Making Software Safety Assessable and Transparent

Risto Nevalainen[1], Alejandra Ruiz[2], and Timo Varkoi[3]

[1] Spinet Oy, Finland
[2] Tecnalia, Spain
[3] Finnish Software Measurement Association – FiSMA ry, Finland
risto.nevalainen@spinet.fi, alejandra.ruiz@tecnalia.com,
timo.varkoi@fisma.fi

Abstract. Most formal assessment and evaluation techniques and standards assume that software can be analysed like any physical item. In safety-critical systems, software is an important component providing functionality. Often it is also the most difficult component to assess. Balanced use of process assessment and product evaluation methods is needed, because lack of transparency in software must be compensated with a more formal development process. Safety case is an effective approach to demonstrate safety, and then both process and product are necessary evidence types. Safety is also a likely candidate to be approached as a process quality characteristic. Here we present a tentative set of process quality attributes that support achievement of safety requirements of a software product.

Keywords: software process, process assessment, software safety.

1 Introduction

Critical systems are defined as those that in case of an incident or misbehaviour can lead to an accident that will put people or the environment in danger, resulting in injuries and or casualties. Safety is considered as a general property of the whole system and so its plans, developments and implementations must follow strict rules in order to prevent failures of the system and their consequences and risks.

Software-based systems are increasingly important in safety. They replace old wired and analog systems, and they also bring new technologies in safety. They are more standardized and functionality-rich than earlier generations. We can even say that they are more reliable. At least we can use diversity and redundancy more effectively, because digital systems are typically cheaper than old analog systems.

But software brings also problems. Behaviour of software is rather deterministic (i.e. exactly predictable) than probabilistic (i.e. likely to happen). We have to compensate these deficiencies somehow, for example by formal and visible process and by extensive documentation. Still some uncertainty remains and the ultimate “zero defect” or “high reliability” goal is very difficult to achieve.

To some extent we can even challenge the current definitions of safety. For instance, Leveson [1] states that: *“Highly reliable software is not necessarily safe. Increasing software reliability will have only minimal impact on safety.”* With control

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 1–12, 2013.

systems reliability and safety are often confused and the same principles that have worked with hardware are applied to software. In addition, there is no explicit relationship between process quality and product quality. However, the development processes affect the quality of software, including its safety. The main concern should be in management and development of requirements [2].

This paper presents three approaches for assessments of software safety: alignment of process assessment and product evaluation methods with a new concept of property; presenting a new process quality characteristic for safety; and application of safety cases to support safety assessment. The approaches are developed based on our experiences in various safety-critical domains. The approaches are continually evaluated partly in on-going research projects but also in real assessments.

2 Process and Product Perspectives in Assessing Software Safety

2.1 Different Approaches in Safety Assessment and Evaluation

We can evaluate safety-critical software from several viewpoints. The key output, the software product, can be evaluated against a predefined set of e.g. quality requirements. Safety assessment can study both the product and the processes used in development and use of the software, based often on domain specific standards. Process assessment focuses typically on the product development phase. All these approaches (Table 1) produce valuable information in building trust on the safety of the product. So far, harmonization of these approaches is missing for safety-critical software.

Table 1. Comparison of main approaches in software safety evaluation

Topic	Product evaluation approach	Safety assessment approach	Process assessment approach
Main purpose of the approach	To analyse and show compliance of product (artefact) by using selected criteria	To demonstrate compliance with a selected reference (standard)	To demonstrate capability to develop, deliver and improve
Main focus in safety-critical domain	Product quality, especially reliability metric and data, for example MTBF	Compliance with generic or domain specific safety standard, certification	Process evidences to demonstrate achievement of safety management and engineering
Specifics of each approach	Internal, external, in use metric	Inspections, reviews, V&V evidences, technical practices and methods	Professional practices, work products, capability levels
Commonalities with other approaches	V&V metric, measurement and analysis practices	Engineering methods and competences	Process results, like mandatory work products
Typical standard(s) and models	ISO/IEC 25000 family (SQUARE)	IEC 61508, ISO 26262 IEC 60880, IEC 62304	ISO/IEC 15504 (SPICE), Automotive SPICE, Nuclear SPICE

Product evidences can also serve as safety assessment and process assessment evidences and vice versa. So, it is meaningful to harmonise those approaches to support each other. An example could be traceability, which is a direct requirement in safety assessment standards and in process assessment models. Another good example is testing coverage, which can be classified as both product and process evidence for verification and validation (V&V) activities.

