2016 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

Institutionalizing a certified ISO26262 safety process

Experiences and lessons learned

Michael Rohleder, Clemens Röttgermann, Marcus Müller
AMP/New Product Development Center
NXP Semiconductors, Munich, Germany
{*michael.rohleder ,clemens.roettgermann, marcus.mueller }@nxp.com*

***Abstract*—Functional safety in the context of the ISO26262 standard is an essential feature for any semiconductor device targeting the automotive market. Key aspects to also cover corresponding requirements by a safety-aware" development process are identified. It is intended to provide an overview of the required adaptations, but also to discuss subsequent enhancements and lessons learned from the application of this process.**

Keywords—ISO26262, Functional Safety, certification, development process, improvement, tool confidence

I. INTRODUCTION

Freescale (now NXP) has a long history of developing customer-specific devices for safety-critical automotive applications, like e.g. controllers for airbags or braking systems. Shortly after the ISO26262 standard (official title: *"Road vehicles – Functional safety"*) has been released, the Functional Safety program SafeAssure™ [1] has been announced, which is based on an internal safety process. The initial development of the related, internal safety process occurred in parallel and is based on experiences made during the development of the MPC5643L, a microcontroller that achieved a formal ISO26262 certification. This contribution identifies key aspects for addressing associated requirements by extending an existing semiconductor development flow. As such it provides an overview of safety-oriented process elements needed to establish a company-wide generic process for the development of safety devices. Subsequent enhancements and lessons learned during the institutionalization of this process are discussed.

II. SAFETY MANAGEMENT

The *Safety Plan* documents the planning and coordination of all safety activities, especially item integration and testing, validation and software verification, and a functional safety assessment. As such the Safety Plan shall be either included or referenced in the project plan. A template for the Safety Plan has been developed, which specifies the complete set of safety activities defined by the SafeAssure™ process, and the corresponding mapping to the standard NXP semiconductor development flow. Additional tailoring is possible by adapting this template to the specifics of a particular project. Distributed developments require the definition of a *Development Interface Agreement (DIA).*

The *Safety Case* progressively compiles the generated work products, where all work products are subject to *configuration* and *change management* and *documentation*. Corresponding safety activities and the resulting work products may be tailored (i.e. omitted or performed in a different manner). A template for the Safety Case has been made available, which specifies the usual set of work products developed, used, or referenced when following the SafeAssure™ process and its mapping to the ISO26262 standard; along with the corresponding safety argument, the associated list of deliverables, their location within the CM database, and specifics of the associated information flow. Also for this document, additional tailoring is possible by adapting this template to the specifics of a particular project.

A. *Inclusion of safety activities within the development flow*

Safety activities are made an integral part of the development flow, by including the need to supply corresponding work products at the phase gates within this flow. For every phase gate, the standard development flow specifies checklists to ensure the quality of the required deliverables. A quality management system (QMS) tool support the creation and management of these checklists throughout the development lifecycle

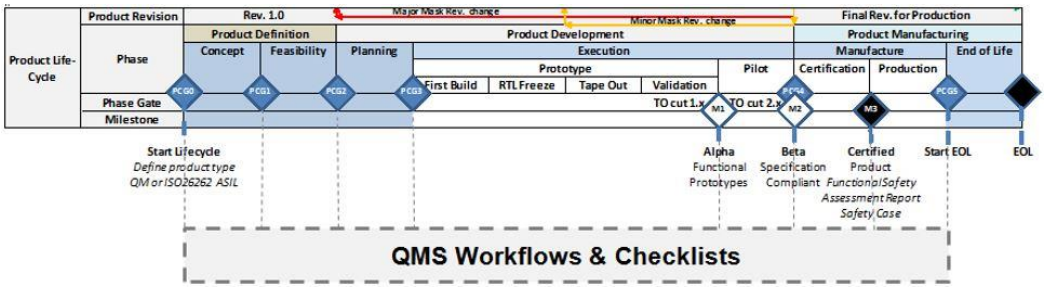


Figure 1, "Standard development flow and associated phase gates"

These checklists have been extended for projects having an ISO26262 requirement to ensure also the quality of the functional safety deliverables. This is done in a way to ensure the requirement of the ISO26262 standard that "safety activities must be distinguishable".

