JOURNAL OF SOFTWARE: EVOLUTION AND PROCESS
J. Softw. Evol. and Proc. (2012)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/smr.580

Improving safety and availability of complex systems by using an integrated design approach in development

Volker Bachmann[1,*,†] and Richard Messnarz[2]

[1]*SIBAC GmbH, Biberach, Germany*
[2]*ISCN Ltd., Graz, Austria*

ABSTRACT

Within the last 5 years the need for a system development and a process that describes this development became more and more obvious. The number of software and electronic engineers rose even in companies that were traditionally working in the field of pure mechanics. The ISO standard 15504 was consequently expanded from the software to the system. Nevertheless, even 5 years after this change, there is nearly no subsystem mechanics completely described in the same tool as the subsystems electronics and software. This break within the tool chain is the tip of an iceberg reaching all the way down to the lived processes in development.

This paper tries to give a solution that was worked out in a group of integrated designers that developed a program for the European Certification and Qualification Association. It shows an example that was tested in an industry project to reach level two, according to the Automotive SPICE standard, on system level in a customer assessment. It shows how to bring together standards that are used in electronics, software, and mechanics to find an integrated design approach to improve safety and availability of systems composed of these parts. Namely, this is the IEC 61508 asking for a risk analysis, which again finds entrance into the failure mode and effects analysis (FMEA) that is a commonly known tool in mechanical development. The FMEA again has an interface to the ISO 15504, which is described here as well.

An integrated design is the outcome of this process by using the tools mentioned and bringing them together properly. Copyright © 2012 John Wiley & Sons, Ltd.

Received 18 July 2011; Accepted 18 July 2011

KEY WORDS: integrated design; FMEA; safety standards

1. INTRODUCTION

Within the last 15 years there was a great shift of the number of employees in companies working in production to those working in development. The reason is that, on one hand, automation in production is reducing the number of employees in this area. On the other hand, the complexity of mechatronical systems is increasing the number of employees in the development sector. Whereas it was not so much the number of the mechanical engineers that increased, there are often more engineers working in the field of electronics and software than there are mechanical engineers. This leads on the one hand to more complex structures and processes within the development departments asking for a standard like ISO 15504, but on the other hand these complex systems that need to be developed ask for engineers who are capable of using an integrated design approach. This integrated design approach must be supported by tools which are already known and in use today [4,5].

Within the last 15 years there have been a number of standards coming up, which 10 years ago stood by themselves. The best known example is the previously mentioned ISO 15504, which was originally formulated for software development. After a few years it became apparent that it should spread out on

*Correspondence to: Volker Bachmann, SIBAC, Biberach, Germany.
†E-mail: info@sibac.de

the whole system. However, until today there is almost no subsystem description on a mechanical subsystem existent. The two worlds of mechanics and electronic/software still live an absolutely parallel existence in every large company in the automotive industry as far as documentation is concerned. Since the documentation can be used as a sign for the development direction, it can also be stated that these two worlds live next to each other in development as well.

Today, many developers believe that most improvement potential lies within mechatronics. However, there is a strong misunderstanding of what mechatronic comprises. Usually, mechanics is not counted as part of mechatronics but only actuators, sensors, and software.

This paper tries to convince that mechanics is an essential part of mechatronical development, and the improvement potential of the future lies in an integrated design approach comprising everything: software, hydraulics, electronics, and mechanics. This is meant as a first attempt to bring known tools and known processes out of these different worlds together to support this approach.

2. SAFETY AND AVAILABILITY IN HYDRAULIC AND MECHATRONIC SYSTEMS

2.1. The risk analysis as a basis for safety evaluation

In the following, a brief example is given to lead to an evaluation of a major fault in an anti-lock braking system (ABS). The general knowledge of a brake system is assumed to be known.

This ABS will be analyzed regarding its possible risk potential. The example describes the ABS without the existence of any safety measures available. Necessary safety measures will be an outcome of this analysis.

At the beginning of the development a system definition is necessary. The following example shows a realistic case in which there is still no completely developed product available. At this point at the very start of a development project the risk analysis should be carried out.

Figure 1 shows a typical ABS consisting of four wheel-speed sensors, a central control unit, and actuators on every wheel [2].

In this example there is only one central electronic control unit (ECU) planned for the system that evaluates the signals coming from the four wheel-speed sensors and controlling the valves at every wheel. The original task is to detect wheel slip during a braking situation and to avoid it before it actually occurs.

The challenge is to design the system in such a way as to prevent dangerous failures or to control them when they arise. During hazard identification the regarded system is considered without the existence of any E/E/PE-safety measures. Necessary safety measures will be an outcome of this analysis. A hazard is defined in the standard (ISO 61508) as a ‘potential source of harm’. The risks associated with unidentified hazards will remain unreduced. The identified hazards are input for the hazard analysis and risk assessment.

The only hazard used for this example, which will be looked at in detail, describes the situation that the control unit actuates the valves in a way that building up pressure in the hydraulic brake system becomes impossible. That means that although the driver is pressing the brake pedal, he cannot apply brake force at the wheel.

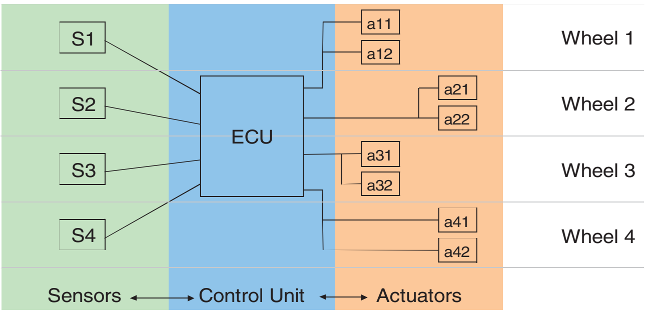


Figure 1. Schematic of an ABS brake System.