Product quality model ISO/IEC 25010 [3] (known also as SQUARE framework) includes eight characteristics in internal and external metric and five characteristics in in-use metric. Reliability is one characteristic, including Maturity, Availability, Fault tolerance and Recoverability as sub-characteristics. Safety is less obvious sub-characteristic in in-use model, belonging to Freedom from risk characteristic. It is called there “Health and safety risk mitigation”. Safety is then most relevant in existing operational systems. This view is limited, because safety can and should be built in the system and software by a rigorous development process. Safety can be seen as a combination of process quality and product quality. Main metric for safety in ISO/IEC 25024 draft standard is operational experience, expressed as number of failures / cumulative operation time per given period (typically hour).

Safety assessment is a widely used method in certification. It contains typically different analyses to calculate system reliability and risks for failure, as required in the selected reference standard. The most common standard is IEC 61508, Functional safety [4]. Quite common is to make safety assessment against some Safety Integrity Level (SIL). This approach is based on probabilistic behaviour of the system. It works well for the whole system and for hardware components.

Software is more problematic, because its behaviour is rather deterministic than probabilistic. Detailed checklists are typically used to cover requirements in the selected reference standard. The result of safety assessment can be a statement of conformity or certificate. This result can be very valuable if the reference standard is reasonably up to date and the system under evaluation does not include too experimental technologies. The open issue is trust on the conformity in reference standard, and is that any guarantee of software safety.

Process quality can be covered by the SPICE process assessment. It is based on an international standard, currently ISO/IEC 15504 [5] (in the future ISO/IEC 330xx series [6]). ISO/IEC 15504 origins are in generic software development. Some domains have developed their own variants of the framework, such as Automotive SPICE and SPICE for SPACE. The latest domain specific Process Assessment Model (PAM), developed based on ISO/IEC 15504 principles, is Nuclear SPICE that is intended to address the highest safety requirements. This work is a part of a large Finnish nuclear safety SAFIR 2014 research program [7].

One important topic in process quality and assessment is the extent of validation and verification (V&V) in the software lifecycle. The safety lifecycle contains normally a quality assurance process, for example independent process review or audit. Additionally, independent technical reviews, independent tests and acceptance phases can exist. Process assessment in safety context is normally a mix of basic SPICE type approach and use of selected safety standard(s).

These basic approaches have also significant overlaps. Safety standards have direct requirements for product or system, even they are mostly process centric. Good conformity with standards has always evidences from both product and process.

Also process assessment has both process and product view. Work products are direct evidences of process in SPICE model, at the same logical level as process specific practices. Generic work products are evidences for higher capability and also for safety in such context.

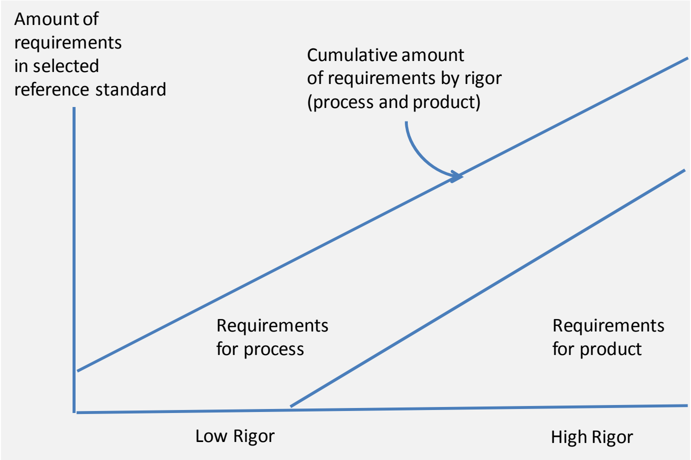

Fig. 1. Requirements for process and product at different levels of rigor in safety standards

2.2 Integration of Product and Process Quality

SQUARE framework defines product quality at three levels: the overall quality, selected product quality characteristics and further selected product quality attributes. In current version of SQUARE there are 13 product quality characteristics. The work is still on-going in defining product quality attributes and their candidate metrics. The origin of SQUARE is in software, so we can interpret the concept of product quality as "software quality".

We have proposed for Process Assessment standardization community to use similar hierarchy for process quality. Currently ISO/IEC JTC1/SC 7 Working Group 10 is progressing to upgrade 15504 set of standards and to develop a new set of assessment requirements as ISO/IEC 33000 set of standards. This development introduces a new concept of process quality. Process quality concept harmonizes the terminology with product quality (Figure 2).

