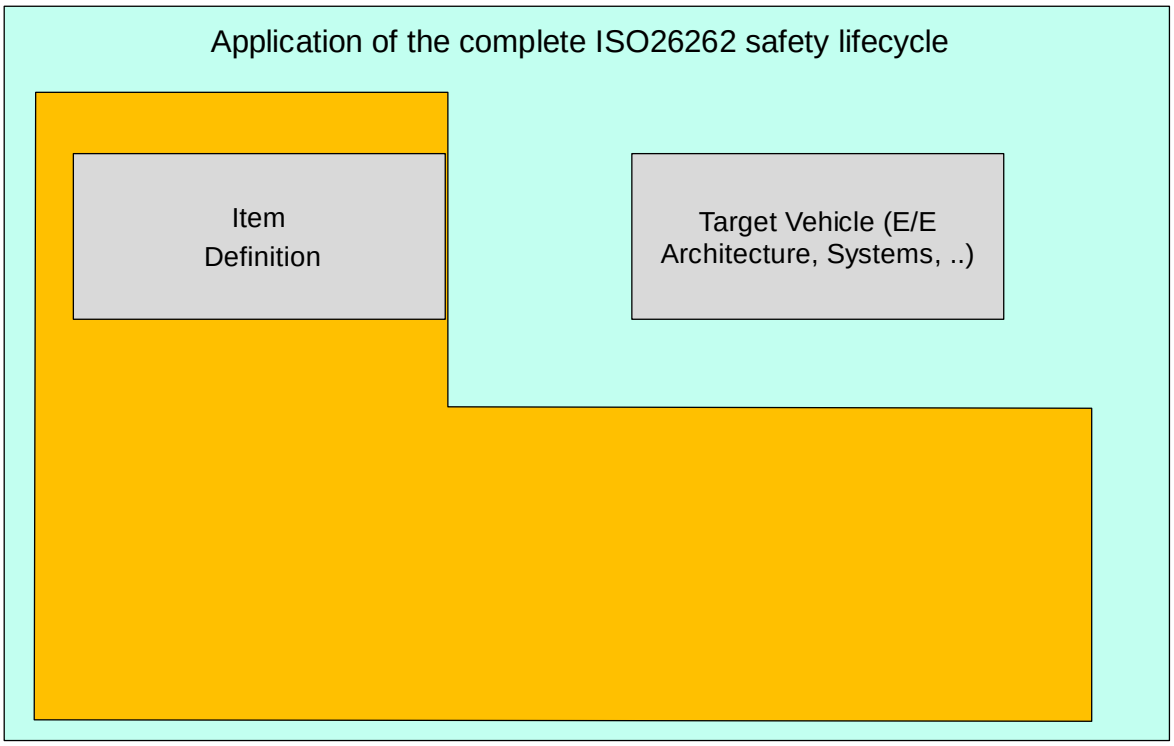
Figure 13 – SEooC – tailored Safety lifecycle

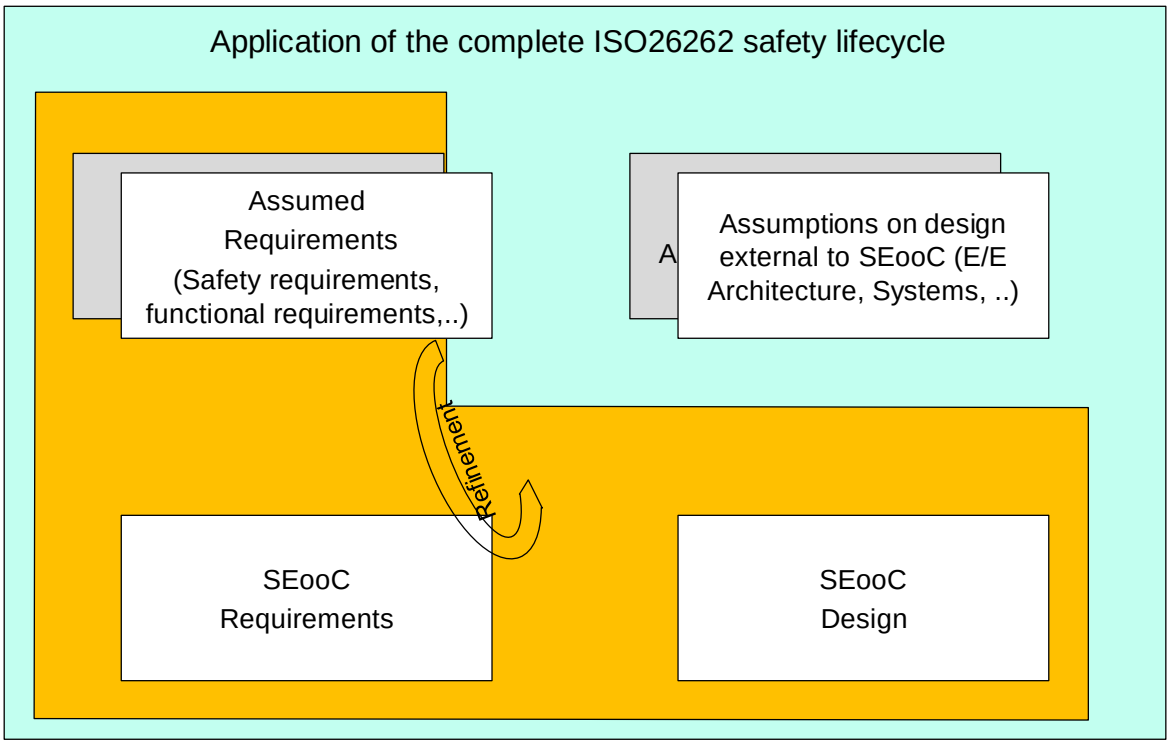
Examples of SEooC:

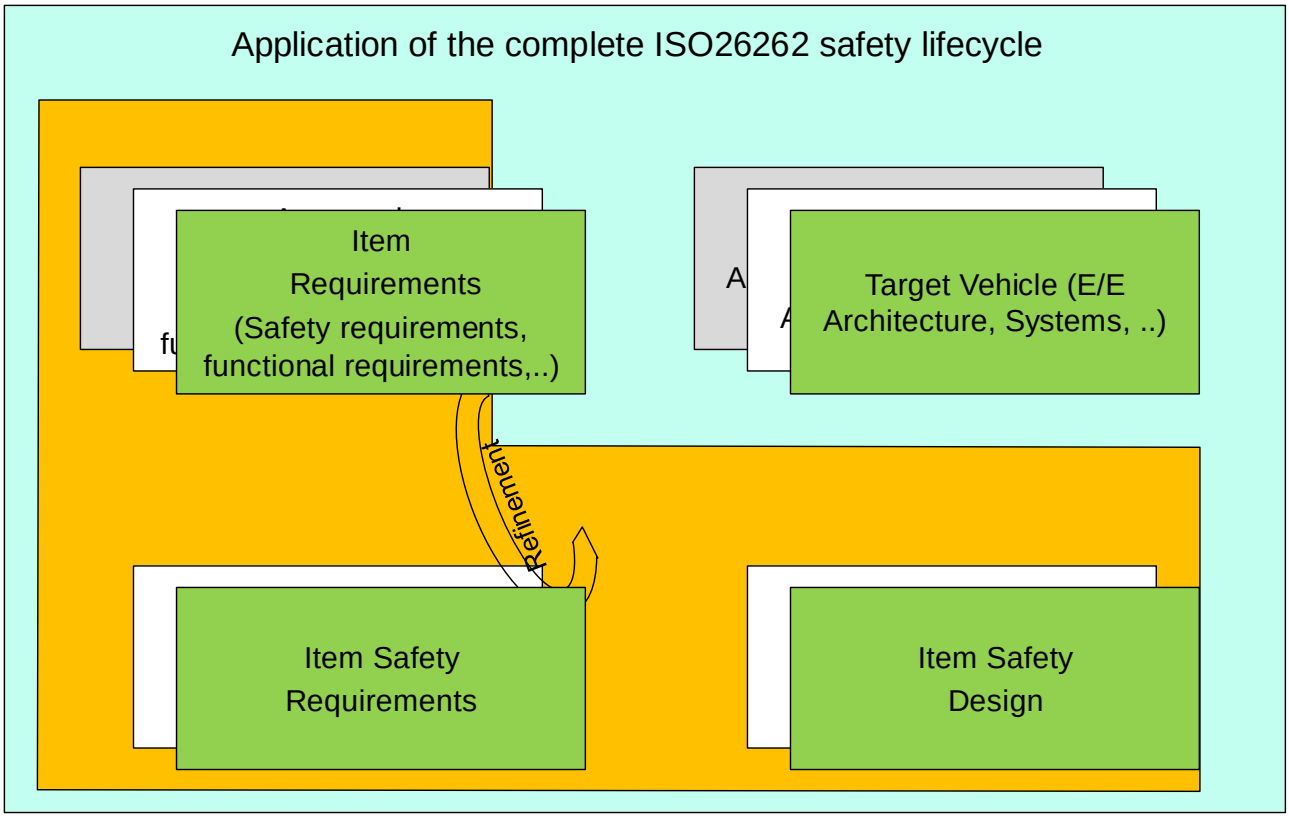
- System, array of systems, a subsystem, like Start & Stop, Electric Propulsion …
- System controllers, ECUs
- Microcontrollers
- Software implementing a communication protocol
- AUTOSAR application
- Software modules and AUTOSAR basic software modules

9.2 Overall SEooC design process

Given the complexity.







9.2.1 Safety Element out of Context Key Points on development

In the following chapter the following key points, regarding the SEooC development, are defined:

- Safety life Cycle
- ASIL – Safety requirements & Assumptions determination
- Verification activities
- Item integration

SEooC development – **Safety life Cycle**

- Each SEooC can be developed **at many Safety Life Cycle level**, depending on the functionalities and types.
- The **Safety activities can be tailored**, in respect of the constraints of ISO 26262.
- Any step of the **safety lifecycle cannot be omitted**. The **steps deferred** during the SEooC development are **completed during the item development**.

SEooC Development – **ASIL – Safety requirements & Assumptions determination**

- **The Capability of the developed SEooC** to comply with assumed safety requirements assigned with a given ASIL shall be demonstrated and verified.

- **The Selection of the ISO 26262 requirements,** that are applied to develop the specific SEooC, shall be verified in respect on the tailored Safety lifecycle.
- **The Developed SEooC shall be based on assumptions** on intended functionality, use context, external interfaces.

SEooC Development – **Verification activities**

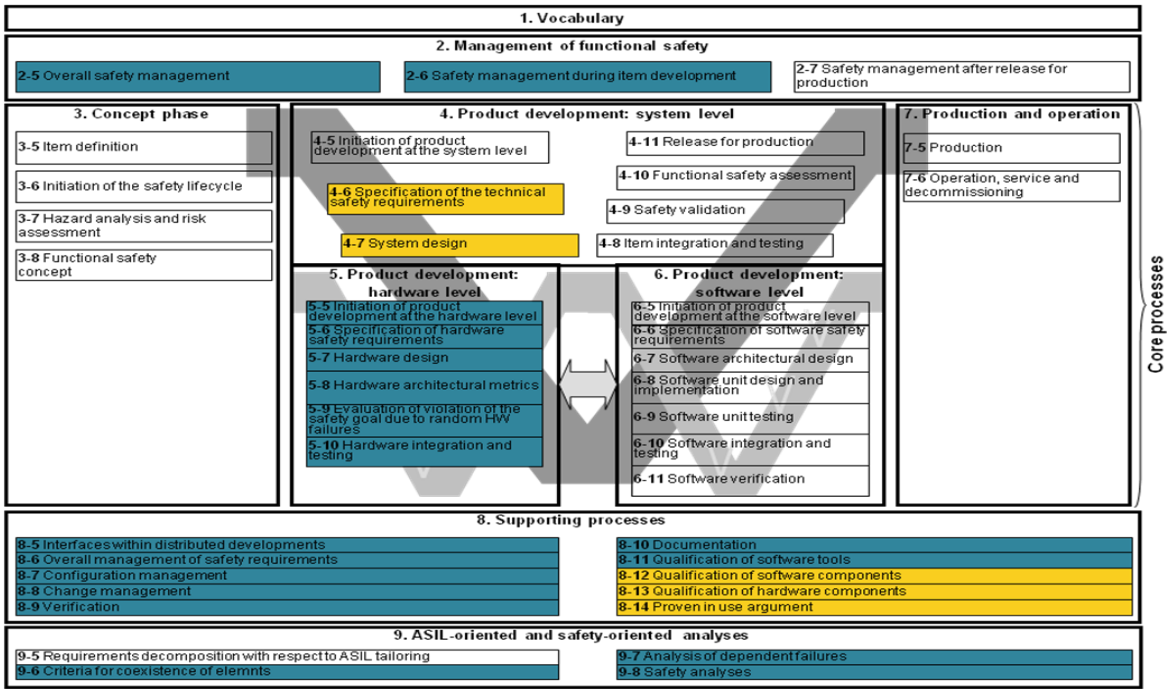
- The Verification activities developed during the different phases of SEooC development shall be performed in respect on the given ASIL.
- The Verification activities shall be carried on to demonstrate that a developed SEooC, at any level, is consistent with the requirements in the context where it is used.
- The verification work product is provided to the item development during the SEooC integration. Item development reaches the phase where requirements on the safety element are formulated.

