Using the CESAR Safety Framework for Functional Safety Management in the context of ISO 26262

Eric Armengaud, Quentin Bourrouilh, Gerhard Griessnig, Helmut Martin, Peter Reichenpfader

Using the CESAR Safety Framework for Functional Safety Management in the context of ISO 26262

E. Armengaud[1], Q. Bourrouilh[1], G. Griessnig[1], H. Martin[2], P. Reichenpfader[2]
[1]AVL List GmbH {eric.armengaud, quentin.bourrouilh, gerhard.griessnig}@avl.com,
[2]Virtual Vehicle Competence Center {helmut.martin, peter.reichenpfader}@v2c2.at

***Abstract*— The development of automotive embedded systems is a complex task because of (1) the large number and complexity of functionalities implemented in electrical and/or electronic (E/E) systems including software, (2) the high level of safety integrity required for these functionalities, and (3) the distributed development process involving different organizations. The functional safety standard ISO 26262 supports the development of automotive electronics by providing a detailed development process as well as methods for the management of functional safety and for the required supporting processes (such as confidence in the use of software tools). These development activities lead to the creation of more than 100 work-products according to more than 1300 requirements to comply with the functional safety standard. We propose in this work an extension of the CESAR safety framework in order to support the definition, management, monitoring and validation of customer projects developed according to ISO 26262.**

***Index Terms*—Functional safety, ISO 26262, automotive embedded systems, confidence in use of software tools**

I. INTRODUCTION

TODAY the complexity of automotive embedded systems is increasing very fast in order to satisfy the goals of low emissions and fuel efficiency (e.g., electrification of the powertrain), or to add additional functionalities (e.g., ESP). Automotive embedded systems are responsible for the management and monitoring of the E/E components (e.g., high voltage battery, electric motor). It is obvious that the correct and safe operation of the vehicle depends on the correct operation of its electronics and control software. To that purpose, standards for functional safety (such as ISO 26262 [1] or IEC 61508 [2]) have been introduced in order to provide measures to reduce the existing risk to an acceptable risk. These measures concern both (a) the organization (e.g., the implementation of a safety culture, based on an already existing quality management processes) and (b) the product (e.g., a well-structured development process including dedicated safety analyses). The application of new safety standards such as ISO 26262 usually results on a development effort and time increase in order to guarantee the product quality and thus provide the requested safety.

The management of functional safety for the development of safety-relevant automotive embedded systems is facing different challenges. First, during set-up, the planning of the safety activities shall be performed (in parallel to the project plan) by the elaboration of the safety plan. This work-product lists the different safety-related work-products to be developed, the respective responsibilities of the different partners as well as the interfaces between the partners (e.g. exchange of information). This step is crucial in order to identify all activities required and ensure good cooperation between the different teams within the project. Secondly, during project execution, guidance shall be provided by the safety expert to the different technical experts during the development lifecycle such that they can complete their development tasks according to the requirements from the functional safety standard. The challenge here is to efficiently manage the information contained in the ISO 26262 and retrieve the relevant requirements according to the given work-product and the required Automotive Safety-Integrity Level (ASIL). The third challenge at the end of the project (or at important milestones) is to provide evidence of correct completion of the development activities with respect to the functional safety standard. The difficulty is to perform the mapping between the project specific files and company-internal process descriptions with the ISO 26262 work-products. Finally an evidence for the fulfillment of each single ISO 26262 requirement of the safety lifecycle has to be provided. A further challenge concerns the confidence in the use of software tool (included in part 8 of the ISO 26262). Hence, in order to decrease the development time, improve the system quality and minimize the risk of human error, tools and tool chains are increasingly supporting development activities in order to automate some tasks. However, it is necessary to get enough confidence and trust in the tools and tool-chain that are used in order to achieve the required level of safety. Tool planning, evaluation, classification and qualification are required for the development of safety-relevant products in the context of ISO 26262.

The CESAR[1] project [3] aims at improving the processes and methods for safety-critical embedded systems development. An important output for the automotive domain is the "CESAR safety framework" that lists the activities requested by the ISO 26262 as well as the mapping to the corresponding roles and the inputs & outputs documents [4]. The contribution of this work is (1) the enhancement of the CESAR safety framework in order to support functional safety management during the set-up, execution and validation of the project, and (2) a method for the documentation of the project specific development process that enables tool classification for the "confidence in the use of SW tools" activity required by the ISO 26262.

The document is organized as follow: Section II reviews the state of the art regarding functional safety management and tool classification. In Section III, an approach for safety planning and tailoring is presented. The proposed excel tool is successively enhanced in order to provide a method for guiding development activities along the development process and further for the monitoring and documentation of safety activities during the project. This information serves finally as basis for the safety case. After that, Section IV presents a method for systematic description and analysis of the project-specific development process that is further used as input for the analysis regarding confidence in the use of SW tools. Finally, Section IV concludes this work.

II. STATE OF THE ART FUNCTIONAL SAFETY MANAGEMENT

This section introduces the ISO 26262 standard and in particular the part of it addressing the functional safety management. Furthermore, we present some work related to the application of this standard and also about the integration of other functional safety standards into internal company processes.