B. *Confirmation Measures, Techniques or Measures*

The ISO26262 specifies and requires a set of confirmation measures, where a specific type of assessment (referred to as *review*, *audit*, or *assessment*) and degree of independency (I0: *optional*, I1: *by a different person*, I2: *by a person from a different team*, I3: *by a person from a different department or organization*) is defined for every work product specified within the standard. Confirmation measures are performed to check the correctness w.r.t. *formality*, *content*, *adequacy* and *completeness.* Whenever possible, corresponding aspects have been formalized within the checklists provided by the QMS. In case this was not possible, appropriate templates are provided.

Additionally, the ISO26262 requires to specify the *techniques* or *measures* used during the various stages of the product development to achieve compliance. For this purpose a standard checklist has been generated that has to be filled with the appropriate information for a particular product. Many elements within this checklist are related to some standard mapping made possible by a common design and verification flow fully supported by tools.

III. CONFIGURATION AND CHANGE MANAGEMENT

One important requirement of the ISO26262 standard is the definition of an appropriate configuration management and change management process. In this context

- *configuration management* ensures that work products, and the principles and general conditions of their creation can be uniquely identified and reproduced in a controlled manner at any time.
- *change management* is responsible for analyzing and control changes to safety-related work products throughout the safety lifecycle.

Figure 2, "Generic Configuration Management and Change Management Plan"

For this purpose a corresponding document "*Configuration Management & Change Management Plan*" has been created for the MPC5643L device. When defining the SafeAssure™ process the content of this document has been generalized to serve as the basis for the creation of a generic document that specifies all related requirements and processes.

IV. CONFIDENCE IN USE OF SOFTWARE TOOLS (TOOL SUITABILITY)

A. Requirements defined by the ISO26262 Standard

The ISO26262 does further specify the need to determine the level of confidence in the use of a software tool, when applicable, to ensure the user can rely on the correct function of a software tool for those activities or tasks required by the standard. The intent is to minimize the risk for systematic faults in the developed product due to malfunctions of the software tool. For this purpose, the standard requires an appropriate planning of the tool usage and an evaluation of the tool by analysis of

- the possibility that a malfunction can introduce or fail to detect errors in a related item (expressed in classes of Tool Impact: TI1 and TI2), and
- the confidence in preventing or detecting such errors in its corresponding output (expressed in classes of Tool error Detection: TD1, TD2, and TD3).

Confidence in prevention or detection can be accomplished through process steps (i.e. by performing an equivalence check after the synthesis process), redundancy in tasks or software tools (i.e. performing the same task redundantly by different tools) or by rationality checks. For this purpose, measures internal to the software tool, as well as measures external to the software tool (e.g. guidelines, tests, or reviews) implemented in the development process of the safety-related item or element can be considered and assessed.

Table I. Determining the Tool Confidence Level (TCL) in accordance with the ISO26262

Tool Confidence Level		**Tool error Detection**		
		TD1	**TD2**	**TD3**
Tool Impact	**TI1**	*TCL1*	*TCL1*	*TCL1*
	TI2	*TCL1*	*TCL2*	*TCL3*

This evaluation has to be performed in the context of a development process that is adequate with respect to compliance with ISO26262, if activities or tasks required by this standard rely on the correct functioning of the software tool used. This applies to all commercial tools, but also to open source, freeware, or shareware tools as well as to tools that have been developed in-house. The evaluation can be performed independently from the development of a particular safety-related item or element (i.e. by a cross-organizational activity), when the determined tool confidence level is confirmed for this purpose.

When the determined tool confidence level indicated a need, then appropriate qualification methods, dependent on this tool confidence level and the maximum ASIL of all safety requirements allocated to the item or element that is to be developed using the software tool, have to be applied. Permitted qualification methods are:

1. Increased confidence from use
2. Evaluation of the tool development process
3. Validation of the software tool
4. Development in accordance with a safety standard; e.g. ISO26262, IEC61508 or RTCA DO-178.

B. Tailoring

Many aspects of the tool evaluation effort defined within the ISO26262 standard require some tailoring for the kind of development process used for semiconductor devices and its associated tools. The usual design flows exhibit a significant amount of creation steps that are followed by a corresponding verification step, which is a bonus for the tool error detection. On the other side, the software tools used for this purpose are regularly updated to account for new developments, increased technology needs, but also to fix bugs that have been found earlier. Those frequent regular updates make the used tools set often unique for every project and its product being developed.