SEooC Development – **Item integration**

- **The Assumptions** allow to SEooC **to address a superset of items**.
- The SEooC can be used in **multiple "different but similar items" later.**
- **The SEooC developer provides to the item designer** requirements and assumptions related to the SEooC.
- **The Item integrator checks and validates the SEooC assumptions.**
- **The Item integrator checks and validates multiple SEooC**, with SEooCs interfacing directly to each other (considering the interfacing SEooC).

9.2.2 Development of a Hardware component as a Safety Element out of Context

In the following pictures, that describe the whole ISO 26262 Safety Lifecycle, have been highlighted all the involved requirements in developing a Hardware component as a Safety Element out of Context, including not only the development core processes but also the supporting processes and the management of the functional safety.



Step 1 - Assumptions on System Level

Definition of the assumptions (Example a microcontroller):

Internal - Technical Safety Requirements The contribution of the MCU to the total probability of violation of a safety goal shall be no more than certain value of the allowed probability for the relevant ASIL.

External - System Level Design The system will implement a safety mechanism on the power supply to the MCU to detect over voltage and under voltage failure modes.

Step 2 – Execution of Hardware Development

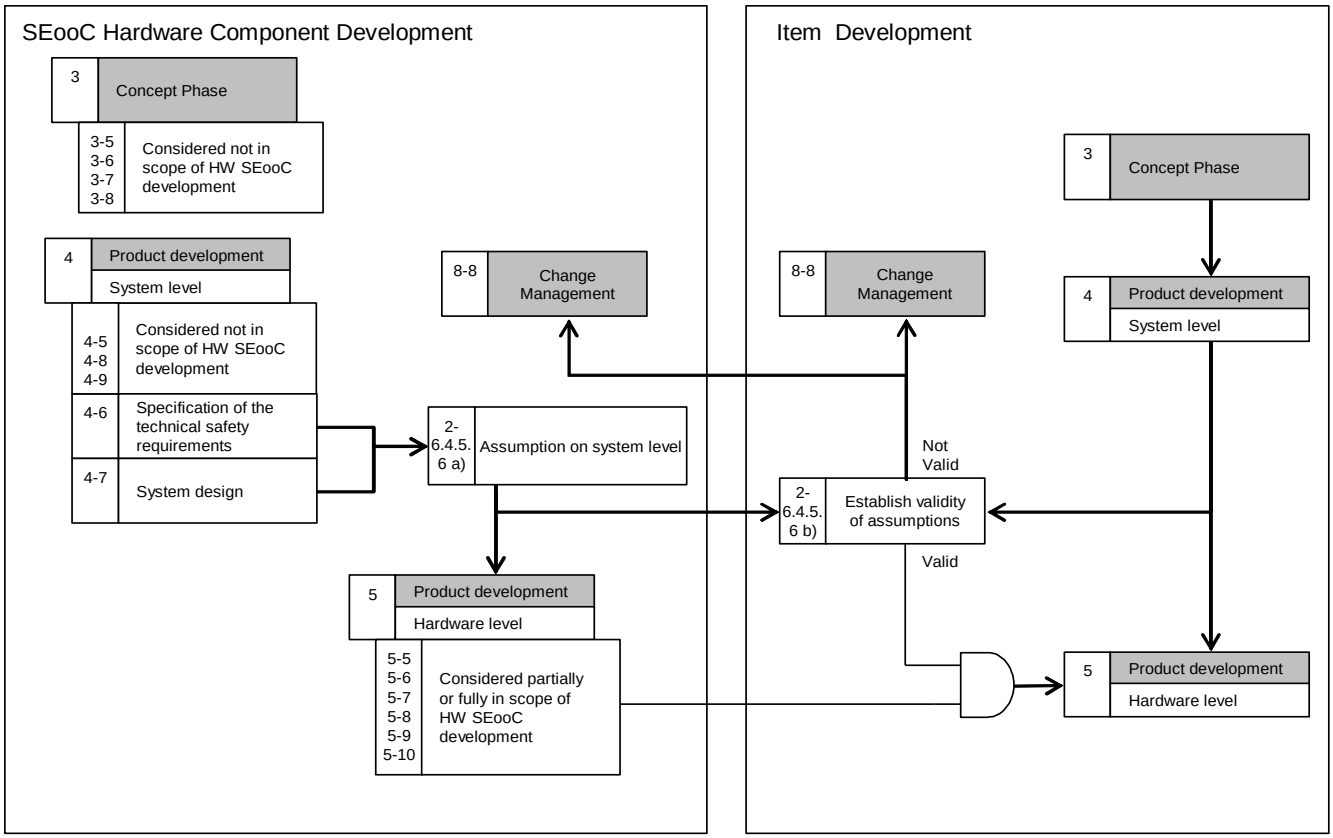
On the basis of the assumptions (internal and external), the SEooC is developed in respect of ISO 26262-5 and each applicable work product is prepared and the safety analyses with analysis of dependent failures internal to the MCU is performed.

Step 3 – Work Products

Information from the work products is provided to the system integrator, including documentation of assumed requirements, assumptions related to the design external to the SEooC, and applicable work products of ISO26262 such as the report of probability of violation of safety goal due to random HW failure.