The ISO 26262 [1]: "Road vehicles – Functional safety" is the adaptation of IEC 61508 to comply with the automotive specific application related to E/E systems within passenger road vehicles. This version has been released and is publicly available on the ISO website. The standard is limited to series production passenger cars up to 3.5 tons, but could also be applied or serve as basis for other automotive domain products like heavy duty vehicles (e.g., trucks) or off road vehicle. This new International Standard includes guidance to avoid risks, caused by "systematic failures" and "E/E random hardware failures", by providing feasible requirements and processes.

The ISO 26262 is divided into nine parts, and:

- **Provides an automotive safety lifecycle :** This includes processes for the whole lifecycle of development (management processes, development; production; service and decommissioning) and supports tailoring of the necessary activities during these lifecycle phases;
- **Provides an automotive specific risk-based approach for determining risk classes**: Automotive Safety Integrity Levels (ASILs);
- **Reduces the Risk on hazard** by using these ASILs for specifying the item's necessary safety requirements for achieving an acceptable residual risk;
- **Provides requirements for verification, validation and confirmation measures** to ensure a sufficient and acceptable level of safety being achieved.
- **Provides requirements for the relation between the development partners** (e.g., OEM and suppliers).

The functional safety management topic is addressed in the following parts of the standard:

- Part 2, Management of functional safety which is divided in three main chapters: overall safety management (i.e. project independent), safety management during concept phase and the product development (i.e. project dependent), and safety management after the item's release for production.
- Clause related to initiation of product development in the following parts: 3 Concept phase / 4 Product development at the system level / 5 Product development at the hardware level / 6 Product development at the software level: The objective of these sub-phases is to plan and initiate the functional safety activities for the respective parts of the standard. It usually includes refinement of project plan and safety plan, eventually the elaboration of verification, validation or integration & testing plan.
- Part 8, Supporting processes, which describes the necessary supporting processes and activities to support functional safety, like, the procedures to manage distributed developments or to achieve an acceptable level of confidence in the use of software tools.

This functional safety standard is the main input for one of the activities of the CESAR project regarding the application of safety in the automotive domain. The partners of the project, including OEMs and Tier 1 suppliers analyzed the ISO 26262 in detail and created their own safety framework which defines and describes the required safety activities to carry out in each development phase of an automotive embedded system. For each of these defined safety activities, the necessary inputs and related outputs are associated, as well as additional information such as the methods that can be used, the roles and responsibilities involved and potential tool support. The CESAR framework has been elaborated in Excel and then implemented in an Eclipse Process Framework (EPF) Composer, a process modeling tool that allows a publication in HTML format [4] [5].

The two main enhancements of the proposed work in comparison to the CESAR framework are (1) the level of granularity of the information available and (2) the mapping to

[1] www.cesarproject.eu

safety activities and the deployment during customer projects. The documentation in the CESAR framework is available at a "safety activity level" (e.g., Part III-5 "Item definition"), necessary to understand the overall concept and approach of ISO 26262. In this work, the description has been refined at "ISO 26262 requirements level" (e.g., Part III-5.4.2 "The boundary of the item, its interfaces, and the assumptions concerning its interaction with other items and elements, shall be defined"), necessary for safety case documentation and functional safety assessment. Section III describes how the refined information has been used for covering different safety activities such as definition of DIA (Development Interface Agreement), safety monitoring or conformance review.

Approaches to incorporate the requirements of the generic functional safety standard IEC 61508 to a specific company internal process have been described in [6]. In particular it presents a framework to perform functional safety that have the following properties: simplicity (easier to use as the standard itself), self-containment (containing all information necessary to comply with the standard), flexible / adaptable (for different products and different processes among the company), scalable (variability of the product complexity), certifiable, minimum deployment impact. The approach proposed in this work is similar but focus on the specificities of the ISO 26262 and encompass additional aspects such as the development among several partners and the tool qualification.

The purpose of tool classification / qualification (confidence in the use of software tools) is to analyze the required level of confidence of SW tools in order to get enough confidence and trust in the tools and tool-chain that are used in order to achieve the required level of safety. Due to the different functional safety norms, there is no cross-domain standard for tool classification within development [8]. In avionics domain the safety standard DO-178B [9] requires tool qualification for all tools involved in the creation of airborne software. Railway systems and tooling industry uses IEC 61508 [2]. It defines functional safety as *part of the overall safety relating to the EUC (Equipment Under Control) and the EUC control system which depends on the correct functioning of the E/E/PE safety-related systems, other technology safety-related systems and external risk reduction facilities*. EN 50128 [10] that is intended to cover the development of software for railway control and protection including communications, signaling and processing systems, provides a specific interpretation of IEC 61508 for railway applications. The IEC 61508 standard also served as basis for automotive industry, for which ISO 26262 has just been released as an international standard. Both standards have different approaches for tool qualification. For railway systems IEC 61508 defines this as a *tool certification*, ISO 26262 denotes this as *confidence in the use of software tools*. These activities aim at analyzing the compound of tools required for the development of a given product and proposing methods to increase the reliability of the development tool chain.