During the development of the MPC5643L an evaluation methodology has been developed, which provides some stringent classification criteria for the usage of software tools within the design flow, and the associated tool impact. It also defines further details for determining the confidence level for tool error detection in the context of an usual semiconductor development flow; also taking into account the specifics of the tools developed by the EDA industry. This methodology has been successfully used for the tool suitability evaluation of this product. Since its argumentation is fully tailored towards the specifics of EDA tools used for the development of semiconductor devices and the corresponding flows, it cannot be used for other domains like the development of embedded software or similar related areas also in need for some tool suitability evaluation.

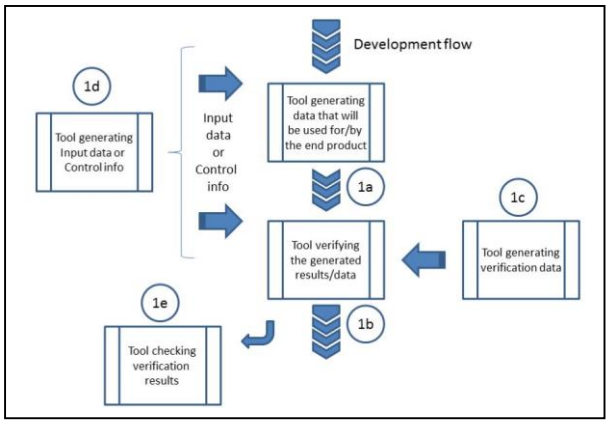


Figure 3: tool suitability analysis in the context of the design flow

C. Tool Support

Because this tailoring of the evaluation methodology is associated with the development flow used for the semiconductor device, and many flow steps or tools are often similar or equivalent between projects, there is a high potential for reuse, especially with respect to:

- Tool related information
- SoC related information (small differences between derivatives)
- Argumentation related to a flow step or a specific tool usage

To enable some cross-organizational reuse of the associated data, the tailoring methodology and tool assessments have been captured in a tool supported database. The intent of this database is to enable the reuse of the corresponding information, either to confirm its applicability for a project, or to use it as the starting point for a similar assessment effort.

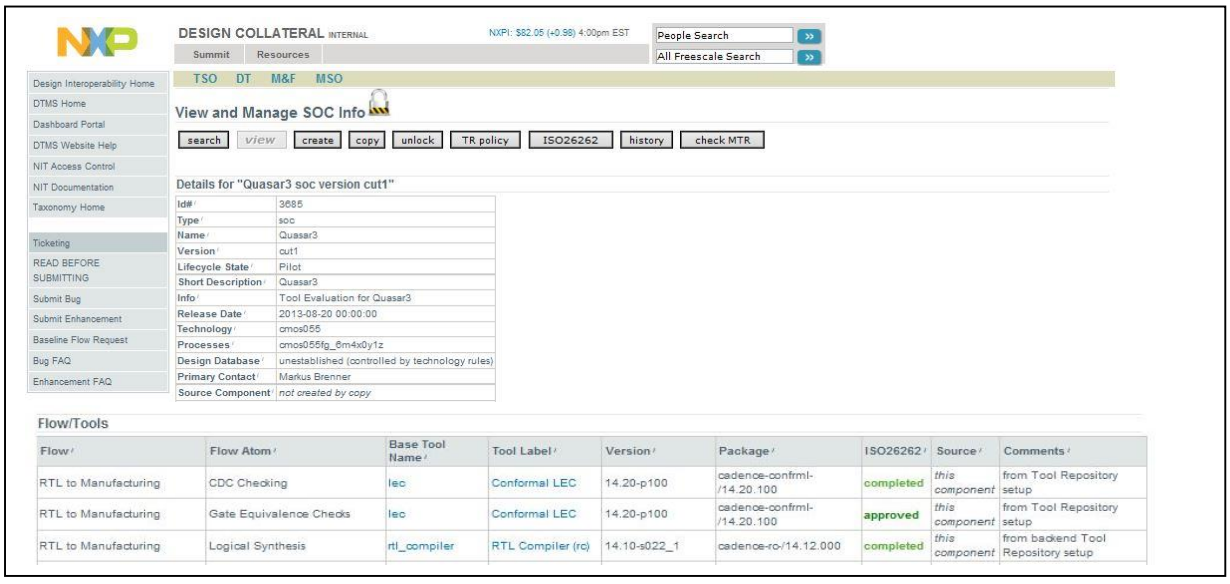
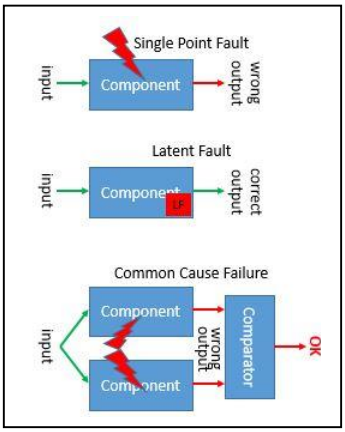


