SAE INTERNATIONAL

Writing Good Technical Safety Requirements

2016-01-0127

Published 04/05/2016

Agish George, William Taylor, and Jody Nelson
KVA

CITATION: George, A., Taylor, W., and Nelson, J., "Writing Good Technical Safety Requirements," SAE Technical Paper 2016-01-0127, 2016, doi:10.4271/2016-01-0127.

Abstract

One of the key premises of the ISO 26262 functional safety standard is the development of an appropriate Technical Safety Concept for the item under development. This is specified in detail in Part 4 of the standard - Product development at the system level. The Technical safety requirements and the technical safety concept form the basis for deriving the hardware and software safety requirements that are then used by engineering teams for developing a safe product. Just like any other form of product development, making multiple revisions of the requirements are highly undesirable. This is primarily due to cost increases, chances of having inconsistencies within work products and its impact on the overall project schedule. Good technical safety requirements are in fact the foundation for an effective functional safety implementation. Presently the ISO 26262 standard does not provide any direct guidance on any specific method to derive technical safety requirements for a given safety goal for an item. This paper provides guidelines to come up with a comprehensive and concise set of Technical Safety Requirements using safety analyses techniques like FTA or FMEA. The paper is intended to support those safety engineers tasked with developing the technical safety concept. Additionally, the paper recognizes that in practice projects face challenges such as lack of stakeholder interest, multi-party development and missing or incomplete upstream work products. The paper captures these real world challenges and provides proposed solutions. The paper concludes by citing a few methods for Fault tolerant Time Interval (FTTI) determination at the ECU level; a key parameter that is critical for the effectiveness of the technical safety concept.

Introduction

The ISO 26262 standard provides the framework to specify, design, implement, integrate, verify and validate automotive electrical and electronic (E/E) systems that are safe from systematic as well as random hardware failures. Since its introduction in final form in 2011, the ISO 26262 standard has been widely adopted in the automotive industry for achieving functional safety. There are requirements laid down by the standard for each sub-phase in the product safety lifecycle - management, conceptualization, product development, production and operation.

The standard comprises 10 parts - Part 1 is practically a glossary on the vocabulary used in the standard. Part 2 of the standard focuses on the need for an organization wide safety culture and the safety management during the product life cycle. Part 3 or the concept phase is where the item is defined along with the hazards and the ASIL rating that goes along with it. This is also where the functional safety concept for the item is defined based on the preliminary architecture assumptions.

Parts 4, 5 and 6 form the core of product development at the system, hardware and software level respectively. The focus of Part 4 clause 6 is on the specification of the technical safety requirements. The technical safety requirements form the foundation for product safety requirements downstream at the hardware and software level. One of the keys to great product quality is to have a good requirements definition.

Part 8 clause 6 specifically speaks about attributes and characteristics of safety requirements and how to manage safety requirements. Part 7 defines requirements for production and service.

The standard does not talk about specific techniques to be used for deriving technical safety requirements from the functional safety concept. The paper attempts to provide a safety engineer a guideline to be able to consistently derive technical safety requirements using well know safety analyses methods.

Objectives

The objectives of this paper are:

1. Discuss the importance of the technical safety requirements in the safety strategy
2. Discuss some practical difficulties faced by real word projects and propose some measures to minimize the impact of those.

3. To present a methodology for relatively new safety engineers to develop technical safety requirements using safety analyses like FTA.
4. To briefly outline some methods for determining fault tolerant time interval at the ECU level.

Importance of Good Technical Safety Requirements

The technical safety requirements are basically, a safety engineer's, understanding on what needs to be done to ensure the item or part of the item (e.g. microcontroller or SW objects) in question achieves functional safety at the desired ASIL level. A particular design approach is not mandated for deriving TSR's. It is key to have TSR's written in a clear, concise and unambiguous language and notation.

A well written TSR serves the following purposes:

1. **Basis for Safety Concept** The technical safety requirements and their allocation to the relevant hardware and software elements form the basis for the technical safety concept for the item. Weakness in the technical safety requirements and their ability to achieve the desired ASIL level will weaken the safety implementation.
2. **Input for Hardware and Software work products** The technical safety requirements along with the technical safety concept are key inputs to the system design, product development at the hardware level and the product development at the software level. Any ambiguity or errors that arise in the technical safety requirements risk being percolated to downstream work products like the hardware safety requirements and software safety requirements. Good product quality is not a coincidence and comes from a strong requirements engineering culture. Recalls and service updates are very costly and can wipe out profits from a product line.
3. **Input for Item integration and testing** Part 4 clause 8 of the ISO26262 standard discusses Item integration and testing. Integration itself refers to integration at multiple levels viz. hardware-software, system and vehicle. The tests at each level are intended to provide sufficient confidence that the technical safety requirements are capable of preempting unintended behaviors that could violate a safety goal.
4. **Cost for product development** A preliminary draft of the technical safety concept and the technical safety requirements would serve as a clever project management tool for effort estimation and hence cost estimation. Just like any other aspect of product development late changes, scope creep and rework in the technical safety requirements can result in cost overruns and delay to the overall project schedule.
5. **Customer Confidence** In practice the technical safety requirements may be completely developed by the Tier1 supplier or there may be a work split between the Tier 1 and OEM. The actual work-split and work-products are detailed in an ISO work product known as the DIA (Development Interface Agreement). A well written technical safety concept (TSC) and technical safety requirements ensures that the multiple parties involved understand the approach and are working towards a viable implementation. This invariably helps in maintaining and even boosting a customer's confidence in the capabilities of a supplier.

For the reasons listed above it is imperative to have well defined technical safety requirements.

Challenges in practice

This section tries to discuss some of the practical difficulties faced in actual projects while trying to implement ISO26262.

1. **Missing upstream work product** The functional safety concept is the specification of the functional safety requirements with associated information, their allocation to architectural elements, and their interaction necessary to achieve the safety goals. According to ISO26262 Part 4 Clause 6.4.1, Technical safety requirements need to conform with the functional safety concept developed during the concept phase. However it is possible that a specific project comes across the situation where the FSC is missing and it is not within their work split responsibility. In this scenario the FSC needs to be reverse engineered or tailored in to complete the safety case. Another scenario is where the FSC is available but the functional safety requirements are not elicited properly and results in a kind of partial work product. Both these scenarios are undesirable and beat the purpose the standard intended in having this work product. *Suppliers should ensure through a DIA that upstream work products are delivered on time by a mutually agreed party*.
2. **Multiparty development** It is not uncommon today to have products that are developed by multiple organizations. This is especially true for software development. It is not unusual to see products where the OEM develops high level application SW, the Tier 1 is providing most of the Base software and a third party supplier provides the CAN stack. In these situations it is key to get the work split and schedule captured in the DIA and agreed by all parties. Lack of collaboration and team work across various teams working on the same product can be detrimental to safety verification and validation exercises for the product. *Once again attention to identify multi party responsibilities and timelines through a DIA early on in the project should help contain this topic*.
3. **Challenges with legacy systems** In recent years OEM's have felt a strong need to get more and more control units to meet ISO26262 standards. With model year updates it is typical that most electronic control units in the vehicle maintain status quo or are changed minimally. However new ECU's or ones with engineering changes may decide to take steps to comply with ISO26262. However there are typically inputs which an ECU may receive from a sensor or CAN which may become a part of the safety concept and thereby get ASIL requirements cascaded to them. This drives changes in the ECU's or hardware components sending the signals. However an OEM may defer making these changes imminently due to a decision to carryover legacy components. *In such a case the deficiencies in the system*

need to be clearly communicated and mutually agreed and documented in the functional safety assessment. Also any legal liabilities if any that may arise due to these deficiencies should be documented and agreed.

4. **Critical computer resources** This can be seen in the same context as legacy systems. Microcontrollers used in carry over ECU hardware may have inadequate resources like available run time, memory partitioning capabilities, free RAM or Flash to implement the safety control strategies required to meet the target ASIL for the item. Sometimes challenges like runtime or CPU throughput can be overcome by careful selection of the safety strategy where there is such an option. Eg - choosing an ADC based feedback may take less CPU throughput compared to timer interrupt based one. Suitability of the intended hardware for implementing safety concepts appropriate for the item should be evaluated at the request for quote phase. *This would help all parties capture a realistic picture of the cost, time and effort that would be involved in implementing ISO26262 afresh in an item.*
5. **Stakeholder lack of interest** The successful implementation of ISO26262 is subject to the cooperation and teamwork of all stakeholders involved including OEM, Tier1 and any sub suppliers or third parties. The Tier1 can have a tough time developing a product in compliance to ISO26262 if an OEM's commitment is low. The challenges vary from missing safety concept work products like HARA and FSC to inability to perform safety validation. *As mentioned for point 1 and 2 safety management that kicks in from project start and that relies on an multi party encompassing DIA can help relieve this issue. On the other hand if the implementation of ISO26262 is solely in the interest of the supplier then these missing work products and the role of the OEM can be simulated internally by the supplier.*

Writing technical safety requirements

This section demonstrates derivation of technical safety requirements using some systematic methods like Fault Tree Analysis or FTA. The possibility of also using the FMEA method is briefly shown too.