We can see immediately useful combinations, for example to study which processes contribute in some selected product quality attribute.

Going further, we can add concept of property both in product and process quality. Property can be seen as a real life instance of some attribute. Obviously, real life does not classify so beautifully as our quality concepts. So, any property can be either product property, process property or both. The usefulness of property concept is in its details. It can be observed and even measured objectively.

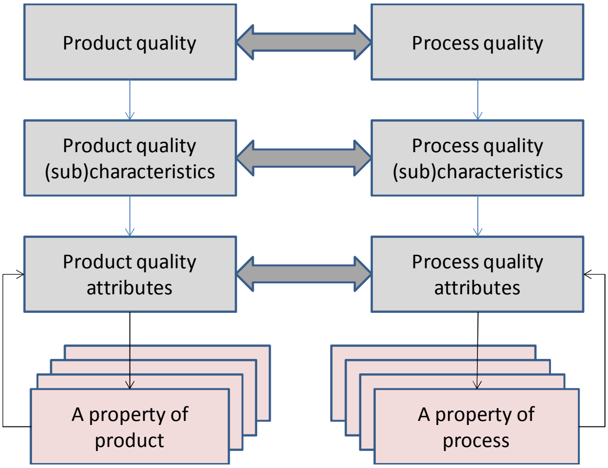
Product quality
Process quality
Product quality (sub)characteristics
Process quality (sub)characteristics
Product quality attributes
Process quality attributes
A property of product
A property of process

Fig. 2. Alignment of product and process quality concepts

2.3 Use of Methods and Properties in Nuclear SPICE

In this chapter we use Nuclear SPICE [7] as the reference model to explain how methods and properties can be used as evidence for product and process. Nuclear standards IEC 61513 and further IEC 60880 are based in IEC 61508 series and may include all SIL levels in their requirements. Nuclear standards have a different concept of classes (1, 2, 3) and categories (A, B, C). Nuclear SPICE is designed to satisfy all classes and all categories. So, it shall cover also all techniques and methods included in IEC 61508. Of course, engineering judgment is needed in using methods because otherwise model would be too heavy for practical use.

IEC 61508-7:2010 Annex C [8] lists the topics in which the concept of safety property is proposed. In most cases, it is fairly easy to find corresponding processes in Nuclear SPICE as the list above, as seen in Table 2.

Table 2. Mapping topics of safety properties with Nuclear SPICE processes

Topic in IEC 61508-7 Annex C	Corresponding process in Nuclear SPICE	Comments
Software Safety Requirements Specification	DEV.1 Software requirements analysis	
Software architecture design	DEV.2 Software architectural design	
Support tools and programming language	-	This process is mainly missing in SPICE
Software detailed design	DEV.3 Software detailed design	
Software module testing and integration	DEV.4 Software construction DEV.5 Software integration	
Programmable electronics integration (hardware and software)	ENG.5 System integration	ENG.5 needs interpretation but has good match
Software aspects of system safety validation	ENG.5 System integration ENG.6 Systems qualification testing SAF.2 Safety Engineering	Scattered coverage in SPICE processes.

Table 2. (*continued*)

Software modification	SUP.8 Software problem resolution SUP.9 Software change request management	
Software verification	SUP.4 Verification	
Functional safety assessment	SAF.1 Safety Management SAF.3 Safety Qualification	Scattered coverage in SPICE processes.

We can also see that in some cases mapping is not straightforward. For example in topic “Support tools and programming language” no one Nuclear SPICE process covers it, at least not in this level of details. It needs interpretation and also further development of Nuclear SPICE.

One important finding is also that the set of properties in IEC 61508 does not cover all relevant topics in nuclear safety. Nuclear SPICE has an extensive set of processes for system level specifications and design, and properties are missing there. Also quality assurance (SUP.3 in Nuclear SPICE) has no properties. The conclusion is that also other kinds of evidences are needed than what we have specified so far. The software SPICE (ISO/IEC 15504-5:2012) can be used as such, but it needs more engineering judgment and interpretation in some topics, than what is maybe acceptable.

3 Safety from Process Quality Viewpoint

3.1 Process Quality Characteristic

In the current development of process assessment standards, new concepts are adapted that enable new approaches to address process quality. Safety can be considered as one example of process quality characteristics, which could be used in assessing process quality when developing software for safety-critical domains.

