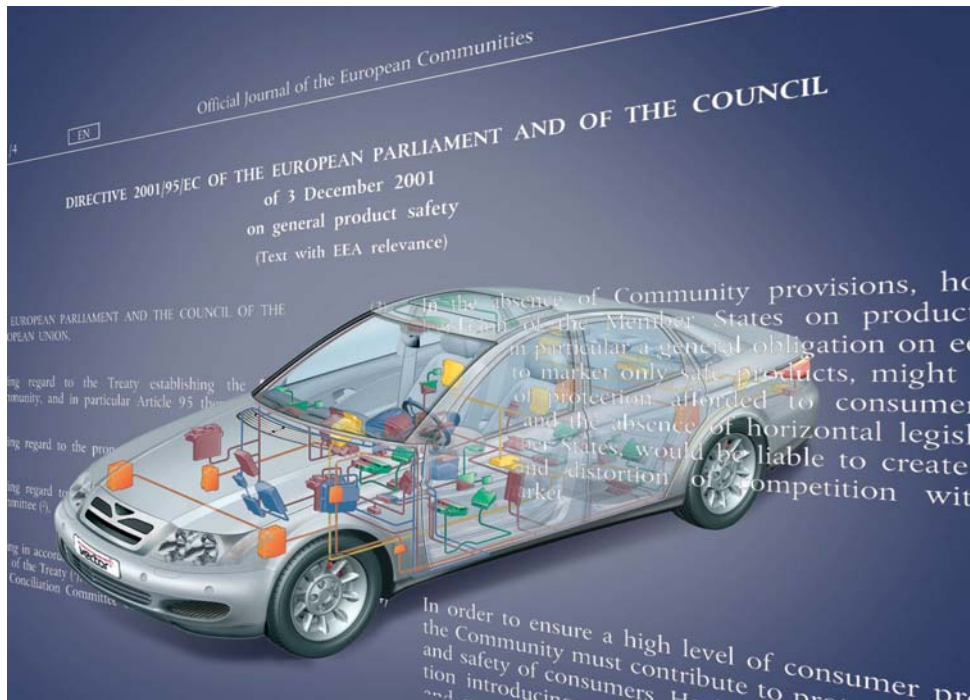


Safety-Relevant Development Processes – Should Everything be Restructured?



(Title image)

With the intensified entry of software, even into safety-critical areas such as steering and braking [1] product safety is becoming a hot topic. Besides leading to costly recall actions that also take a toll on a producer's image, substantial financial risks are also incurred, above all in product liability and producer liability. How can product safety be assured, and what consequences does this have for the development process? Does everything need to be restructured – or is it possible to build upon tried and true methods? In this article Dr. Klaus-Jürgen Amsler, Senior Consultant, Dr. Joachim Fetzler, Director of Vector Consulting GmbH and Dr. Meinhard Erben, Legal Counsel for IT Law, explain what is meant by a safety-relevant development process.

The implementation of a safety-relevant development process is described in light of legal and technical constraints. This approach begins with an existing process environment and advances step-by-step to a development process that meets safety needs and is efficient.

Legal Constraints

The TREAD Act [2] in the USA requires companies and employees in positions of responsibility to report - comprehensively and proactively - all actions taken in response to defective parts. Failure to do so may result in severe penalties. The EU has incorporated similar product monitoring duties into the existing General Product Safety Directive (2001/95/EC) which was adopted into German law in June 2004.

A safety-relevant development process is characterized by the production of products that satisfy the demands of product safety, i.e. they are defect-free on every system level, e.g. vehicle, control module and software. Defect-free in a legal sense means to be substantially without design or production defects. Software defects are generally design defects, since they always occur under identical conditions, while a faulty production process can lead to manufacturing defects where there is inadequate quality assurance.

This liability doctrine protects not only the individual user, but also every third party, whereby differentiation is made between product liability and producer liability:

- Product liability refers to the results of the production process, i.e. the finished product.

- Producer liability applies when the producer neglects its organizational and diligence duties during development and production.

The following remarks focus on development and production processes and thereby, from a legal perspective, on (potential) producer liability. They address the question: What duties apply to the producer in development and production processes?

Organizational and Diligence Duties of the Producer

The producer must set up its business so that design and production defects - even those in supplied parts - can be excluded to the greatest extent possible or be detected by controls. Since it is (nearly) impossible to develop software-based systems without defects, the producer is obliged to observe all organizational and diligence duties incumbent upon it in yet more meticulous detail, i.e. the software-based system must - to the greatest extent possible - be tested so intensively as to exclude serious hazard to life or limb.

This applies especially to the development of embedded systems for motor vehicle applications, since complex mechatronic systems (mechanics, hardware and software) are involved. Development environments with consistent design, implementation, integration, testing and simulation concepts must be implemented in that area.