The methods stated above are merely techniques for eliciting technical safety requirements. The technical safety requirements still need to be consistent with the functional safety concept and implement them completely without any contradictions.

For the sake of discussion and application of the methodology we will take the example of a relatively simple item like a Rear Gate module. The primary function of a Rear Gate module would be to open or close the Rear Gate (Boot lid) based on a valid request from the driver. The open or close request could come from a hardwired digital input switch or a CAN message. The request would be interpreted as an open or close based on the current position of the boot lid. The boot lid movement is achieved by using a H-bridge controlled motor. A hazard that can happen due to a potential malfunction of the control module is closing the gate unintendedly while a human being is picking up things from the boot.

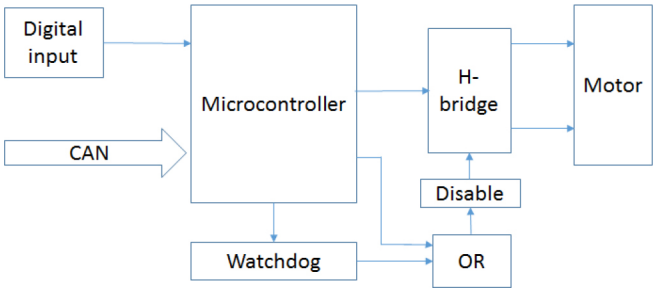

Figure 1. System Block diagram for the rear gate module

From the hazard above we can derive the safety goal "Prevent unintended closure of the rear gate" and is rated ASIL A. The standard provides five classifications with ASIL A being the least stringent and demanding whereas ASIL D on the contrary being the most stringent. QM classification would call for no special measures from the ISO26262 standard and the use of an appropriate industry specific quality standard would suffice. The ASIL level is identified based on highest ASIL level hazard identified during the Hazard Analysis and Risk Assessment during the concept phase.

We will now look at how we can use two popular methods for safety analyses to derive technical safety requirements for this safety goal.

Method: Fault Tree Analysis

Fault Tree Analysis is a deductive safety analysis method. The system is analyzed systematically top to down for faults that can contribute to a top event. This method of analysis is an effective tool to elicit all the possible lower level events (errors or faults and their inter relationship) that can contribute to a particular top level event (failure) in the system.

FTA's were developed in the early 1960's by Bell Labs and has since been extensively used for safety engineering in a number of industries including aerospace, automotive and nuclear to name just a few.

The fault tree is constructed using gates and events. The tree like graph is derived based on logically connecting top level failure events with lower level fault events using gates. Two common gates used in a fault tree are the AND gate and the OR gate. The AND gate is used to when all the connected lower level events need to occur for the upper level event. The OR gate is used when either lower level event can trigger the upper level event.

A *cut set* is a combination of gates and events that can cause the top level event. A *minimal cut set* is the smallest combination of events that result in the top level event.

The beauty of the fault tree is that helps a safety engineer identify all the potential single point faults and dual point faults that can cause the unsafe malfunction at the top level. Based on the ASIL of the item and probability of occurrence of the events represented by the cut sets a safety engineer can prioritize and determine where to focus in specifying safety requirements and safety mechanisms to strengthen the safety moat around the item.

Let's look at how we can create a FTA for the rear gate module and use it for deriving technical safety requirements for the safety goal - Prevent unintended closure of the gate. The violation of the safety goal is designated as the undesired top level event. In this case this would be - Unintended closure of the gate.

The next step is to brainstorm through the system architecture to identify the faults and failures that can cause the top level event. These errors would constitute the next lower level of events. It is highly recommended to do this exercise in a workshop atmosphere with the right stakeholders involved (experts from various disciplines system, software and hardware). An alternate approach could be to reference design documents generated by the stakeholders in case they are not available as it is very important to model the fault correctly based on sub-system design.

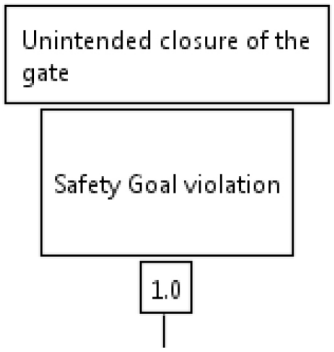

Figure 2. FTA top level event

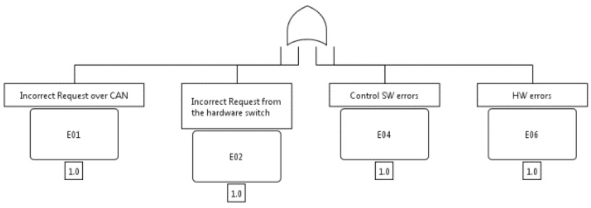

Figure 3. FTA with lower level errors

In the example under discussion we can elicit the following lower level errors -

1. Incorrect request over CAN
2. Incorrect request from the hardware switch
3. Control software errors and
4. Hardware errors

The next step is to elicit the actual lower level faults that can cause these errors. In complex systems there could be multiple levels of errors before we reach the lowest level fault.