The adopted principles include that process quality is composed of quality characteristics, where the required set of characteristics depends on the applicable stakeholder needs and organization’s business goals. In addition, process quality shall be measurable. The key terms are defined as follows [6]:

Process quality

- ability of a process to satisfy stated and implied stakeholder needs when used in a specified context

Process quality characteristic

- a measurable aspect of process quality; category of process attributes that are significant to process quality

By nature, process assessments are based on sampling and thus do not provide proper data for any probabilistic assessments. Therefore, it is important to understand that process assessment cannot be used to qualify a software or system product. However, process assessments can point out the risks related to achieve the required level of product or system quality, including safety.

3.2 Safety Process Quality Attributes

We have specified a preliminary model to address safety by process assessment [2]. At this stage, the model is tentative and intended to serve as a starting point for discussion on how processes influence in implementing software safety requirements. The model consists of two sets of process quality attributes (PA) for process assessment in safety domain. The basic set is intended to include attributes that meet the elementary requirements for trustworthy software development. The extended set adds process attributes that support management of processes that support safety activities. The attributes should be applied typically on development and quality assurance processes. Management processes are often too generic for this purpose. Applicable processes are described in the Nuclear SPICE PAM.

Table 3. Basic set of process quality attributes for safety

Process Attribute	Description
PA 1	Process performance - process achieves its defined process outcomes
PA 2	Dependability - reliability; process performs as required in normal conditions - availability; process can be performed when needed - maintainability; process can be modified easily to add capabilities; performance can be improved; faults and errors can be corrected
PA 3	Requirements control - traceability - specifications coverage - constraints - safety analysis - reuse
PA 4	Safety engineering - safety demonstration - reviews - verification and validation - quality assurance

Table 4. Extended set of process quality attributes for safety

Process Attribute	Description
PA 5	Safety management - safety strategy alignment - safety life cycle; defined activities involved in the implementation of safety-related systems - responsibilities and resourcing - monitoring - test and simulation environments
PA 6	Compliance - standards - defined process - process tailoring

Table 4. (*continued*)

PA 7	Quantitative management - quantitative analysis; measurement objectives; measures - quantitative control; techniques; causes of variation
PA 8	Risk management - management of events that effect achievement of business goals - qualitative and quantitative risk analysis for a process; probabilistic risk analysis
PA 9	Information security - preservation of confidentiality, integrity and accessibility of information during the execution of a process

The presented process safety attributes and sets shall be further elaborated and extended in descriptions. To enable efficient assessments, the process attributes will be completed with appropriate assessment evidence classes, including Generic Practices, Generic Resources and Generic Work Products.

4 Safety Cases as a Support Tool for Safety Assessment

Safety case is a requirement in many safety standards. Safety cases were originally inspired in the nuclear industry and have been used for more than 50 years. They have successfully been used also in other industries like chemical, military systems, the off-shore oil industry, rail transport, and recently in the aviation and automotive industries. According to the definition from ISO 26262, it is defined as an “argument that the safety requirements for an item are complete and satisfied by evidence complied from work products of the safety activities during development” [9]. Safety argumentation provides a valuable tool for the development of critical systems. A safety case is based on a goal representing an assertion that can be assessed as true or false. To reach to this target goal, we should construct an argumentation based on several items as described by J.R. Inge [10]:

- The scope of the system or activity being addressed, together with details of its context or environment.
- The management system used to ensure safety goal.
- The requirements, legislation, standards and policies applicable, with evidence that they have been met or complied with.
- Evidence that risks have been identified and appropriately controlled, and that the residual level of risk is acceptable.
- Independent assurance that the argument and evidence presented is sufficient for the application in question.

But sometimes, safety cases are seen with criticism due to several reasons according to Johnson and Robins [11]. One of the main complaints against safety cases is that it is always possible to find or produce evidence that something is safe. It is the confidence level that is put into that evidence what gives strength to the argumentation. Unfortunately, for safety analysis there is no complete mathematical theory to base arguments and guarantee completeness. It is also important to highlight that argumentation reasoning is sometimes made under non-explicit assumptions. Rasche enumerates the following general problems related to safety cases [12]:

- The amount of work required to construct a safety case including the specialized and costly (outside) resources required
- Problems associated with obtaining and validating data to justify a probabilistic risk analysis.
- Too much focus on technical risk and not enough on meeting the needs of workers

Unfortunately there is not a method from preventing in given inappropriate argumentation. The ideal scenario for creating strong, complete safety cases is to provide an independent, non-subjective argumentation. This could be reached by demonstrating that major hazards of installation and the risks to personnel therein have been identified and appropriate controls provided.