For ISO 26262, such actions comprises (1) tool planning, which represents the mapping of the tools used in the project to the development activities, (2) tool evaluation, which represents the analysis of the tool impact on the product as well as the probability to detect possible faults introduced by the tool, and (3) tool qualification, which represents the means for minimizing the probability of an undetected tool error . The decision about the tools and their qualification measures strongly depends on the development process and on the ASIL of the system developed. The challenge of this activity is (a) to build a complete and systematic understanding of the development activities and the associated tools, (b) to identify for each tools their use-case – for which purpose the tool are being used, and (c) to identify the possible malfunctions and countermeasures that can be applied to minimize the risk of tool error.

As the automotive standard arose recently, there are only few examples and reference practices for automotive industry. Some tool vendors such as dSpace [11] or Mathworks [12] propose a reference workflow for this topic. These frameworks provide guidance for the usage of the specific tool in order to minimize error and increase confidence of correct operation. This is a tool-vendor centric approach that does not take into account the specificities of the system developer (e.g., specific process or use of dedicated additional tools in a tool-chain). The proposed framework causes a strong dependency to one software vendor and the proposed tool chain. Tool error detection is focused in verification tools of the vendor for the specific tool chain of the project. In the end this anchorage of development tools might lead to inflexibility for new development projects. Furthermore development activities and the usage of software tools depend on the involved partners and can change for a new project. For this reason a tool vendor independent approach is necessary to establish tool confidence that complies with the requirements from ISO 26262.

III. FUNCTIONAL SAFETY MANAGEMENT IN PROJECT

A. Functional safety management – project preparation

The development of a car necessitates the involvement of different partners for the development of the different assembly parts of a car (e.g., chassis, powertrain), for the development of the various components of the assembly parts (e.g., battery, engine, control unit) and finally for the development of the single sub-components: (e.g., application software, basis software, electronic hardware of the control unit). This work-split is mainly due to the cost optimization and specialization of each partner. Figure 1 provides an example of such a work-split.

Safety is a property of the vehicle and therefore responsibility of the car manufacturer. However, vehicle safety relies on the safe operation of each component and on the correct (safe) integration of these components. Therefore, the responsibility needs to be split among the Original Equipment Manufacturer (OEM) and the suppliers. For this reason, the ISO 26262 requires the definition of a DIA and specifies

requirements to manage development across multiple organizations. The DIA lists the activities to be performed by the OEM and / or the supplier(s) for:

- supplier selection,
- initiation and planning of the distributed development,
- execution of the distributed development,
- safety assessment at supplier's premises,
- planning of the activities after start of production

For each defined activity the documents or data that shall be exchanged between the OEM and the suppliers are detailed.

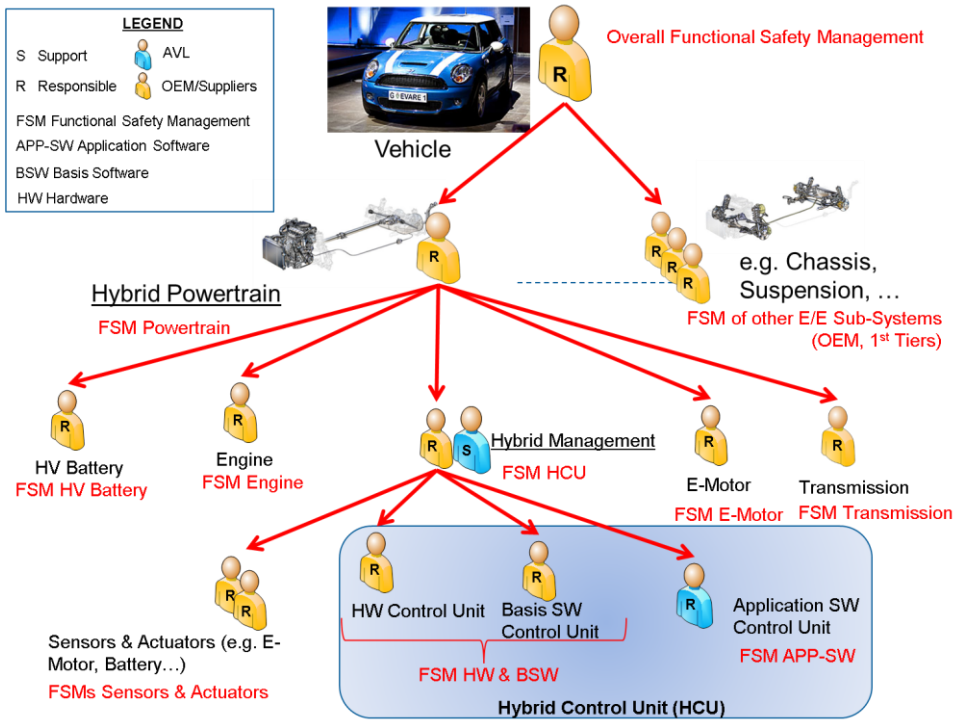

Figure 1: Example of work split

The first step of a project is to determine which ISO 26262 work-products have to be delivered and by which partner. For these activities a Development Interface Agreement (DIA) template is created, that includes the following information (see Figure 2):

- Work product ID, required to trace the work-product across the activities
- Work product name, such as "Item Definition"
- Distribution: this column is used to filter the work-products that are company internal only (e.g., organization specific rules and processes for functional safety, see Figure 2) and the ones that are distributed among the project partners (e.g., item definition)
- Distribution comments: explanations about the distribution
- Comments (e.g., brief description of the content of the work-product)
- RASI – Definition of responsibilities in the project for filtering the work-products (Responsible (R), Approval (A), Support (S), Information (I))
- Description of the activities for each partner on the associated work-product (example is given below)