At the lowest fault level the fault tree would look like the following. The figure below is only to indicate how big and complex the tree can get and not expected to be readable. Each cut set is dealt in detail below.

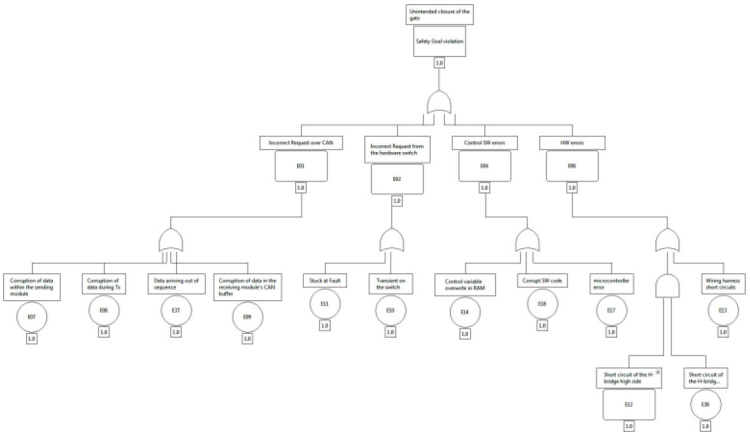

Figure 4. Complete fault tree

The lowest level enumerates all the different kinds of faults that can violate the safety goal.

The next step would be to separate each cut-set and then define technical safety requirements for each minimal cut set.

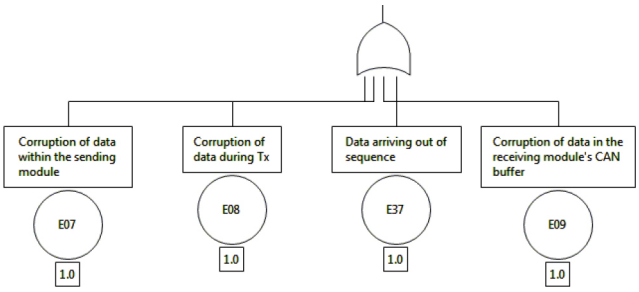

Figure 5. Cut-set for error - Incorrect request on CAN

Table 1. TSR's for cut set - Incorrect request on CAN

Fault	Technical safety requirement	TSR ID
Corruption of CAN data within the sending module	The sending module shall implement end to end diagnostics for the CAN message requesting gate closure	TSR1
Corruption of data during transmission	The sending module shall implement end to end diagnostics for the CAN message requesting gate closure	TSR1
Data arriving out of sequence	The sending module shall implement end to end diagnostics for the CAN message requesting gate closure	TSR1
Corruption of data in the receiving module's CAN buffer	The rear gate module shall evaluate the rolling counter and checksum for the CAN message before using the data for gate closure The rear gate module shall process the CAN message requesting gate closure once every 'x' ms	TSR2

The CAN message requesting the gate closure should implement end to end diagnostics which includes a rolling counter, checksum and timeout. This will ensure that rear gate module can detect even if anything is amiss with the received CAN message.

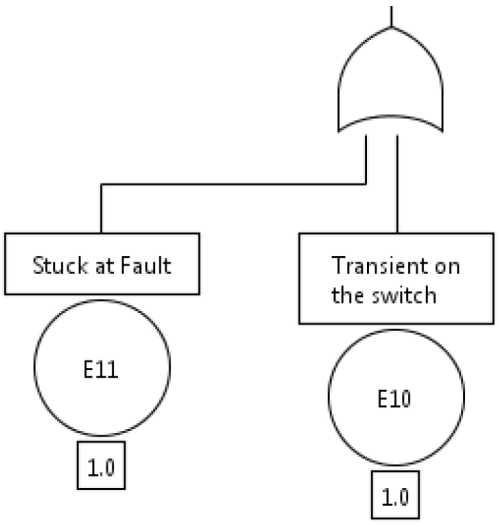

Figure 6. Cut-set for error - Incorrect request from the HW switch

Table 2. TSR's for cut set - Incorrect request from HW switch

Fault	Technical safety requirement	TSR ID
Stuck at fault	The rear gate module shall be able to detect short circuit errors (both Battery and Ground)	TSR3
Transient on the switch	The rear gate module shall debounce the digital switch input signal on digital input pin 5 for a minimum of 50ms before determining the status.	TSR4

A stuck at fault can be detected by checking for the switch reading ON for a non-plausible period of time. Transients and spikes at the input can be filtered with RC filter in the hardware and additional debouncing in the software.