4.1 Safety Cases Argumentation

Standards tend to be prescriptive regarding specific solutions and process-oriented rather than product oriented. Safety Cases could be the bridge between these two approaches and balance the process-oriented with the product-oriented approaches. A strategy can be used to describe generic approaches to the arguments that are used in support of a goal or claim, such as reference to appropriate standard sections. There has been some research on this line that indicate that a solution for this could be the use of goal-based safety case such as described by Stensrud et al. at WOSORCER workshop 2011 [13] where they propose a hybrid approach to transform prescriptive elements in the standard IEC 61508 from a table format into a safety case format, creating then safety case patterns that map with the prescriptive elements from the standard.

Weaver et al. [14] presented a safety case framework that includes the top level software safety argument where the top level goal is that the system is acceptably safe. The top level goal is further broken down into sub-goals including that the safety requirements are valid. On the decomposition of these sub-goals we are able to link with the ideas previously commented from Stensrud et al. [13] to map with the IEC 61508 requirements and go deeper into the product characteristics.

In general, within the safety assurance research community as described by Flood and Habli, safety cases are increasingly viewed as consisting of three types of arguments [15]:

- risk (or “primary”) arguments – that aim to establish that the system is acceptably safe to be deployed.
- confidence (or “backing”) arguments – that are used to justify that sufficient confidence can be placed in evidence and inferences of the risk arguments
- compliance arguments – that show that requirements of the applicable standards have been satisfied

Special attention should be put into the confidence arguments, which are the key to make strong and credible argumentation. In ISO 26262 standards one of the requirements is to demonstrate a safety culture within the company. Well defined safety engineering processes are important as they offer confidence on the argument, and also the evidence that those processes are being followed demonstrate that the best practices identified by the company are put into practice.

4.2 Evolutionary Safety Cases

For a long time, it has been usual to leave the development of the safety case to the end of the project; however this approach can lead into a costly strategy as changes to the design at that time are very expensive. With the same view of early validation and verification, safety cases as internal audits can help in reducing the possible risks.

The creation of evolutionary safety cases along the project as a way to both mitigate possible risks, follow up design decisions with impact on safety, and at the same time as a powerful tool to support management from the safety point of view.

ISO 26262 encourages the idea of incremental safety cases. It recommends that the safety case should be developed along with the system. The standard proposes to have refined safety cases in which with each phase, the information is completed and the strategy for the next phase is defined.

We can define three different stages of the safety case:

- Preliminary safety case: At this stage, we will include information regarding: system scope; top safety requirements; main hazards; possible strategies; development approach; type of evidences needed. In this stage it should be assured that all hazards are covered and the mitigation strategies are possible to be put into practice (within budget, time etc.).
- Interim safety case: At this stage, we are able to increase the confidence on the design in comparison with argumentation from the preliminary safety case. In order to strength the argumentation, evidence for the preliminary validation is important to address the independence of the validation results, giving more confidence.
- Final safety case: includes complete arguments, from all types of argumentation described in previous section. The evidences such as: observation, measurements, testing and analysis of the implemented system support all possible arguments.

It is important to highlight that safety cases can be modified or changed throughout the operational life of the system, as additional safety evidence becomes available or new risk appears.

4.3 Safety Cases as a Support Tool for Safety Assessment

It is not rare, while doing safety assessment, to be presented long reports referencing to evidences, but those reports have lack in clarity on how those evidences relate to the safety requirements and how it is understood to comply with the standard.

Avionics standards do allow an applicant to propose “alternative methods of compliance" for some objectives, provided it can be shown how their new methods satisfy the “intention" of the objectives. The difficulty is that the intent of most objectives is not formulated explicitly. Thus, a reasonable enhancement to guidelines such as

DO-178B would be to include documentation of the intent of each objective. We could go further than this, and to supply a full argument that the evidence required by the standard does ensure satisfaction of explicitly stated safety goals. Such argumentation would be generic at standard level, but it could be also applied at the level of safety demonstration for a particular certification project.

ISO 26262 proposes to tailor some activities in order to propose forms to comply with the standard and at the same time that are adequate for the project. This involves interpretation of the standard and needs to be understood and agreed by both the company and the person in charge of the assessment.

The idea behind a safety case described by Tim Kelly [16] is that the application of an argumentation approach to the concept of target compatibility would require definitions, assumptions, and limitations to be made visible. This allows a much clearer evaluation for the contribution and limit to the overall correctness of the software and therefore its contribution to safety of the system.