We can note that the elaborated DIA template is based on the work-products (and their refinements) and not on the activities or requirements of the ISO 26262. Indeed, in a project, the work-products are the main means to exchange information between partners. The ISO 26262 defines 69 work-products compared to the requirements that are about 1300. It is also easier to make a link to the deliverables defined in the contract. But for each work-product, as mentioned above, the description of the activities for each partner is associated. For example, the Figure 2 presents a possible work-split for the work-products of the concept phase:

- Item Definition: AVL is providing a template and finalizing the item definition with the inputs of the OEM, who shall also review and accept the document. The suppliers are only involved for information.
- Impact analysis: this work-product is not performed due to the new development of the item.
- Safety plan: AVL is defining and planning the required safety activities according to the project scope and involvement and also providing support to the OEM with regard to guidance for the safety plan (e.g., topics to be included). On the other hand, the OEM shall of course define and plan its own safety activities but also coordinate, trigger

and make sure that all the suppliers are also creating and maintaining their own safety plan.

- Hazard analysis & Risk assessment: AVL is performing this first safety analyzing activity and presenting the results to the OEM who shall perform a verification review and accept the results. The suppliers are only involved for information.

The use of this DIA template in customer projects is really important to provide a clear allocation of the responsibilities in the project and make clear that safety is not the issue of only one partner. It provides furthermore good guidance for the project partners with less experience in the ISO 26262 in order to provide an overview of the tasks required by the standard and to plan accordingly their efforts. The DIA is one of the first safety documents to be discussed, agreed and exchanged with the partners in the project.

Based on the tailored DIA, each partner has to define a safety time schedule of their allocated tasks. To ease this step, we propose a MS-project template listing all the safety activities (sections of the ISO 26262, e.g., Part III-5 "Item definition") with their dependencies. The partners still need to refine the safety activities into tasks, assign one responsible and the time required for task completion. Based on this template, the safety timing plan can be tailored according to the DIA.

Workproducts defined in ISO FDIS 26262			AVL Work-products		DIA (RASI)					
Work-product ID	Work-product Name	Comments	Distribution	Distribution comments	OEM	1st Level Sup.	AVL	OEM Activities	1st Level Supplier Activities	AVL Activities
Concept Phase										
ISO_WP301	Item definition	Definition and description of the item under safety scope with its boundaries.	X	Distributed to all project partners	A	I	R	- OEM shall provide the information requested by AVL for the Item Definition. - OEM shall review the item definition to make sure that the item is well described and understood.	None	- AVL shall provide a template for the main function description. - AVL shall finalize the Item Definition.
ISO_WP302	Impact analysis	Hint: this report is only relevant for enhancements or changes of existing products		none				- OEM shall document the non-applicability of impact analysis in this project	None	None
ISO_WP303	Safety plan (refined)	Definition and planning of the safety activities to be performed in this project. This needs to be consistent with the Workproduct "Organization specific rules and processes for functional safety" for each organization.		This is a company internal work-product and is therefore not handed over. It is shown in case of a functional safety assessment.	I	I	R	- OEM shall define and plan its safety activities according to its own safety culture and its project scope. - OEM shall coordinate, trigger and make sure that the suppliers define and plan their safety activities.	None	- AVL shall define and plan its safety activities according to its own safety culture for the support on Tractor development level. - AVL shall support on request the OEM by providing guidance on this topic, e.g. list of topics that shall be included within the safety plan.
ISO_WP304	Hazard analysis and risk assessment	Identification and classification of the hazards.	X	Distributed to all project partners	A	I	R	- OEM shall perform a review according to OEM experience - OEM shall confirm full understanding (content) and acceptance --> H&R Release sheet has to be signed	None	- AVL shall provide the H&R content with explanation to ensure the understanding (e.g. Workshop) - Discussion of new situations, hazards... with OEM. - Recommendation for release

Figure 2: Example of Development Interface Agreement

B. *Functional safety management during development*

The next task after the completion of the safety planning is the execution of the development. Hence, the ISO 26262 standard defines more than 100 work-products and more than 1300 recommendations on these work-products (see Section II). The challenge is to perform project tailoring according to the ASIL and identify the requirements to be fulfilled and the methods to be used for the elaboration of each work-product.

The approach proposed in this work is based on an enhancement of the DIA presented in Section III. An additional spreadsheet lists all the 1300 recommendations of the ISO 26262 and links them to the relevant work-products, see Figure 3. Since recommendations and work products are organized in a matrix, each recommendation can be assigned to one or more work products by filling out the crossing cell with an attribute of the specified dependency. Filtering capabilities enables to focus on one dedicated work-product for one specific ASIL, thus efficiently identifying the work to be performed for this specific work product. Different kinds of dependencies are listed:

- *Input* (I): Link to another existing information that "shall be available" or "can be considered", usually this is listed at the beginning of each chapter (e.g., for the hazard analysis and risk assessment an input is item definition in accordance with 5.5.)
- *Refined* (R): the description of fulfillment of this requirement is refined in the selected work-product (e.g., the work-product safety plan is first defined in the part 2 management of functional safety and then "refined" during the whole development: concept phase, system level-, software level-, hardware level).
- *Output* (O): the fulfillment of this requirement shall be described in the selected deliverable
- *Link* (L): Requirements do not only reference a work product, they can also reference other requirements, which have to be fulfilled. If a work product WP_x has to fulfill a requirement R_{WPx}, which itself references other requirements, those requirements are marked with an "L" in the column

of the work product WP_x.