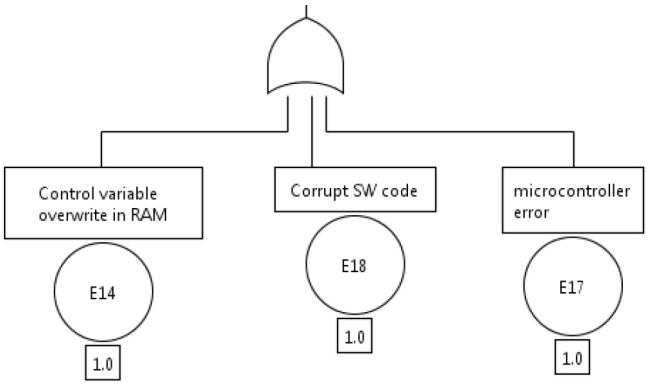

Figure 7. Cut set for error - Control SW errors

Table 3. TSRs - Control SW errors

Fault	Technical safety requirement	TSR ID
Control variable overwrite in RAM	The rear gate module shall implement a RAM test at startup	TSR5
	The control variable shall be double stored in RAM as a complement and be crosschecked each time before use.	TSR6
Corrupt software code	The rear gate module shall check for the checksum integrity for Flash memory sections at each startup.	TSR7
Microcontroller errors (ALU error or Clock error or	The rear gate module shall implement an ALU test which can test the integrity of a few core instructions	TSR8
	The rear gate module shall implement a windowed watchdog	TSR9
	The windowed watchdog shall be able to disable the H-bridge if it is not tickled within the time window	TSR10

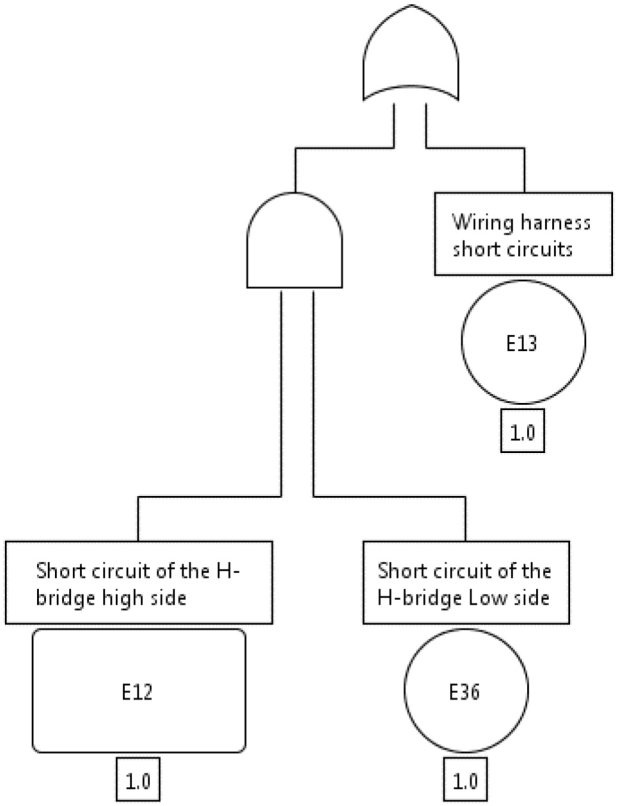

Figure 8. Cut-set for error - Hardware errors

Table 4. TSR's for error - Hardware error

Fault	Technical safety requirement	TSR ID
short circuits of H-bridge high side	The rear gate module shall be able to detect short circuit faults on the H-bridge high side using a current feedback mechanism. The rear gate module shall warn the driver in case of a confirmed short circuit of the H-bridge high side	TSR11 TSR12
short circuits of H-bridge low side	The rear gate module shall be able to detect short circuit faults on the H-bridge low side using a current feedback mechanism. The rear gate module shall warn the driver in case of a confirmed short circuit of the H-bridge low side	TSR13 TSR14
Wiring harness short circuits	The rear gate module shall be able to detect unintended activations of the motor using a current feedback mechanism The rear gate module shall be able to disable the H-bridge power stage in case an unintended activation of the motor is detected	TSR15 TSR16

The short circuit faults of the H-bridge - high side or low side are not single point faults which mean on their own they will not cause a violation of the safety goal. In ISO26262 parlance they are called dual point faults. However, if either fault remains latent or undetected and a second fault occurs on the H-bridge at a later point in time this will result in a direct violation of the safety goal. Therefore once a dual point fault is detected the driver should be warned so that corrective actions can be taken at a garage. In a fault tree dual point faults manifest as two or more faults connected by an AND gate in a cut set. The technical safety concept should strive to detect all such dual point faults identified in the fault tree.