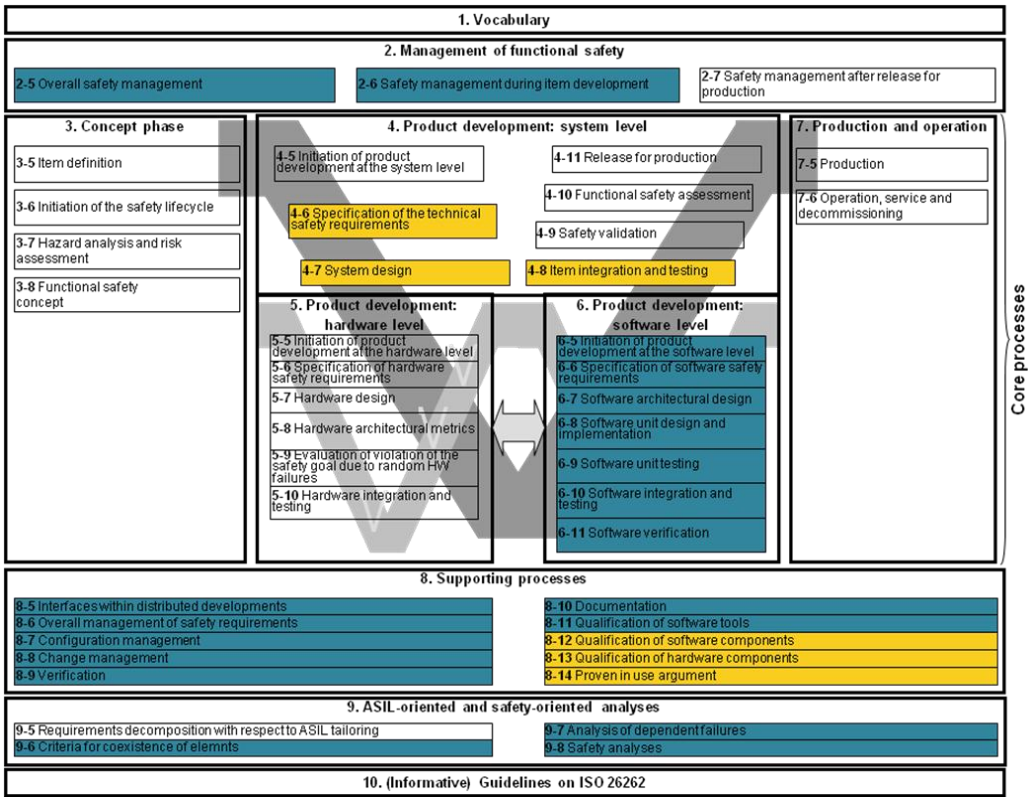
Step 4 – Integration of the SEooC into the item

Validity of each SEooC assumption: internal and external, are checked. In case of mismatch a change management activity, beginning with impact analysis, is conducted.


SEooC Hardware Component Development
3
Concept Phase
3-5
3-6
3-7
3-8
Considered not in scope of HW SEooC development
4
Product development
System level
4-5
4-8
4-9
Considered not in scope of HW SEooC development
4-6
Specification of the technical safety requirements
4-7
System design
8-8
Change Management
2-6.4.5.6 a)
Assumption on system level
5
Product development
Hardware level
5-5
5-6
5-7
5-8
5-9
5-10
Considered partially or fully in scope of HW SEooC development
Item Development
3
Concept Phase
4
Product development
System level
8-8
Change Management
Not Valid
2-6.4.5.6 b)
Establish validity of assumptions
Valid
5
Product development
Hardware level

9.2.3 Development of a Software component as a Safety Element out of Context

In the following pictures, that describe the whole ISO 26262 Safety Lifecycle, have been highlighted all the involved requirements in developing a Software component as a Safety Element out of Context, including not only the development core processes but also the supporting processes and the management of the functional safety.



Step 1 - Assumptions

Definition of the assumptions:

Scope of the software component as an SEooC - to state the relevant assumptions regarding the purpose of the software component, its boundaries, its environment and its functionalities.

Safety requirements of the software component - higher level safety requirements that potentially impact the software component in order to derive its software safety requirements.

Step 2 – Execution of Software Development

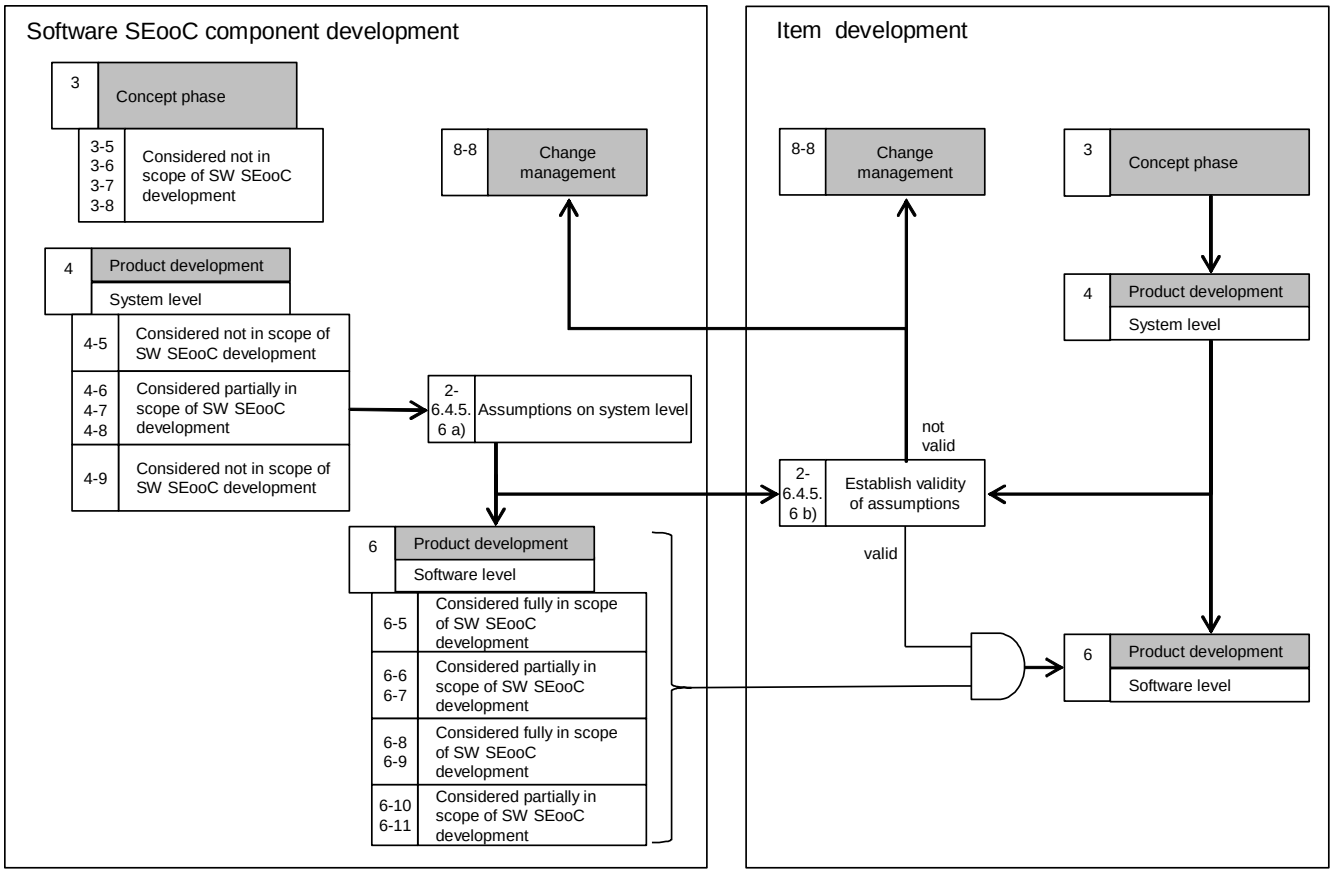
Once the necessary assumptions on the software component are explicitly stated, the SEooC is developed in accordance with the requirements of ISO 26262-6 corresponding to its ASIL capability.