For the software producer, as well as the motor vehicle producer, it is eminently important - also for evidentiary reasons - to precisely observe standards on the state of the art in technology, e.g. IEC61508 [3] or Process Matur-

ity Models such as CMMI [4] and SPICE [5]. The reason for this is that questions of evidence play a central role in product and producer liability. In contrast to the burden-of-proof rules that apply in other legal situations, in the case of producer liability the proof burden is reduced for the injured party, or there may even be a shifting of the burden of proof; essentially the injured party only needs to show that the product exhibited an objective safety deficiency, i.e. that the defect could have been avoided in development. The judicial process frequently allows the so-called "to all appearances" criterion to apply as proof. The producer can (and must!) then prove that it is not responsible for the defect, because in the course of conducting business it observed due diligence with regard to the design and production of the product. To ensure that this proof is even possible, it is absolutely essential to maintain and be able to verify analytical and design-related quality assurance measures.

What should be done from a legal perspective?

The producer must fulfill the organizational duties cited above and must also be able to prove fulfillment of these duties at a later time. This requires that the producer rely on effective Contract and Project Management and then also apply it. The minimum requirements here are:

- Structured development processes subdivided into phases.
- Clear, complete, unambiguous and understandable specifications including testing and acceptance criteria.
- Procedures for Requirements and Change Management.
- Quality Assurance measures.

Technical Safety Requirements

In the area of technical safety requirements, in the year 2002 the basic international safety standard IEC61508 was adopted into the German body of standards and was published as DIN EN61508. It represents the foundation for the state of the art in software-based safety-related systems.

IEC61508 formulates organizational requirements with rules for functional safety (management and evaluation), document management and definition of a safety life cycle in the context of total process orientation.

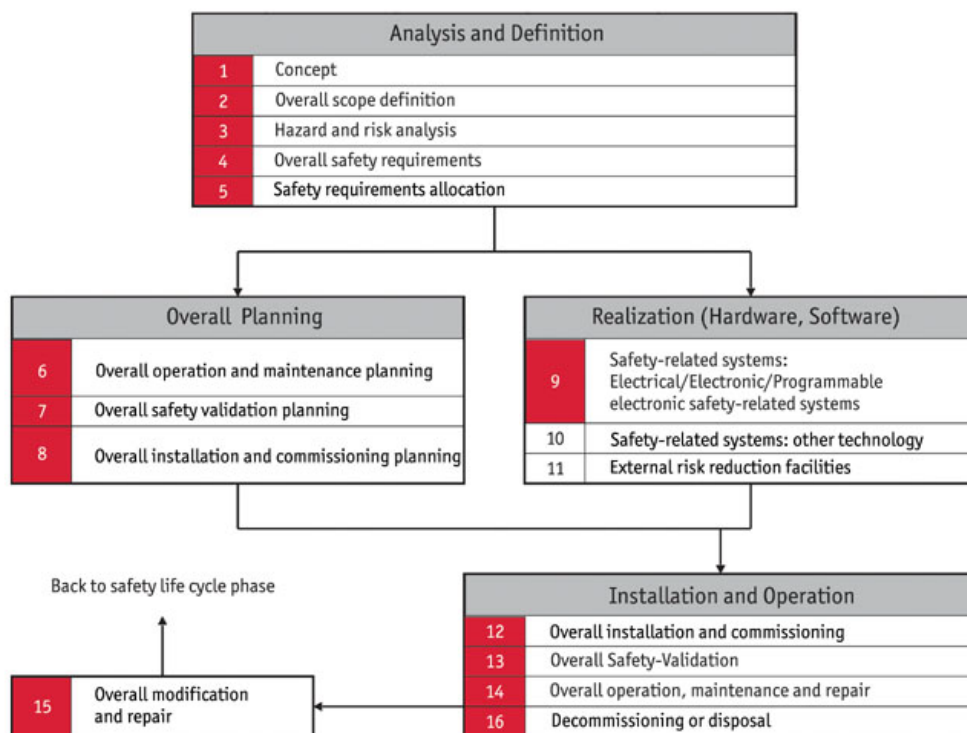


Figure 1: Safety Life Cycle per IEC61508

Technical requirements relate primarily to the safety-related integrity of the hardware architecture, i.e. the probability of random hardware failures and systematic safety-related integrity for avoiding and handling fail-

ures. The latter should be applied to software and involves implementing recommended practices and methodologies in each phase of the software safety life cycle.

In mapping IEC61508 requirements to the three levels of the Automotive Systems Engineering [6] approach - Engineering Processes, Management Processes and Process Management - it is clear that the requirements emphasize the level of Management Processes.