Figure 4: Tool supported database for performing the tool suitability assessment

In a later certification of another SafeAssure™ device by another certification body, this database-supported evaluation methodology has been ranked as "*pushing the state-of-the-art for this kind of assessment*".

V. HAZARD ANALYSIS AND RISK ASSESSMENT

ISO26262 requires hazard analysis and risk assessment. This is based on bottom up and top down approaches. The bottom up method is FTA (*Fault Tree Analysis*), but for a Safety Element out of Context (SEooC) like a general purpose MCU is the focus on bottom up FMEDA (*Failure Modes, Effects, and Diagnostic Analysis*). This is a systematic analysis technique to obtain subsystem/product level failure rates, failure modes and diagnostic capability [2] for the detailed determination of error causes and their impact on the systems. It defines failure rates λ, failure modes, and failure mode distribution. Quantitative numbers for failures rates of SPF (single-point-faults), LF (latent-faults) and common cause failures need to be provided. These types of faults and failures are briefly illustrated in the figure right-hand side. The ISO26262 specifies the diagnostic coverage which needs to be achieved for a specific ASIL level.



Let's take a closer look at the diagnostic coverage. A problem is that it is highly dependent on the diagnostics measures which a customer enables when using a SEooC. Examples here are: Is the lock-step enabled? Is the clock monitoring enabled? Is the LVD/HVD detection enabled? Are the MPU's enabled? This requires re-calculation of the FMEDA for all different use-cases which turned to be a significant effort. The NXP approach to that problem is the *Dynamic FMEDA*, a tool which assists Field Application Engineers to quickly support the customers with a FMEDA tailored to their actual application and environment.

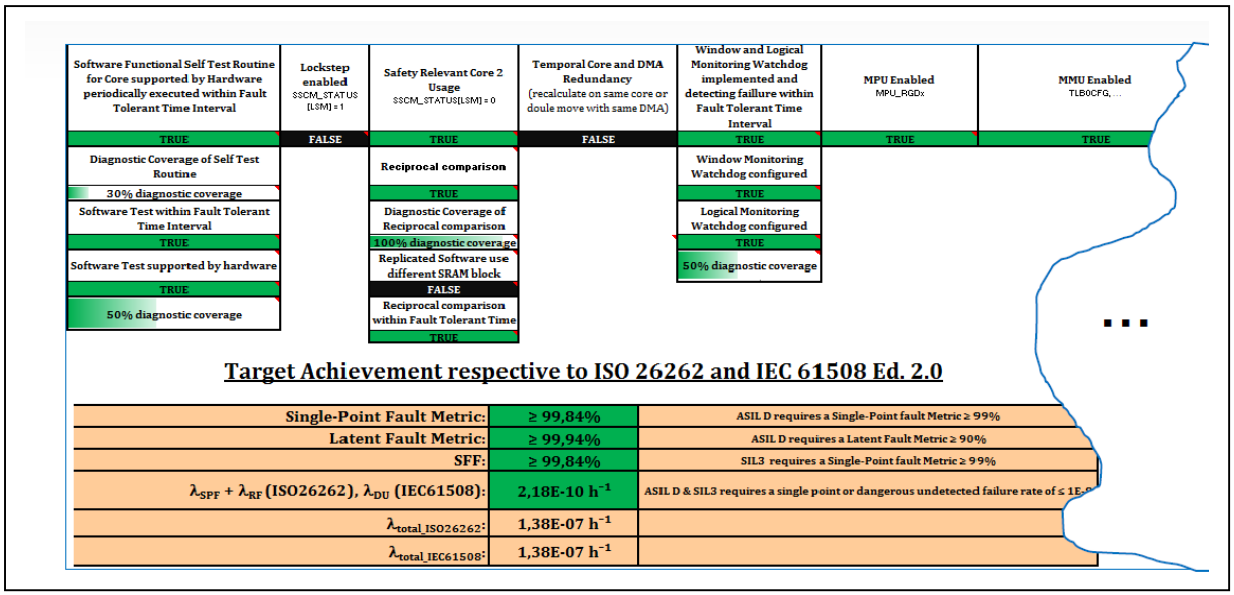


Figure 5, "Dynamic FMEDA (Failure Modes, Effects, and Diagnostic Analysis)"