The technical safety requirements should attempt to cover all the single point faults identified in the fault tree. The degree of coverage required depends on the ASIL level and the metric target for the item.

The technical safety requirements should identify faults that can render a safety mechanism ineffective. These faults though not capable of directly violating a safety goal; would result in a safety goal violation if the fault it was trying to detect would later occur. Per definition these faults would be dual point latent faults if undetected. The idea would be to make them detected by defining technical safety requirements to detect them as soon as they occur and warn the driver through a warning lamp or message. An example of such faults would be an ADC fault that would render the current feedback mechanism ineffective.

Derivation of technical safety requirements using a FTA can be summarized in the following steps:

1. Define the safety goal violation as the top level event
2. Bring in the system experts to a single room
3. Construct the fault tree identifying lower level failures and faults that can cause the top level event (safety goal violation)
4. Identify minimal cut-sets of the tree that contribute to an error that in turn violate the safety goal. Probability of occurrence of each event can be used for guidance on which faults or errors should be included. The probability thresholds could vary based on the ASIL level of item i.e. inclusion of even less likely events for ASIL D.
5. Define technical safety requirements for each minimal cut set
6. Cut-sets with multiple faults conjoined by an AND gate should be used to define technical safety requirements for dual point faults.
7. Identify technical safety requirements for dual point faults that can occur by virtue of faults that render safety mechanisms defined in 5 ineffective.

A Second Method: FMEA

A second approach to deriving technical safety requirements could be made using a FMEA. In contrast to a FTA, Failure Mode Effect and Analysis or FMEA is an inductive analysis approach analyzing the effect of a fault or failure on the system.

An example of a FMEA structure for the CAN request to demonstrate this method

Component/ Function	*Potential Failures*	*Potential Failure Effects*	*Severity*	*Potential Failure Causes*	*Occurrence*	*Detection*	*RPN*	*Recommended Actions / TSR's*
Sending of closure request on CAN	corruption of Request data	Potential to send incorrect request to Rear Gate	9	corruption of data within sending module	2	8	144	The sending module shall implement the CAN stack software to meet atleast ASIL A requirements
	corruption of Request data	Potential to send incorrect request to Rear Gate	9	corruption of data during transmission	2	8	144	The sending module shall implement end to end diagnostics for the CAN message requesting gate closure
	corruption of Request data	Potential to send incorrect request to Rear Gate	9	Corruption of data in the receiving module's CAN buffer	2	8	144	The rear gate module shall evaluate the rolling counter and checksum for the CAN message before using the the data for gate closure

The functionality of the sending module - “sending of closure request on CAN” is chosen as the function whose failure mode is to be analyzed. The next step is to list the potential failures and their failure causes. The severity is rated as 9 as its safety related. The recommended actions are basically the technical safety requirements and in this example are:

1. The sending module shall implement the CAN stack software to meet at least ASIL A requirements

2. The sending module shall implement end to end diagnostics for the CAN messages requesting gate closure.
3. The rear gate module shall evaluate the rolling counter and checksum for the CAN message before using the data for gate closure

An FMEA can be very effective in identifying common cause failures e.g. - what happens if the power supply fails. The key is to use either method to identify all the single point faults and potential dual point faults that can cause the hazard.

Irrespective of the method used the technical safety requirements should meet the following characteristics mandated by the standard:

1. Unambiguous and comprehensible (there is a common understanding of the meaning)
2. Atomic (cannot have more than one requirement at the same level bundled together)
3. Internally consistent (contains no contradictions)
4. Feasible (can be implemented within the constraints of item development)
5. Verifiable (can be tested for later in the safety lifecycle)

Each TSR should be verified against these five characteristics.

All safety requirement should have the following attributes

- A unique ID that makes it traceable throughout the safety lifecycle
- An ASIL rating and
- The current status - proposed, reviewed, assumed or accepted

Table 5. Attributes for technical safety requirements

TSR ID	Technical safety requirement	ASIL	Status
TSR1	The sending module shall implement the CAN stack software to meet atleast ASIL A requirements	A	PROPOSED
TSR2	The sending module shall implement end to end diagnostics for the CAN message requesting gate closure	A	PROPOSED
TSR3	The rear gate module shall evaluate the rolling counter and checksum for the CAN message before using the data for gate closure	A	PROPOSED

The standard also calls for the ensuring the following properties are met for managing safety requirements:

- Hierarchical Structure (requirements are arranged in several successive levels and are aligned with the corresponding design phases)
- Completeness (safety requirements at one level fully implement all safety requirements of the previous level)
- External consistency (multiple safety requirements do not contradict each other)

  Maintainability (requirements can be modified or extended during the safety lifecycle)

These properties can be easily adhered to by using industry standard requirement engineering tools (e.g. - DOORS, Polarion etc)

Fault Tolerant Time Interval

Safety mechanisms ensure that the item can transition into and maintain a safe state in case an error that can violate the safety goal is diagnosed. An important aspect of this is being able to do this within the fault tolerant time interval or FTTI allocated to the item. According to Part 1.45 of the ISO26262 standard fault tolerant time interval or FTTI is the “time-span in which a fault or faults can be present in a system before a hazardous event occurs “.

The actual time available for the ECU itself could be much less than the vehicle level FTTI associated with the safety goal. While developing the technical safety requirements including safety mechanisms for single point faults it is key that the safety engineer makes an analysis and evaluation on the timing constraints versus capabilities of the existing hardware and software.

Some aspects that need to be included in this analysis includes

- Time required for the input signal to reach the ECU pin (CAN signals, sensor inputs) ,
- Hardware processing delays (RC filters, signal conditioning) ,
- Software processing delays (scheduling, data propagation delays between modules etc)
- Time required for output actuation (power stage and actuation, CAN signal propagation etc)

One or more of the following methods are suggested for deriving the FTTI at ECU level

- Timing diagram based analysis
- Expert judgment
- Vehicle tests in controlled conditions
- Simulation and HiL tests

Conclusion

The ISO 26262 standard Part 4 clause 6 describes the need for technical safety requirements in the product safety lifecycle. Writing good technical safety requirements is a prerequisite for ensuring sufficient level of product safety is achieved. The standard mentions the properties and characteristics that technical safety requirements need to achieve. However the standard does not throw light on any methods for deriving technical safety requirements. The paper discusses how popular and well understood safety analyses techniques like a FTA can be used identify single point faults and multiple point faults that can cause a safety relevant malfunction and result in the hazard. The minimal cut sets identified can be used to derive the bulk of technical safety requirements.

A distinct advantage of using the FTA method to build the safety requirements is that it provides a safety engineer the necessary confidence on the coverage of faults in the safety concept. Moreover this can be used during joint reviews with the OEM to demonstrate the coverage achieved by the technical safety concept.

These methods are meant to be help relatively less experienced safety engineers elicit technical safety requirements. There are additional aspects like requirements for production and operation which need to be considered for completing the safety requirements.

On a final note writing good technical safety requirements is an ever evolving process and organizations should capture lessons learnt and best practices and aim at continuous improvement.

References

1. International Standards. (2011). ISO 26262 Functional safety for road vehicles. Geneva, Switzerland.
2. Fault Tree Handbook by Vesely William Dr
3. Developing Functional Safety Requirements using Process Model Variables by Krithivasan Gokul
4. Implementation and Verification of Technical Safety Requirements for a Dynamic Torque Vectoring Feature in an Electronic Brake System by Worden James, Schneider Michael, Traskov Adrian Dr.
5. Functional Safety Analysis at the Software Architecture Level by Barnes Doug
6. Virtualized Fault Injection Methods in the Context of the ISO 26262 Standard by Reyes Victor
7. Writing Software Requirements Specifications (SRS) Written by Vie Donn Le, Jr.
8. Quantified Fault Tree Techniques for Calculating Hardware Fault Metrics According to ISO 26262 by Das Nabarun

Contact Information

Agish George, is a Senior Functional Safety Engineer at kVA and a Functional Safety Certified Automotive Engineer, and can be reached at +1-248-479-8969 or
agish.george@kvausa.com

Abbreviations & Definitions

ASIL - Automotive Safety Integrity Level, a term defined in ISO 26262 [2] and used to define the requirements for safety integrity in automotive E/E systems.

CAN - Controller Area Network

DIA - Development Interface Agreement

HARA - Hazard Analysis and Risk Assessment

FTA - Fault Tree Analysis

ECU - Electronic Control Unit.

FMEA - Failure Mode Effects Analysis

OS - Operating System

QM - Quality Management, a term defined in ISO 26262 [2] and used to define the requirements for safety integrity in automotive E/E systems.