Step 3 – Work Products

Each applicable work products are made available for further integration in different contexts, including the work products related to the verification of the assumed software safety requirements.

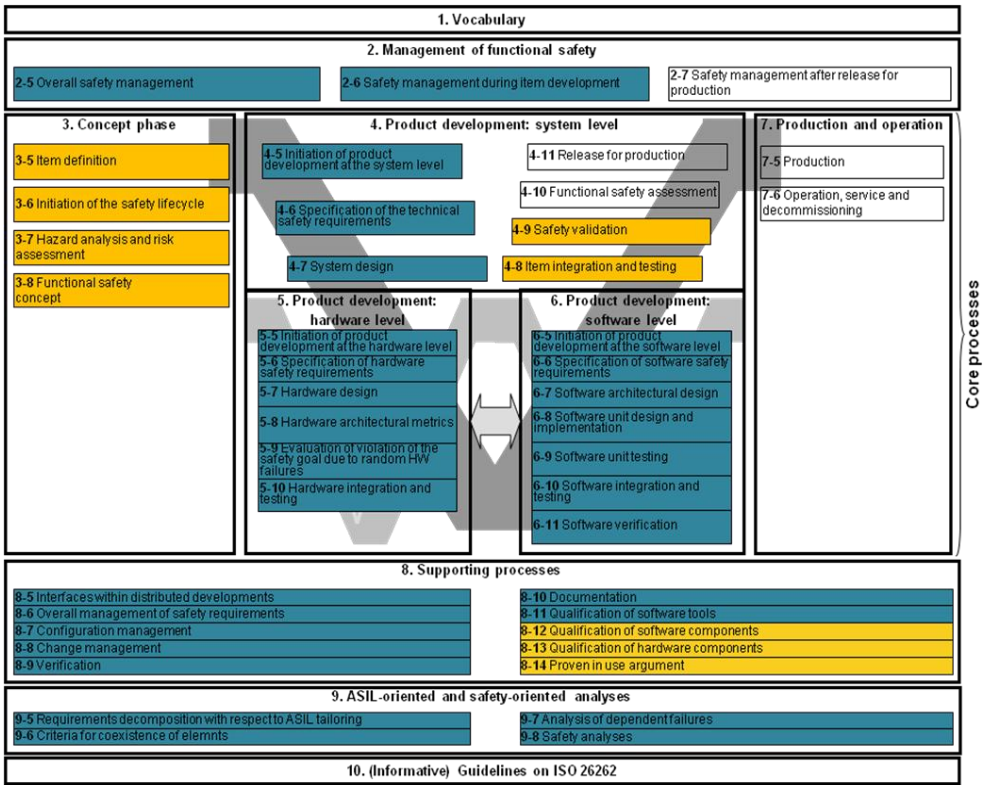
Step 4 – Integration of the software component in a new particular context

Validity of each SEooC assumptions are checked with regard to this context, in case of mismatch a change management activity beginning with impact analysis is conducted.


Software SEooC component development
3
Concept phase
3-5
3-6
3-7
3-8
Considered not in scope of SW SEooC development
8-8
Change management
4
Product development
System level
4-5
Considered not in scope of SW SEooC development
4-6
4-7
4-8
Considered partially in scope of SW SEooC development
4-9
Considered not in scope of SW SEooC development
2-6.4.5.6 a)
Assumptions on system level
6
Product development
Software level
6-5
Considered fully in scope of SW SEooC development
6-6
6-7
Considered partially in scope of SW SEooC development
6-8
6-9
Considered fully in scope of SW SEooC development
6-10
6-11
Considered partially in scope of SW SEooC development
Item development
8-8
Change management
3
Concept phase
4
Product development
System level
not valid
2-6.4.5.6 b)
Establish validity of assumptions
valid
6
Product development
Software level

9.2.4 Development of a System as a Safety Element out of Context

In the following pictures, that describe the whole ISO 26262 Safety Lifecycle, have been highlighted all the involved requirements in developing a System as a Safety Element out of Context, including not only the development core processes but also the supporting processes and the management of the functional safety.



Step 1 - Assumptions on System Level

Definition of the assumptions, the SEooC can be defined selecting on of the following class of assumptions:

External – Vehicle Level to define the purpose, functionalities and external interfaces of the SEooC. Examples of on the assumption on the scope of the SEooC can be:

a) The system shall be designed for vehicles with the following general characteristics:
   - o A gross mass up to xxx kg.
   - o The system shall be designed for rear wheel driven vehicles.

b) The system shall be interfaced with a EE vehicle architecture with the following characteristics:
   - o VDC (vehicle dynamic control) component will provide information on: vehicle speed,...
   - o IPC (instrument panel cluster) component will provide a gateway function with other sub-systems.

c) Functional requirements:
   - o The system shall activate the function at key on.
   - o The system shall deactivate the function at certain vehicle condition or when requested by the driver.

Internal - Safety Requirements to define assumptions on the functional safety requirements allocated to the SEooC. Examples of on the assumption can be:

a) Safety Goal 1
   - To deactivate the function at vehicle speed above a certain threshold.
   - Related ASIL x.
   - Safe state: none.

b) Safety Goal n

Step 2 – Development of the SEooC

a) The SEooC safety goals have been derived from the External Assumption applying the Part 3 of ISO26262.

or

b) The SEooC technical safety requirements have been derived from the assumed functional safety requirements of the item.

Then it is developed as written in ISO 26262.