Using the filtering capabilities on the huge matrix, the requirements related to a given work-product for a given ASIL can be extracted very easily. This information is useful (1) for the developer in order to get a complete view of the recommended methods that shall be used and (2) for confirmation review in order to cross-check if all the activities according to ISO 26262 requirements have been correctly performed and documented.

The proposed method is currently in use in different customer projects at AVL. It has highly increased the efficiency of internal developments in providing guidance to the different AVL experts during specification, development and validation activities. Furthermore, the capability to extract requirements for the different tailored work-products is important for the relation with customers and suppliers in order to properly identify and agree on the requested quality and content of the different documents.

				Work Products: Part 2						
		(I)nput…green (S)upport…yellow (R)efined…blue (O)utput…red (L)ink…orange	2	2-5.5.1 Organization specific rules and processes for functional safety, hresulting from 5.4.2 and 5.4.5.	2-5.5.2 Evidence of competence, resulting from 5.4.3.	2-5.5.3 Evidence of quality management, resulting from 5.4.4.	2-6.5.1 Safety plan, resulting from 6.4.3 to 6.4.5.	2-6.5.2 Project plan (refined), resulting from 6.4.3.4.	2-6.5.3 Safety case, resulting from 6.4.6.	2-6.4.4 Functional safety assessment plan, resulting from 6.4.9.
Requirements										
Part		C	Full ID							
Pa-ID	Description			ISO_WP201	ISO_WP202	ISO_WP203	ISO_WP204	ISO_WP205	ISO_WP206	ISO_WP207
2	The objective of this clause is to define the requirements for the organizations that are responsible for the safety lifecycle, or that perform safety activities in the safety lifecycle.		2-5.1	o						
2	Existing evidence of a quality management system complying with a quality standard, such as ISO/TS 16949, ISO 9001, or equivalent.		2-5.3.2	i	i	i				
2	The organization shall create, foster, and sustain a safety culture that supports and encourages the effective achievement of functional safety.		2-5.4.2.1	o						
2	The organization shall institute, execute and maintain organization specific rules and processes to comply with the requirements of ISO 26262.		2-5.4.2.2	o						
2	The organization shall institute, execute and maintain processes to ensure that identified functional safety anomalies are explicitly communicated to the applicable safety manager(s) and the other responsible persons.		2-5.4.2.3	o			L		L	

Figure 3: Identification of the required development activities and tailoring of the ISO work-products according to the ASIL level

C. *Functional safety management – documentation of activities*

The challenge for the preparation of the safety case is to systematically map the project deliverables with the ISO 26262 work-products and to monitor and document the fulfillment of all the activities required by the functional safety standard. For that purpose, a link between the project deliverables and the ISO 26262 work-product shall be prepared. This list can be further used in order to monitor the safety activities and document the fulfillment of the tasks.

The mapping between the project deliverables and the ISO 26262 work-products is performed with an excel matrix. Additional information on the files (e.g., mapping within the development process, short description of the content, dependencies, availability of a review) is added for documentation. The output files can be linked as *part of* or *fully part of* to the work products. Part of means that the output file is referenced by a work product from ISO 26262, fully part of means, that the output file is fully integrated in the specific work product. This matrix enables to manage traces between the development activities and the ISO 26262 work-products (for which work-product(s) a given file serves an input, and for a given work-product which files are required). This mapping is important to validate the company internal development process and ensure that all ISO 26262 work-products are complete.

The second tool for project monitoring and documentation is the checklist for the development activities, see Figure 4. This checklist regroups the following information for each safety activity and for each ISO 26262 requirement relevant for the project:

- *Status:* status for the activity
- *Finding / Actions:* findings or actions required for the safety activity
- *Remarks / Rationale:* additional information, argumentation regarding choice performed
- *Responsible:* responsible for this activity
- *Due Date:* due date for this activity

This check-list enables monitoring and documentation of the project and serves as basis for the safety case when all the activities have been successfully completed.

IV. CONFIDENCE IN THE USE OF SOFTWARE TOOL

The first objective of this activity is to provide criteria to determine the required level of confidence in a software tool when applicable. The proposed method is based on the work described in [7] and relies on the following steps, see Figure 5. First, during workflow analysis, the entire development process of the product is structured into workflows (e.g., development of application software) and further refined into development activities (e.g., requirement elicitation). The purpose of that step is to systematically identify all the development activities and their dependencies as well as the respective tools involved. During the second step – determination of use cases – the activities identified previously are re-organized according to their respective tool. This step aims at identifying for each tool how the tool is used (the use case(s) for the tool). Note that the systematic approach for workflow analysis leads to a systematic identification of use cases for the tools involved in the development of the product. The third step is the identification of tool errors. During this step, a guidance is provided in order to identify all the possible tool errors for each use case. Finally, during the last step, the existing analysis and verification measures are mapped to the possible tool error in order to analysis the error detection capability of the tool chain. The following describes the four steps more in details.