On the SPICE assessment different indicators are defined. The indicators can be seen as the different goals to achieve. The base practices are the strategies which can be followed in order to comply with the objectives and the output work products can be seen as evidences for those strategies to been followed. The association with the SPICE assessment is easily mapped into the compliance argument type that was described before. The capability dimension of the SPICE model and how this capability is improved offers the confidence argumentation. However in the SPICE model the explicit risk arguments are missing or re implicit. Those arguments are linked mainly with the product properties (Fig. 2). The implementation of base practices can differ from a company to another. The negotiation between the company and the assessor can be more efficient and fruitful when the rationale (argumentation) is well understood and shared by all parties, and safety cases can be very helpful in this. Safety cases are a powerful tool to express the argumentation behind the compliance of the different requirements from the standards and at the same time, are able to express in a comprehensive and clear way many design decisions in relation with safety requirements.

5 Conclusions

This paper proposes three different approaches to improve assessability of software safety by presenting. First, integrated approach on product and process quality balances the use of process assessment and product evaluation methods. A new concept of property is added both in product and process quality. Second, safety is considered as a process quality characteristic. This enables assessment of software development processes using a specific set of process safety attributes. Third, safety cases can be used to support safety assessment and demonstration. Safety cases provide the argumentation for meeting the safety requirements of systems. Use of these approaches needs to be considered case by case. The overall critically of the application is the main driver in selecting an appropriate scope and combination of methods for safety assessment. The aim is to improve trust on software safety and to minimize the risks.

Acknowledgements. This work has been partially funded by the National Authorities involved in RECOMP and the Advanced Research & Technology for Embedded Intelligence and Systems (ARTEMIS) within the project RECOMP under Grant agreement no. 100202. Any opinions, findings and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of funding agencies. This work has been partially funded by Finnish national nuclear safety program SAFIR2014. In project CORSICA, new approaches and V&V methods have been developed for software safety. A method called Nuclear SPICE implements many of the ideas in this article. The research leading to these results has received funding from the FP7 programme under grant agreement n° 289011 (OPENCOSS).

References

1. Leveson, N.G.: Engineering A Safer World: Systems Thinking Applied to Safety. MIT (2011)
2. Varkoi, T.: Safety as a Process Quality Characteristic. In: Proceedings of SPICE 2013 Conference (accepted for publication, 2013)
3. ISO/IEC 25010:2011, Systems and software engineering–Systems and software Quality Requirements and Evaluation (SQuaRE)–System and software quality models (2011)
4. IEC 61508-3 Ed. 2.0, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements (2009)
5. ISO/IEC 15504-5:2006, Information technology – Process assessment – Part 5: An exemplar Process Assessment Model (2006)
6. ISO/IEC 33001 DIS, Information technology – Process assessment – Concepts and terminology. ISO/IEC (2013)
7. FiSMA 2011-1: S4N Method Description - Nuclear SPICE PRM and PAM. FiSMA (2012)
8. IEC 61508-7 Ed. 2.0, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 7: Overview of techniques and measures (2009)
9. ISO 26262, Road vehicles – Functional safety, ISO (2011)
10. Inge, J.R.: The Safety Case: Its development and use in the United Kingdom. In: Equipment Safety Assurance Symposium, Bristol, UK (2007)
11. Johnson, C.W., Robins, D.A.: Myths and barriers to the introduction of safety cases in space-based systems. In: 29th International Systems Safety Society, Las Vegas, USA (2011)
12. Rasche, T.: Development of a safety case methodology for the Minerals Industry – a discussion paper. Minerals Industry Safety and Health Center (2001)
13. Stensrud, E., Skramstad, T., Li, J., Xie, J.: Towards Goal-Based Software Safety Certification Based on Prescriptive Standards. In: First International Workshop on Software Certification, WoSoCER (2011)
14. Weaver, R.A., McDermid, J.A., Kelly, T.P.: Software Safety Arguments: Towards a Systematic Categorisation of Evidence. In: Proceedings of the 20th International System Safety Conference (ISSC), System Safety Society, Denver (2002)
15. Flood, M., Habli, I.: Multi-Viewpoint Safety Cases. In: Proceedings of the 6th IET International System Safety Conference, Birmingham, United Kingdom (2011)
16. Kelly, T.: Arguing Safety - A Systematic Approach to Managing Safety Cases. PhD thesis, Department of Computer Science, The University of York (1998)