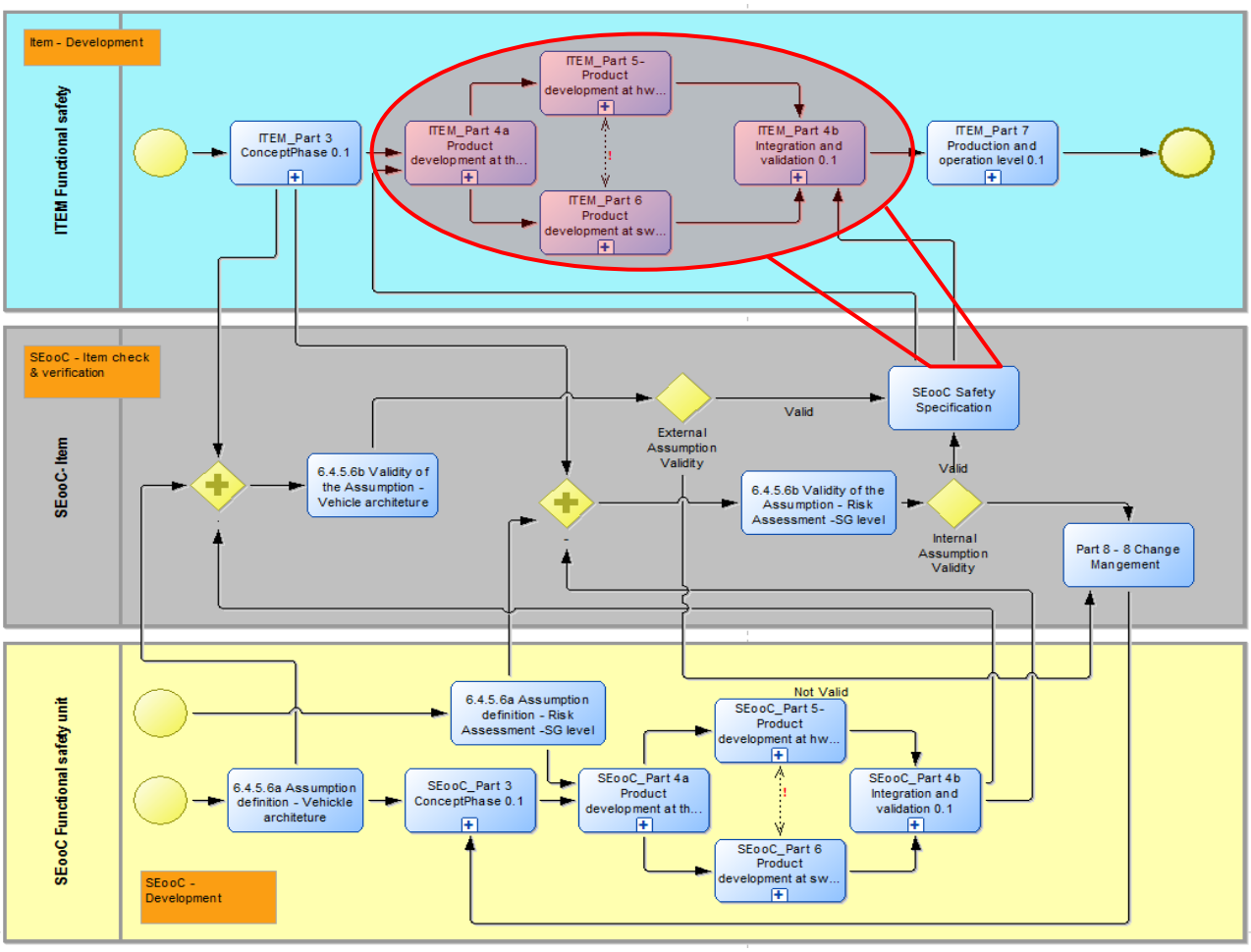
Step 3 – Work Products

Each applicable work products are made available for further item integration, including the work products related to the verification of the assumed the safety requirements.

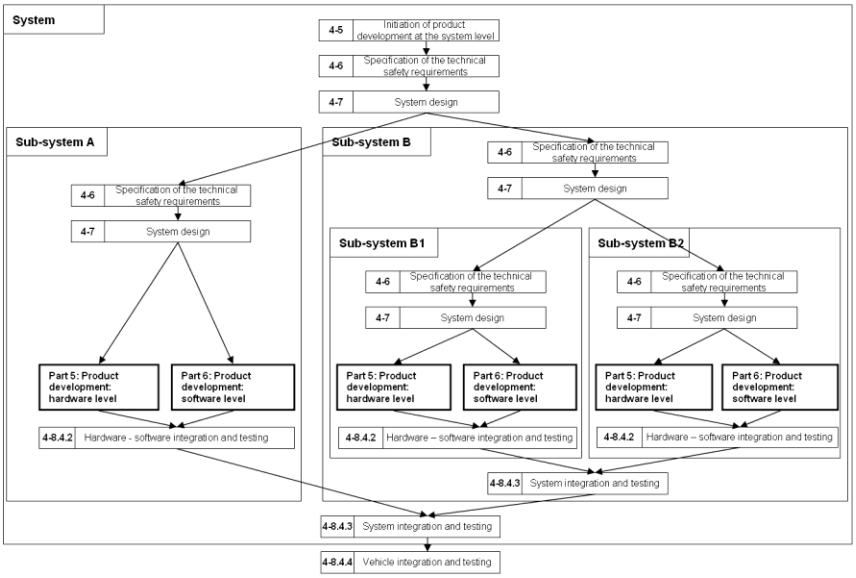
Step 4 – Integration of the SEooC into the item

Validity of each SEooC assumptions are checked with regard to different contexts, in case of mismatch a change management activity beginning with impact analysis is conducted.

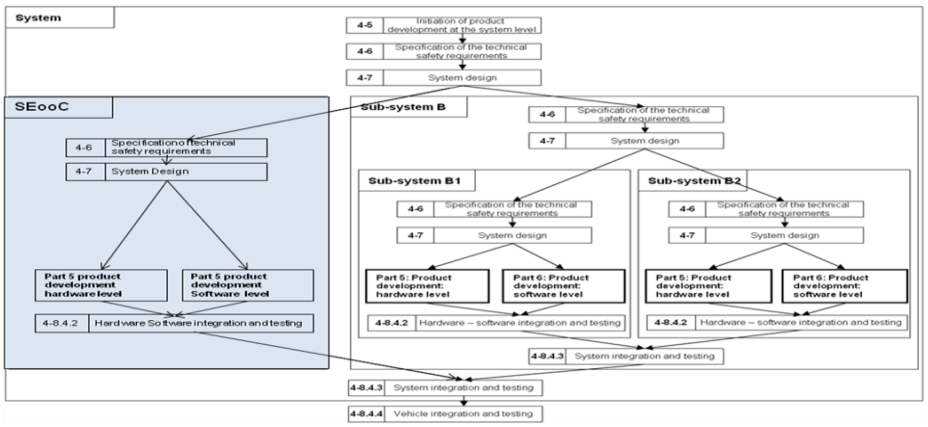
In the following pictures has been extracted the design flow, defined in ADONIS tool, in which the SEooC integration is highlighted.