VI. SAFETY REQUIREMENTS : CONCEPT REFINEMENT

The ISO26262 specifies characteristics for safety requirements, which are *unambiguous and comprehensible*, *atomic*, *internally consistent*, *feasible*, and *verifiable*. It also requires safety requirements to be *unambiguously identifiable* and *allocated* to an item or an element. Additionally, the standard requires a set of safety requirements to be *complete*, *externally consistent*, and *maintainable*, have a *hierarchical structure* exhibiting *no duplication of information* within any of its levels, and an *organizational structure* according to an appropriate grouping scheme. The ISO26262 also requires every safety requirement to have a set of *attributes*, including

- an unique *identification* remaining unchanged throughout the safety lifecycle
- a *status* (i.e. *proposed*, *assumed*, *accepted*, *reviewed*), and
- an automotive safety integrity level (*ASIL*).

The above listed definitions within the ISO26262 standard result in the need to derive lower level safety requirements with a well-defined methodology that applies additional information whenever this is made available. For this purpose, three documents have been defined that provide the additional information for performing the corresponding refinements, as shown in Figure 7, "Refining and assigning safety requirements:

- *Safety Concept*: specifies conceptual decisions, derives lower level safety requirements, and defines the safety measures and mechanisms to be implemented by architectural elements.
- *Architecture*: specifies the mapping to a specific SoC architecture (which may be shared across SoCs).
- *SoC Guide*: specifies the implementation by a (set of) specific hardware features.

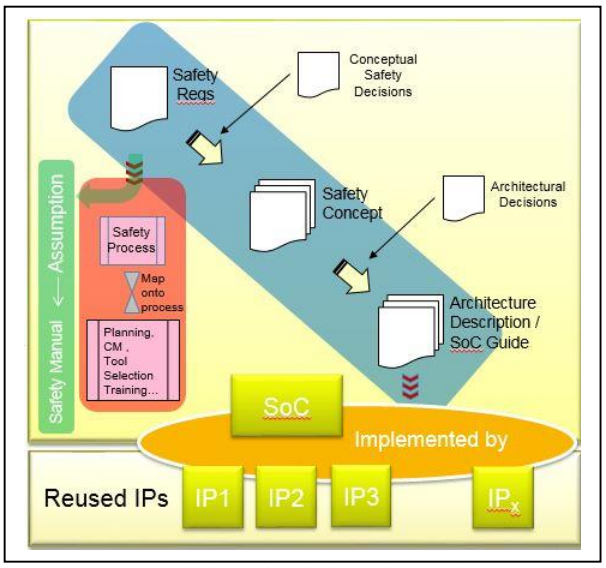


Figure 7, "Refining and assigning safety requirements"

Derived requirements inherit their ASIL from the safety requirements from which they are derived, unless there is a decomposition into redundant safety requirements implemented by sufficiently independent elements.

Conceptually, a safety element out of context (SEooC) permits three methodologies to fulfill safety requirements:

- by mapping them to the development process; the majority of the corresponding requirements have now been made an integral part of the standard development process for SafeAssure™ products. Additional, SoC specific needs may need to be reflected in the development process for a particular product.
- by mapping them to assumptions, which provide information about the assumed environment, expected software behavior, etc. The standard requires that corresponding information and usage assumptions have to documented in the Safety Manual
- by mapping them to one or multiple architecture elements within the SoC. Corresponding hardware features may be implemented at the SoC level (by combination or interaction of two or multiple blocks) or at the module level.