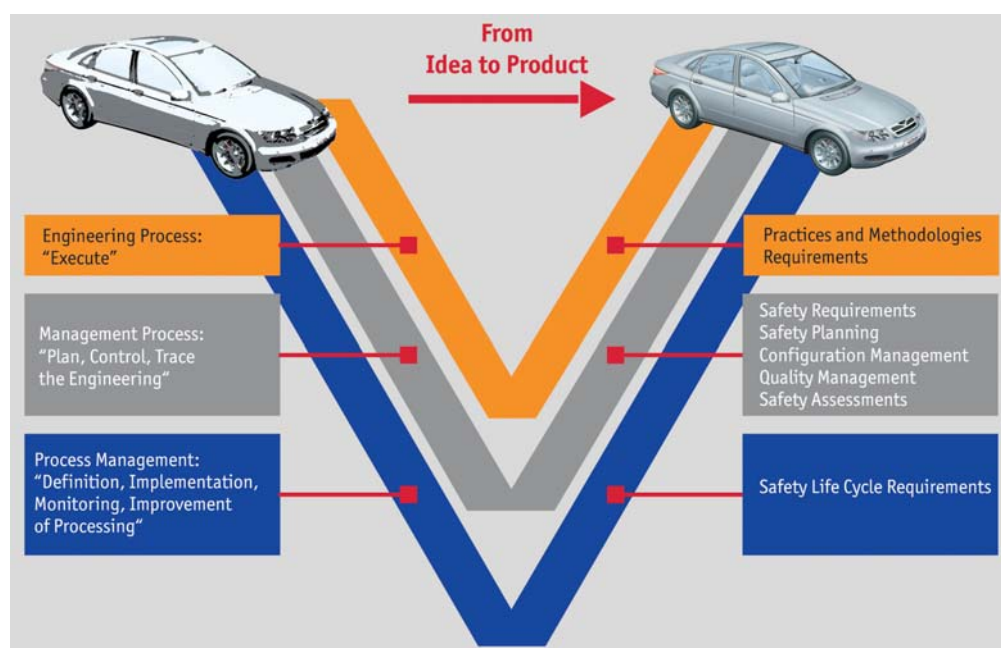


Figure 2: Mapping IEC61508 Requirements to the Process Levels of Automotive Systems Engineering

Requirements for a Safety-relevant Development Process

Accordingly, a safety-relevant development process must possess certain characteristics in the areas of Requirements, Project, Configuration and Quality Management. These process areas are described in Maturity Models such as CMM [7], CMMI [4] and SPICE [5]. A development process that fulfills Maturity Level 2 of this model already represents a good foundation for a IEC61508-conformant development process and also already fulfills the minimum requirements defined above from a legal perspective.

Based directly on this observation one can derive a way to achieve a safety-relevant development process with Process Management measures in three steps:

- Determine the process maturity level, based on a maturity model.
- Perform actions needed to reach the minimum Maturity Level 2 and simultaneously increase the efficiency of the development process.
- Consider specific safety-relevant requirements as complementary process building blocks, e.g. performing hazard and risk analysis to determine requirements for safety integrity or assessing functional safety.

This stepwise approach to achieving a safety-relevant development process can therefore be viewed as a measure taken as part of continuous process improvement.

Summary

The systematic application of CMM, CMMI and SPICE maturity models represents an important foundation, not only in fulfilling the requirements of IEC61508 but also in satisfying minimum requirements from a legal perspective. Starting from this foundation effective Process Management is needed to establish, in successive steps, an IEC61508-conformant development process and to thereby deal with the issue of producer liability proactively.

In conclusion: To achieve a safety-relevant IEC61508-conformant development process very little needs to be restructured. It is possible, and advisable, to build precisely upon proven methodologies, provided that they meet the (minimum) requirements presented.

Revised: 07/2004

Word count: 1,074

Character count: 9,085

Figure 1: Safety Life Cycle per IEC61508

Figure 2: Mapping IEC61508 Requirements to the Process Levels of Automotive Systems Engineering.

All figures: Vector Consulting GmbH

Literature References

1. "European Automotive Market for X-by-wire Technologies", Frost & Sullivan, 2001
2. TREAD Act:
www.ipa.fhg.de/Arbeitsgebiete/BereichA/210/leistungsangebote/Tread_Act/
3. IEC61508 "Functional safety of electrical/electronic/programmable electronic safety-related systems", www.iec.ch
4. CMMI: www.sei.cmu.edu
5. SPICE: www.isospice.com
6. D. Lederer, J. Fetzer, G. Heling, G. Baumann, "Automotive Systems Engineering - The Solution for Complex Technical Challenges?", Proceedings 5th International Stuttgart Symposium on Motor Vehicles and Combustion Engines, Feb. 2003, pp. 593-607
7. M.C. Paulk, Ch.V. Weber, and B. Curtis, "The Capability Maturity Model. Guidelines for Improving the Software Process", Addison-Wesley, 1995

Authors

Dr. (Engineering) Klaus-Jürgen Amsler, Senior Consultant at Vector Consulting GmbH

Tel. +49-711/80670-284, Fax +49-711/80670-444,

E-Mail: klaus-juergen.amsler@vector-consulting.de

Dr. (Engineering) Joachim Fetzer, Director of Vector Consulting GmbH

Tel. +49-711/80670-150, Fax +49-711/80670-444,

E-Mail: joachim.fetzer@vector-consulting.de

Dr. Meinhard Erben, Legal Counsel for IT Law

Tel. +49-711/80670-0, Fax +49-711/80670-111,

E-Mail: meinhard.erben@vector-informatik.de

Vector Consulting GmbH

Ingersheimer Str. 24

D-70499 Stuttgart Germany

www.vector-consulting.de

Editorial contact person: Holger Heit

Tel. +49-711/80670-567, Fax +49-711/80670-555,

E-Mail: holger.heit@vector-informatik.de