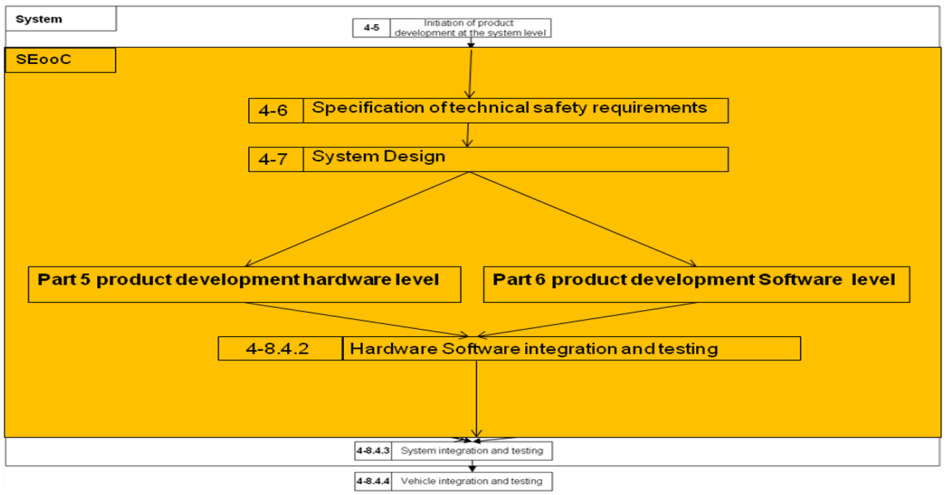
It is possible to build an item out a multiple Sub-Systems



with a SEooC(s), that develop a sub-system, interfacing directly to each other. In this case the validity of the assumptions of the SEooC is established considering the interfacing Sub-Systems.

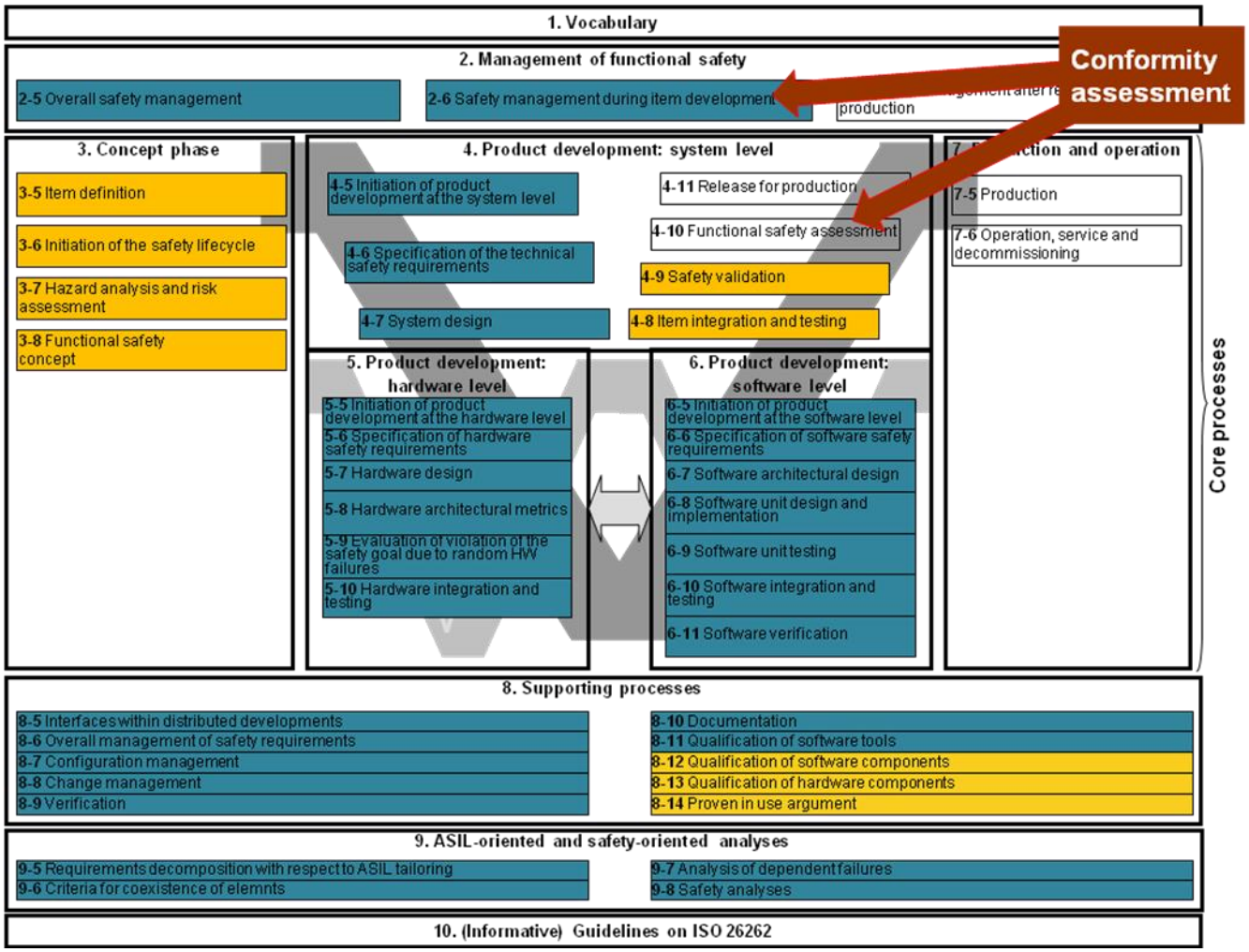


If the Item is composed by a single system then the assumption of the SEooC shall respect the boundary of the item.



9.3 Functional Safety Assessment vs. SEooC

Given



The management of functional safety includes the responsibility to ensure the application of **confirmation measures.**

The **Functional Safety Assessment** shall include [ref. ISO 26262, Part 2]:

- *The **work product** required by the safety plan.*
- *The **process required** for functional safety.*
- *Reviewing the appropriateness and effectiveness of the implemented **safety measures** that can be assessed during the item development.*
- During the functional safety assessment the confirmation reviews and the functional safety audit outcomes are evaluated.
- The result of functional safety assessment shall be:
  - ***Acceptance***

- ***Conditional acceptance*** → if the functional safety of the item is considered evident, despite the identified open issues; in this case, recommendation for conditional acceptance shall be included in the functional safety assessment report.
- ***Rejection*** → in this case, adequate corrective actions shall be initiated and the functional safety assessment shall be repeated.