ISO 26262 Compliance-Checklist for Management of Functional Safety

Requ. from ISO 26262		ASIL				Checklist				
ISO26262-ID	Description	A	B	C	D	Status	Finding / Actions	Remarks / Rationale	Responsible	Due Date
General Requirements according to Compliance										
SA_1: Overall Safety Management										
SA_2.1: Safety Management during concept phase and product development										
2-6.1	The first objective of this clause is to define the safety management roles and responsibilities, regarding the concept phase and the development phases in the safety lifecycle (see Figure 1 and Figure 2).	++	++	++	++					
2-6.1	The second objective of this clause is to define the requirements for the safety management during the concept phase and the development phases, including the planning and coordination of the safety activities, the progression of the safety lifecycle, the creation of the safety case, and the execution of the confirmation measures.	++	++	++	++					
2-6.2.1	Safety management includes the responsibility to ensure that the confirmation measures are performed. Depending on the applicable ASIL, some confirmation measures require independence regarding resources, management and release authority (see 6.4.7).	++	++	++	++					
2-6.2.2	The confirmation reviews are intended to check the compliance of selected work products to the corresponding requirements of ISO 26262;	++	++	++	++					
2-6.2.3	A functional safety audit evaluates the implementation of the processes required for the functional safety activities	++	++	++	++					

Figure 4: Checklist for monitoring and documenting the implementation of the ISO recommendations in the project

V. CONFIDENCE IN THE USE OF SOFTWARE TOOL

The proposed method is based on the work described in [7] and relies on the following steps, see Figure 5. First, during workflow analysis, the entire development process of the product is structured into workflows (e.g., development of application software) and further refined into development activities (e.g., requirement elicitation). The purpose of that step is to systematically identify all the development activities and their dependencies as well as the respective tools involved. During the second step – determination of use cases – the activities identified previously are re-organized according to their respective tool. This step aims at identifying for each tool how the tool is used (the use case(s) for the tool). Note that the systematic approach for workflow analysis leads to a systematic identification of use cases for the tools involved in the development of the product. The third step is the identification of tool errors. During this step, a guidance is provided in order to identify all the possible tool errors for each use case. Finally, during the last step, the existing analysis and verification measures are mapped to the possible tool error in order to analysis the error detection capability of the tool chain. The following describes the four steps more in details.

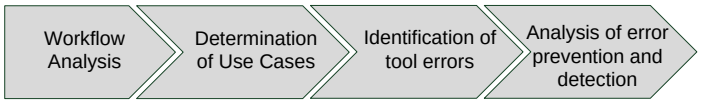

Figure 5: Main steps for classification of tool chains

Step 1: Workflow analysis

A workflow is a collection of different development activities and is part of the entire development process. Each development activity has a description of the actions, assigned roles, tools used for a specific action and defined input and output data for the tools. The information is collected by interviewing the development experts. There are several lists, which collect all the different information in specific spreadsheet tables:

Collection of all workflows in the project: Each workflow is represented with a single spreadsheet. It contains information on all development activities: a description of each activity, the used tools (tools for main performance and also supporting tools) and file data. Each file is named with its file ID (out from file list) and mapped to the specific activity as an input or output.

Tool List: In this spreadsheet table all essential tools for the project are listed. ISO 26262 requires for compliance the definition of following attributes:

- Identification and version number of the software tool
- Configuration of the software tool
- Use case(s) of the software tool
- Environment in which the software is executed
- The maximum ASIL among the safety requirements
- Methods to qualify the software tool, if required based on the level of confidence

File list: this table lists the project related input and output datafiles. Each file has a unique identifier and a short name with a description of file usage. The files are linked to the development activities. Each activity has file identifiers in its input and output column, and each file has links to the activities as input and output (see Figure 6). In this example the *technical requirements* (file *ASS_TecReq*, *PD_01)* is produced in the Workflow *analysis of software system requirements (ASS)* in the first activity and is input for the second activity in this workflow. There are two important attributes for the files, the *review* and the *multi* attribute. In the column "Review" of the file list all the files are marked, which have to be reviewed in development process. This is also shown in the workflow map , where a file to be reviewed is marked with a green flag on bottom of the file icon. In this case *ASS_TecReq* has to be reviewed in the first activity, thus it is marked in as green. In the column *multi* it is possible to discern, whether a file is a single one or a collection of files. Some activities produce as output a collection of file, and then it is easier to group them when analyzing tool confidence. For instance, when generating code, there are a bunch of different *c* and *h files*, which are produced in the specific activity. In order to avoid a huge amount of files in the file list, such produced output can be represented by a placeholder file with the attribute *multi* marked with an *m*. In the workflow map (Figure 6) such multi files have a special icon, which symbolizes a collection of files. Note that these two arguments *review* and *multi* can also be combined and both affect one file.