For all derived and mapped requirements, the ISO26262 standard requires that the corresponding safety requirements are traceable. At least the following traceability information must be provided for every safety requirement:

- each source of a safety requirement at the upper hierarchy level,
- each derived safety requirement at a lower hierarchical level, and
- the specification of verification

VII. VERIFICATION, VERIFICATION, VERIFICATION

A. *Objective and Verification Requirements*

In the terminology of the ISO26262, the terms *verification* and *requirements* are used extremely broad. Why, and how is explained in the following section; which intends to shed some light on the usage of requirement within the ISO26262. First of all, according to the ISO26262, "*the objective of verification is to ensure that the work products comply with their requirements".* This statement does already relate to two different kind of **requirements**:

- safety *requirements*, where traceability down to the implementation is required, but also
- verification *requirements*, which intend to ensure the correctness (i.e. by a pass/fail criteria), completeness (i.e. by specifying the expected verification coverage), and consistency

Furthermore, when the ISO26262 specifies the verification *requirements* in relation to the development phase of an item or element, it also uses the term verification very broadly:

- During the **(product development) design phase**, "*verification* ensures the work products comply with previously established requirements for *correctness*, *completeness*, and *consistency"*. In the semiconductor industry portions of this kind of verification are usually named functional verification, but there are further *verification* activities like equivalence checking, design and electrical rule checks, etc. that are also required to fulfill these *requirements* of the ISO26262 standard.
- During the **(product development) test phase**, "*verification* is the evaluation of the work products within a test environment to ensure that they comply with their requirements". In the semiconductor industry this term relates to the verification of the fabricated product, which is often referred as *validation* work. As this kind of verification relates to a different work product, it requires a separate, complete traceability from the safety requirements down to their implementation and verification.
- During the **production and operation phase**, "*verification* ensures that *safety requirements* are appropriately realized in the *production process*, *user manuals* and *repair and maintenance instructions*; and the safety related properties of the item are met by the application of control measures within the *production process"*. In the semiconductor industry this work not only relates to the production test, but also to related procedures intended to ensure a constant quality of the work products; as well as to appropriate control means for the maintenance work.

And, last but not least, the ISO26262 is very strict in the requirement to provide evidence that the corresponding tests are "*planned, specified, executed, evaluated and documented* in a *systematic manner"*.

B. Verification Documentation

The documentation to be provided as evidence for an appropriate verification has to cover aspects of *verification planning*, provide a *specification* of the verification work, and a *verification report* detailing the results. For the verification planning, the ISO26262 requires to specify:

- content of the work product to be verified
- the methods used for verification,
- some pass/fail criteria,
- the specification of verification environment in accordance with the safety requirements,
- the specification of the tools used (tool suitability effort)
- the definition of a regression strategy (how verification is repeated after changes have been made)

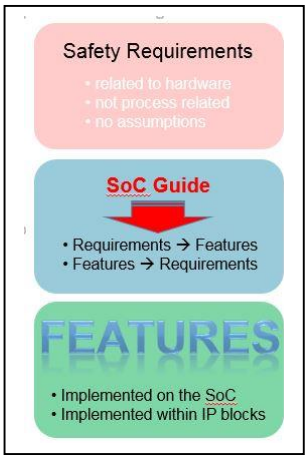
For the verification planning, the work product complexity, prior experiences, and the degree of maturity (e..g. of the used IP in form of an IP maturity assessment, but also for the used software tools and methods) are to be taken into account. For the specification, the ISO26262 requires some exact definition of the test to be performed; including an unique ID of the test, configuration management information for the work product to be verified, preconditions and configurations as well as environmental conditions. Furthermore the verification methods to be used (e.g. review/analysis, simulation, test, …) and the associated input data and, if applicable the expected behavior must be documented.

VIII. FEATURE VERIFICATION

Mapping of hardware related safety requirements to one or multiple architectural elements must be properly verified; of importance is here the correctness, the completeness, and consistency of this verification. Furthermore their implementation is based on reusable IP modules that are combined within a SoC, where either an individual IP module is providing corresponding features, or a feature is implemented by combining one or multiple IP modules in a specific manner.