9.4 EAST-ADL in SEooC development

9.4.1 Overall design process

Given the complexity of the development activities in automotive embedded software development, it is mandatory to structure the methodology so as to enable a relatively fast and easy access to the EAST-ADL language for a small kernel of essential development activities which can then be seamlessly extended to a comprehensive treatment of the language including more specialised development activities which may not necessarily be used in any development project. Hence the methodology is structured into two major components. This structuring is analogous to the structuring of the EAST-ADL language itself.

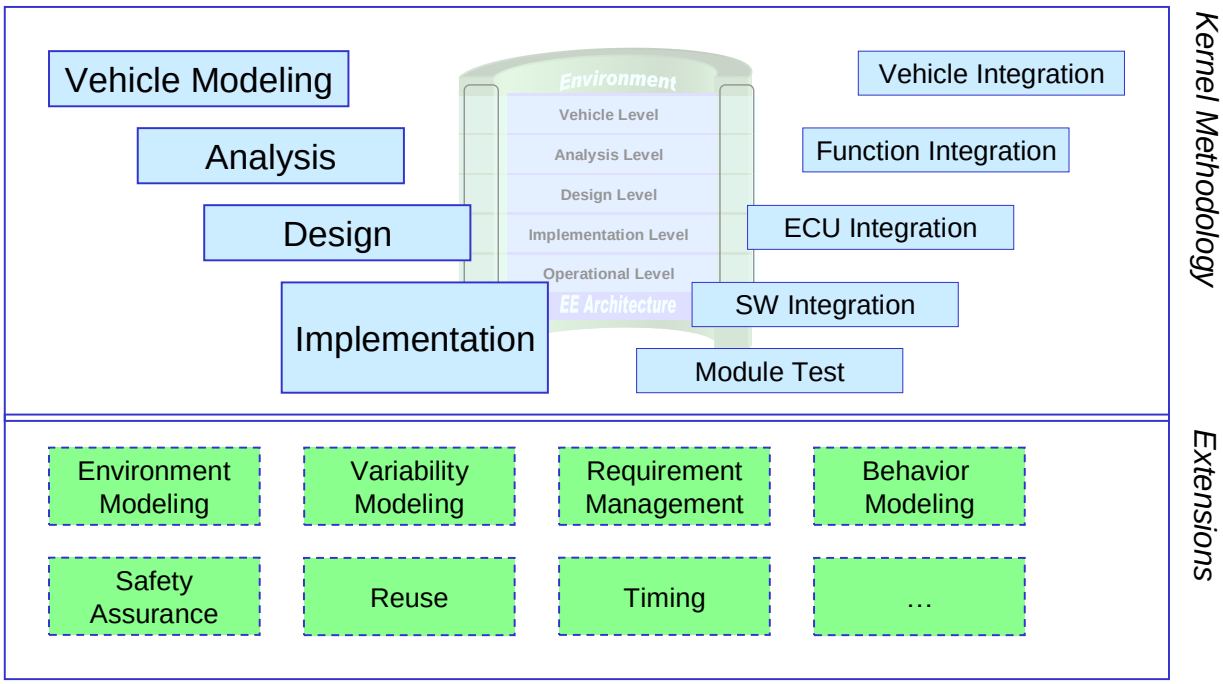


Figure 14 – EAST-ADL Structure

The main component, the kernel development part, comprises a top-down description of the central constructive phases of automotive embedded software development:

- **Vehicle Modeling:** The analysis of external requirements resulting in the construction of a top-level vehicle feature model together with the definition of necessary or intended feature configurations. In addition, for each feature a set of requirements is specified at vehicle level.
- **Analysis:** The creation of a functional analysis model specifying a solution of the requirements without concern about implementation restrictions of automotive series development. The analysis model is a logical representation of the system to be developed and its environment, and the boundary of the system to its environment. All the modeling in this phase will be on a logical behavior level, i.e. it will make no distinction between HW and

SW or about the implementation of communication. Behavior may be specified in detail by executable models.

- **Design:** The creation of a functional design model specifying a solution to the requirements in terms of efficient and reusable architectures, i.e. sets of (structured) HW/SW components and their interfaces, hardware architecture, and a mapping from functional components to HW/SW components. The architecture must satisfy the constraints of a particular development project in automotive series production.
- **Implementation:** The HW/SW implementation and configuration of the final solution. This part is mainly a reference to the concepts of AUTOSAR which provides standardized specifications at this level of automotive software development. However, the use of AUTOSAR concepts is not mandated by the methodology. Other, in particular more traditional implementation concepts can be used in this phase while leaving the other phases unchanged.

The core methodology is extended into a comprehensive methodology for automotive development projects by adding three additional and orthogonal activities to each of these phases:

- Specification of V&V cases to be executed and evaluated during the corresponding integration phase. V&V cases are most typically test cases, but can also include reviews etc.
- Verification of the model on a given abstraction level to the requirements of the model at the abstraction level directly above.
- V&V activities on the model artifacts of a given level itself, i.e. peer reviews, consistency checks, check of modeling guidelines etc.

Automotive software development also has to go through a number of integration and testing phases. While the methodology tries to be comprehensive handling the construction phases, the integration activities are only covered inasmuch they involve V&V activities and the relation to V&V-artifacts defined in the construction phases.

The second main component of the EAST-ADL methodology consists of a set of complementary loosely-coupled extensions to the core development part. Each of these extensions may be used as an add-on to the core activities. Extensions can of course also be combined depending on project needs. The following extensions are currently included:

- **Environment Modeling:** Modeling of the (typically analog or discrete-analog) environment of the system to be developed.
- **Requirements and V&V:** Detailed handling of complex requirements and V&V artifacts.
- **Safety Assurance:** Development of Safety-critical systems.
- **Timing:** Detailed handling of timing requirements and properties.
- **Variability Modeling:** Detailed handling of variability modeling.
- **Behavior modeling:** Detailed handling of behavioral modeling.
- **EV specific modeling**: Detailed handling of FEV relevant issues.

9.4.2 EAST-ADL can support the SEooC application

A structured modelling performed in EAST-ADL can support the SEooC application.

SEooC Based on assumptions:

captured in EAST-ADL dependability model;

during integration, the assumptions are matched vs. Item requirements (Safety goal, FSC, TSC depending on SEooC abstraction level).

Safety case: SEooC safety case is part of Item Safety case.

Each SEooC can be developed at many Safety Life Cycle level, depending on the functionalities and types: EAST-ADL model reflects the character of the SEooC development.

Any step of the safety lifecycle cannot be omitted: This is secured by the methodology steps and the existence of the required work products.

Capability, requirements and assumptions captured with EAST-ADL models and in particular the ASIL constraint.

Verification and integration: EAST-ADL models with requirements and constraints is basis.