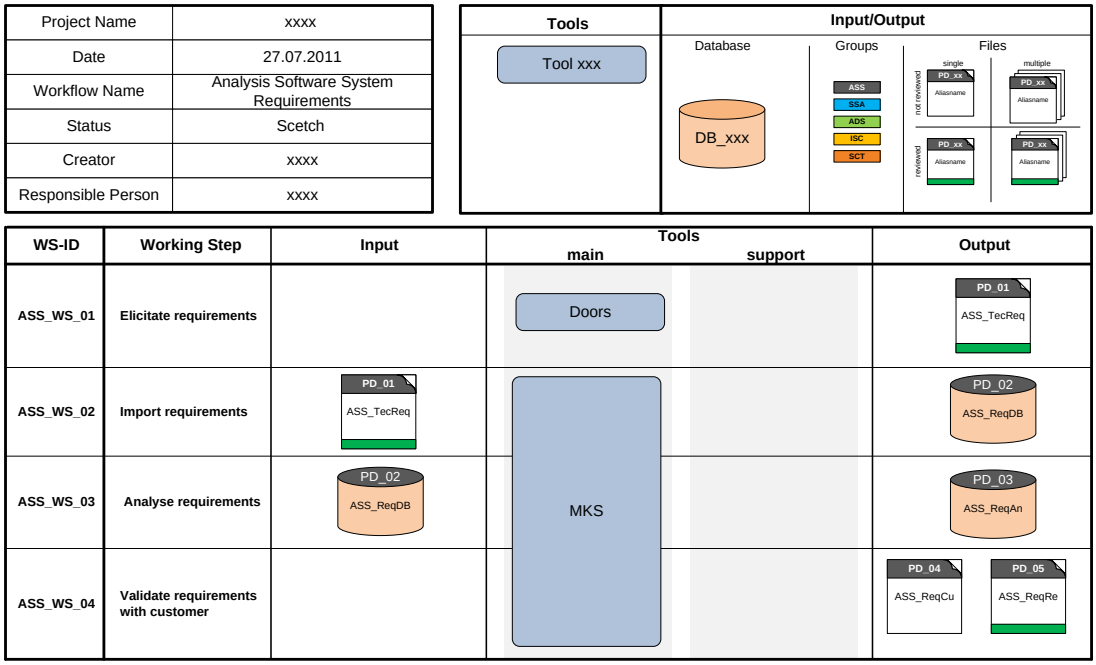

Figure 6: Systematic collection of the development activities and mapping to the output files as basis for tool classification

Step 2: Determination of use cases

A use case is defined as the user's interactions with a software tool or an applied subset of the software tool's functionality. Workflow analysis serves as basis for determination of the use cases. Detailing the use case needs deep knowledge on activities, thus it is inevitable to work out the use cases together with developers who are responsible for the activity. Each activity is the basis for an own use case for tool usage. However there is still a possibility that an activity itself has to be divided into two or more use cases. Especially when tool compounds are used for a single activity, such substeps have to be identified and broken down into single use cases. Figure 7 illustrates the activity "Validate requirements with customer" (*ASS_WS_04*), where a tool compound (*MKS, MS Excel*) is used. This activity is divided into two use cases, where both tools have the same input file *analysed requirements (PD_03*), but produce different output data (*ASS_WS_04_01* creates *analysed requirements by customer ⇨ PD_04* | *ASS_WS_01_02* creates *requirements review report ⇨ PD 05*). If e.g., a whole software tool compound is used to create two output files, the use case determination may lead to one use case per output file. In such a use case also intermediate results are denoted, that would not appear in the workflow step above. Identifying the complete list of use cases

are fundamental for the next phase.

Wf-Name	Analysis Software System Requirements				
Nr.	Working Step	Input	Tool	Tool-Support	Output
ASS_WS_04	Validate requirements with customer	PD_03	MKS	Excel	PD_04, PD_05
ASS_WS_04_01	Validate requirements with customer	PD_03	MKS		PD_04
ASS_WS_04_02	Validate requirements with customer	PD_03	Excel		PD_05

Figure 7: Detailing of an activity into two use cases

Step 3: Identification of tool errors

In this phase possible errors have to be identified within the atomic use cases. This is also done by interviewing responsible persons of each workflow step. In order to provide guidance for the efficient and systematic identification of tool error, a generic error model was developed with five error classes [7]. These error classes include not only errors produced by the tool but also errors, which are caused by the user. This gives an overview on all possible errors of a single activity, although errors produced by users are out of scope of tool classification. The following lists the error classes:

- ***E1: Interface error***. This error occurs, when opening or saving a file, so this error just takes place at the beginning or the end of an activity.
- ***E2: Processing error***. It appears when the tool is processing a routine and has some malfunction.
- ***E3: Process configuration error*** This error occurs when a tool is used according to an erroneous configuration (e.g., wrong parameters for a compiler)
- ***E4: Error in operating environment***. Such errors are caused by operating system, hardware and network failures.
- ***E5: Implementation error*** caused by the user. Such errors are not relevant for a tool classification, but for a holistic approach it is necessary to have an overview on all possible failure possibilities.

This collection of error classes is a fundamental basis for discussion of detection and prevention of possible errors. For each tool use case, the tool expert shall identify whether the error classes are relevant for the specific tool and which concrete error can be identified (e.g., implementation errors will be only relevant, when a tool is used for code implementation).