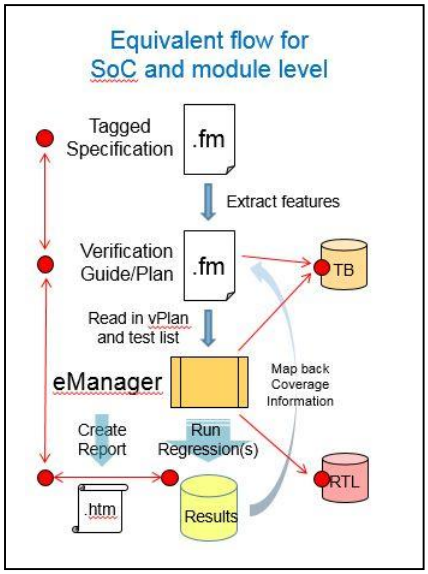
This enables a certain mapping of safety requirements to hardware features that have been implemented within an IP or the SoC:

- Every safety requirement covered by hardware is mapped to one or more hardware feature(s); e.g. **REQ101 → FeatABC**, **REQ101 → FeatXYZ**
- One hardware feature may be required to fulfill one or multiple safety requirements; e.g. **FeatXZY ← REQ011**, **FeatXZY ← REQ042**



Mapping of safety requirements to hardware features as shown in the figure at the right side permits the reuse of existing feature tags, which are standard methodology and usually exist already; at least for a reusable IP block. Subsequently, existing tests can be reused when an association between such a feature and a test for verification purposes does already exist. It is worth to note that for one safety architecture a feature might be related to a safety requirement, while for another safety architecture the same feature might not relate to any safety requirement.

The benefits of mapping safety requirements to implementation features are two-fold. First, it enable the reuse of feature tags that have been defined for any hardware elements that have been implemented. There is a complete infrastructure built around such feature tags, and the capability to reuse this is very beneficial. Second, it enables the reuse of existing verification capabilities with respect to those feature tags for functional safety purposes. Since coverage of features associated with safety requirements can be combined with coverage of non-safety features, the only missing proof is the coverage of all safety requirement related features. For this purpose, the capability to associate a feature tag to a test and its test result is essential, and accomplished by many existing tools. Based on this capability, we have achieved full traceability by combining 3rd party tools for requirements management – for the refinement of safety requirements and their mapping to features – with standard verification feature coverage tools – to manage the feature coverage for verification purposes within the SoC and module verification space.



Having the capability to reuse the feature tags also for the verification of safety requirements enables a flow that performs feature coverage "the classical way". For this purpose, tagging is done as usual by directly inserting feature tags within the specification document(s), which is usually referred to as "spec tagging".

This feature tags are then extracted from the specification documentation and matched with related information made available within the verification documentation. Associated verification tests are run under control of a regression tool; which usually determines the test result from the associated pass/fail criteria and maps this back to the verification plan and specification to generate the corresponding coverage information for a regression run.

There is already existing infrastructure that fully supports such a verification flow; reusing these capabilities also for the verification and tracing of safety requirements down to the test results is very beneficial and permits to reuse an existing and proven environment. It further permits to associate the generated test logs with the corresponding test.

IX. SUMMARY AND OUTLOOK

The origin of the described process has been developed in parallel to the MPC5643L device; both efforts have started years before the ISO26262 standard has been released (Nov 15th, 2011), and culminated in the successful certification of the MPC5643L [3] (certificate issued September, 6th 2012) by Exida [4], an independent accredited certification body. The MPC5643L is the *world's first microcontroller to achieve a formal ISO26262 certificate* for all ASILs, up to and including ASIL D [5]. Based on this success, Freescale (now NXP) has defined SafeAssure™ [1], a program that features a variety of MCUs, processors, analog and power management ICs and sensors intended to provide a simpler way to comply with standards for functional safety at the system level.

To enable the efficient development of related devices, the existing company-wide development process had to be enhanced to cover also requirements for functional safety and associated work products. This has been accomplished by enhancing or providing additional existing procedures, associated templates and checklists as described. For this purpose, many of the work products from the MPC5643L and associated process development have served as a basis. Enhancement proposals based on experience gathered during this development have been implemented to further improve the productivity, automate steps where possible, or provide further support by tools or databases. We have also depicted some of the related work in this paper.

The above steps have resulted in institutionalizing organization-specific rules and processes, which are executed as part of the development of a semiconductor device. The maintenance of this process is an ongoing and continuous effort, which fits very well the requirement of the ISO26262 standard to *create, foster, and sustain a safety culture*.

REFERENCES

[1] http://www.nxp.com/SafeAssure/

[2] https://en.wikipedia.org/wiki/Failure_Modes,_Effects,_and_Diagnostic_Analysis

[3] http://www.exida.com/index.php/Resources/SAEL_Detail/freescale/

[4] http://www.exida.com

[5] http://www.reuters.com/article/idUS104920+06-Sep-2012+BW20120906