HW - Hardware

FSC - Functional Safety Requirement - Technical Safety Requirements

TSC - Technical Safety Concept

FTTI - Fault tolerant time interval

APPENDIX

APPENDIX A: FIGURES

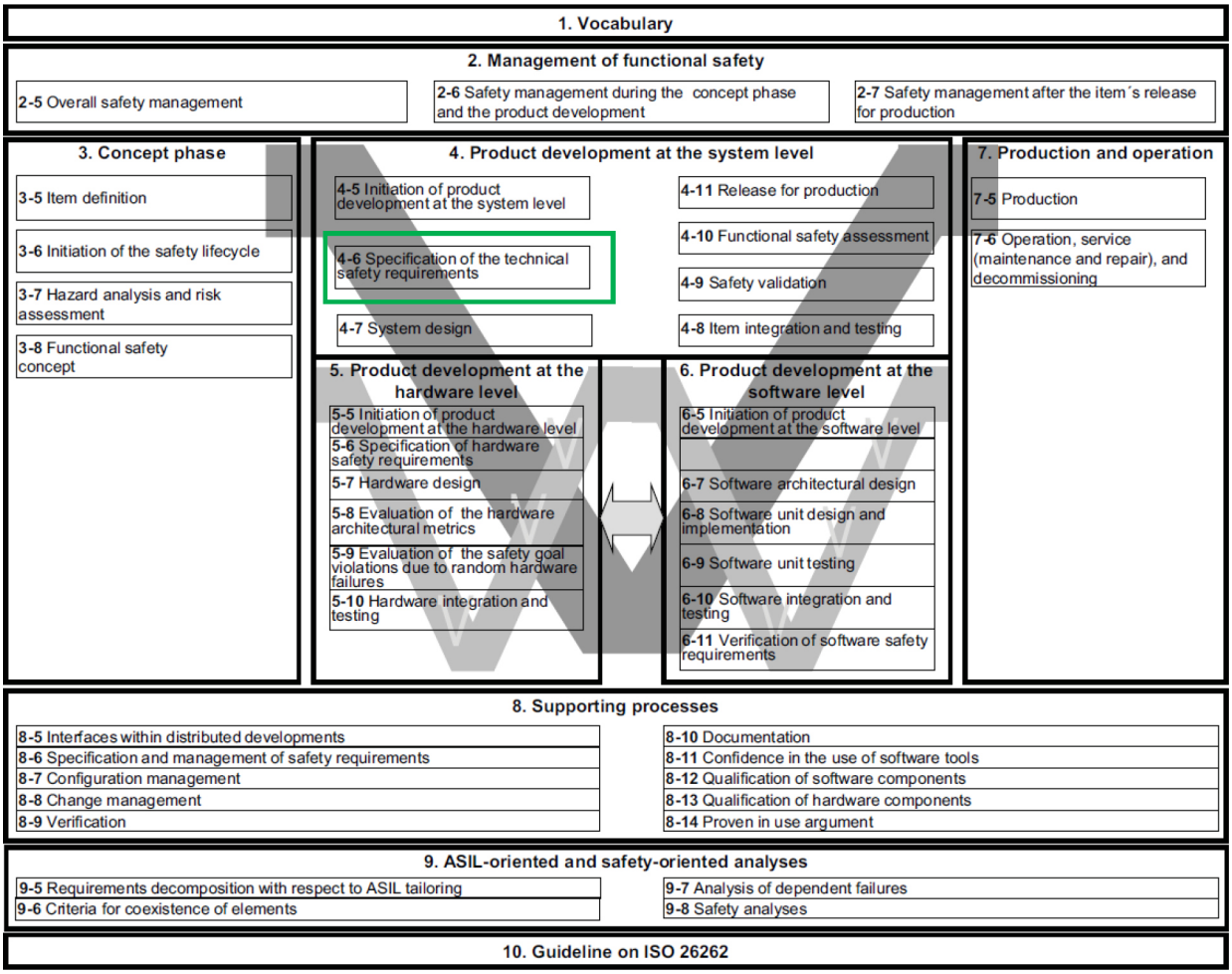

Figure A1. ISO26262 overview

Clause	Objectives	Prerequisites	Work products
6 Specification of the technical safety requirements	The first objective of this subphase is to specify the technical safety requirements. The technical safety requirements specification refines the functional safety concept, considering both the functional concept and the preliminary architectural assumptions (see ISO 26262-3). The second objective is to verify through analysis that the technical safety requirements comply with the functional safety requirements.	Functional safety concept (see ISO 26262-3:2011, 8.5.1) Validation plan (see 5.5.4)	6.5.1 Technical safety requirements specification 6.5.2 System verification report 6.5.3 Validation plan (refined)

Figure A2. Document flow

The Engineering Meetings Board has approved this paper for publication. It has successfully completed SAE's peer review process under the supervision of the session organizer. The process requires a minimum of three (3) reviews by industry experts.

ISSN 0148-7191

http://papers.sae.org/2016-01-0127