Step 4: Analysis of error prevention and detection

In the last phase countermeasures are derived for each use case. For the measures there are three different types of categories:

- Prevention: The error can be avoided by preventive measures due to the development process or configuration management. In an industrial context, the analysis of prevention measures must be based on existing documentation of process information.
- Review: The error can be detected by a review of work-products. In a rigorous analysis the review examines the availability of checklists for specific development steps and verifies the quality and completeness of the review protocols.
- Test: The error can be detected by a test with another software tool within the product-specific tool chain. The analysis of tests verifies the quality of performed tests, e.g., if test cases are generated systematically.

Next step is to estimate the probability to detect a software failure. At this point expert knowledge of responsible software developers is highly important. Such experts have the experience of ongoing projects, thus they are most suitable for error estimation. In the shown example the detection probability is classified in three levels that are directly mapped to tool error detection (TD) levels defined in ISO 26262 - Part 8. Similarly, tool impact level (TI) and resulting tool confidence levels (TCL) are determined. Note that at this point the reviews and tests activities already included in the development process can be taken into account for this analysis. Figure 8 shows a single use case with all relevant error classes and prevention measures. In the column “Details” there are descriptions, how a measure is executed. In this case the possible errors E2 and E5 can be detected with a review of the results (completeness on requirements has to be reviewed), E3 and E4 are avoided with a tested project configuration at the beginning of the project.

Wf-Position im V						TCL-Estimation									
Wf-Name	Analysis Software System Requirements					Error Possibilities		Countermeasures							
Nr.	Working Step	Input	Tool	Tool-Support	Output	Error Class (E1-E5)	Error Description	Prev.	Review	Test	Details	Propability of avoidance or detection of failures	TD	TI	TCL
ASS_WS_01	Elicitate requirements		Doors		PD_01	E1	Input error					no mistake possible, since data is already checked in in previous workflow step	n/a	n/a	n/a
						E2	Processing error		X		Due to software bugs of DOORS requirements can be corrupted/lost	Review done with a checklist, Review of requirements done by user	1	1	1
						E3	Error in process configuration	X			Erromous configuration could cause transcription errors	Error detection more likely and bigger then E1 and E2; Risks are minimized by a correct tool planning and deployment at beginnng of the project	1	1	1
						E4	Error in operating environment	X			Operation system can cause loss of data,	Error detection more likely and bigger then E1 and E2; Risks are minimized by a correct tool planning and deployment at beginnng of the project	1	1	1
						E5	Implementation error by user		X			Review of completeness of requirements	n/a	n/a	n/a

Figure 8: Example for error detection and prevention of a use case

VI. CONCLUSION

Functional safety management in the context of ISO 26262 is a challenging task due to the amount of activities and large number of requirements listed in the standard, as well as the size of the development teams (and number of organization) involved in the projects. The availability of the CESAR safety framework as knowledge data base is very useful for the systematic planning of the safety activities required in the context of ISO 26262. During this work, we have shown how this information can be used in an industrial context and how the tailoring for a company (which additional company internal information is required) can be performed. The resulting framework has been already used in several customer projects and was a central brick in order to synchronize the development activities between the different partners and finally for the success of the projects.

ACKNOWLEDGMENT

The research leading to these results has received funding from the ARTEMIS Joint Undertaking under grant agreement Nr 100016, the Austrian BMVIT under the program "Forschung, Innovation und Technologie für Informationstechnologien" and from specific national programs and / or funding authorities.

REFERENCES

[1] ISO 26262: “Road vehicles – Functional safety”, 2011.
[2] IEC 61508 Edition 2.0: “Functional Safety of Electrical/ Electronic/ Programmable Electronic Safety-Related Systems, part 1-7‟, 2010.
[3] G. Griessnig, et al. (2010). “CESAR: Cost-Efficient Methods and Processes for Safety Relevant Embedded Systems”. In Embedded World 2010- ARTEMIS Session
[4] JP Blanquart, et al “A multi-domain platform for safety process methods and tools for critical embedded systems”. ERTS² 2012
[5] M. Krammer, N. Marko, E. Armengaud, D. Geyer, G. Griessnig: “Improving methods and processes for the development of safety-critical automotive embedded systems”. ETFA 2010: 1-4.
[6] C. G. Billich, Z. Hu, “Experiences with the Certification of a Generic Functional Safety Management Structure According to IEC 61508”, in Proc. 28th International Conference, SAFECOMP 2009, Hamburg, pp. 103–118.
[7] J. Hillebrand, et al. (2010). “Establishing Confidence in the Usage of Software Tools in Context of ISO 26262”. SAFECOMP 2011: 257-269
[8] M. Conrad, J. Sauler, P. Munier: Experience Report: Two-Stage Qualification of Software Tools. In: Proc. 2. EUROFORUM ISO 26262 Conference, Stuttgart, Germany, Sept. 27-28, 2010
[9] DO178B---RCTA/DO-178B/ED-12B, "Software Considerations in Airborne Systems and Equipment," Federal Aviation Administration software standard, RTCA Inc., December 1992
[10] CENELEC (2001): Railway Applications - Software for Railway Control and Protection Systems. EN50128
[11] Michael Beine: „Sichere Software durch Werkzeuge und normgerechtes Entwicklungsvorgehen, Automotive Conference 2010 Stuttgart
[12] Mirko Conrad et al. „Software Tool Qualification According to ISO 26262” SAE international, May 2011