DOI: 10.1002/smr

Hazard analysis is the study of the chains of cause and effect between the various identified hazards and the hazardous events to which they might lead, and of the consequences of the hazardous events. The purpose of this analysis is to derive sufficient information for the assessment of the risks involved. There are two elements of risk, the likelihood of something happening and the potential consequence if it does. Understanding the various causes of a hazardous event allows a calculation or estimation of its likelihood.

A hazard is categorized with regard to the worst case it can cause. Applied to the shown example: ABS avoids build up of hydraulic pressure, which means no brake-force at the wheels. Driving fast and having any kind of obstacle in front of the vehicle illustrates the consequences. If a possible obstacle consists of people, the consequences are even worse.

The method used in this example is the risk graph method. In this method a number of parameters are introduced, which together describes the nature of the hazardous situation when safety-related systems fail or are not available. One parameter is chosen from each of four sets, and the selected parameters are then combined to decide the safety integrity level (SIL) allocated to the safety-related systems.

These parameters:

- allow a meaningful graduation of the risks to be made, and
- contain the key risk assessment factors.

For the chosen example, the following is the case:

Consequence: C3
Comment: Without the possibility to slow down a crash is possible. Depending on the speed, passengers in the car and traffic participants could be badly injured or killed. Death of several people is possible.
Frequency: F2
Comment: Braking at medium or higher speed is an every-day situation.
Possibility: P2
Comment: A car without operating braking system is for most drivers not controllable. The average driver and other traffic participants are normally not able to avoid harm in this situation.
Probability: W1
Comment: Today antilock braking systems are state-of-the-art. Many years of experience and testing in the field state that this occurrence is quite improbable. Established development processes are available.

The risk parameters C3 / F2 / (P2) / W1 result in a safety integrity level 3 (Figure 2)

In Part 4 of the standard, safety integrity is defined as ‘the likelihood of a safety related system satisfactorily performing the required safety functions under all the stated conditions, within a stated period of time’, and a SIL as ‘a discrete level (one of 4) for specifying the safety integrity requirements of safety functions’.

Every hazard can be verbalized as a safety requirement or a so-called safety goal for the safety-related system. Here, the safety goal could be: ‘ABS must ensure a safe build-up of hydraulic pressure’.

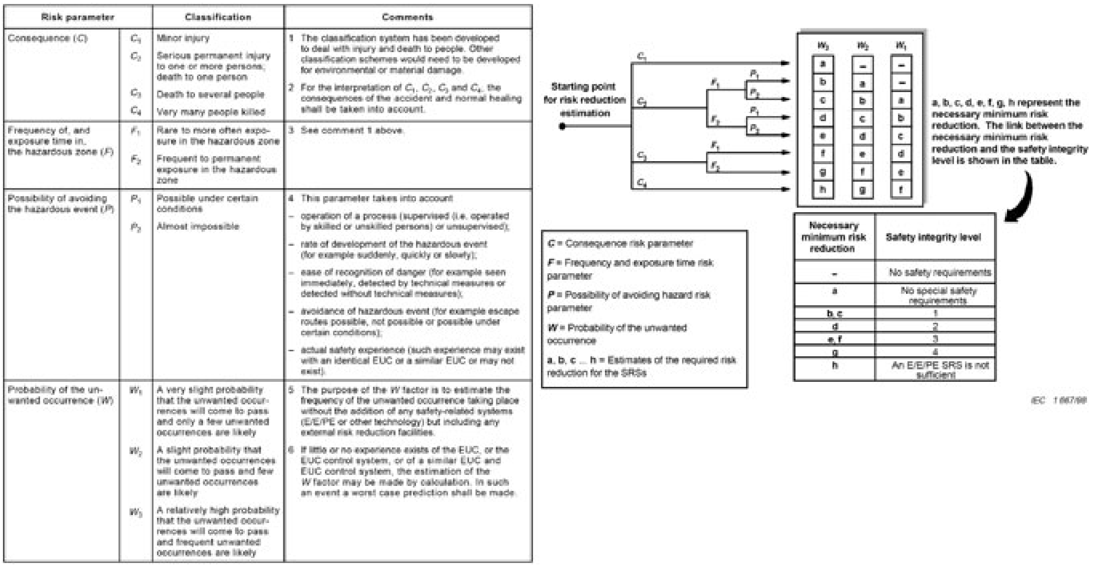
Any safety-related system covers all parts of the system that are necessary to carry out the safety function (i.e., from sensor, through control logic and communication systems, to final actuator, including any critical actions of a human operator).

‘The major difference between a thing that might go wrong and a thing that cannot possibly go wrong is that when a thing that cannot possibly go wrong goes wrong, it usually turns out to be impossible to get at or repair.’ – Douglas Adams, author of The Hitchhiker's Guide to the Galaxy.

2.2. *The advantage of an integrated design derived out of safety and availability evaluations*

The previous chapter showed that the risk analysis should stand at the very beginning of a system development. Its output is a number of hazardous events that are categorized in SIL. The hazardous events are used as an input in the System FMEA on the top level (failure effects in Figure 3 of the system FMEA). The SIL classification determines the severity in the FMEA. The system should

DOI: 10.1002/smr



Risk parameter		Classification	Comments
Consequence (C)	C_1 C_2 C_3 C_4	Minor injury Serious permanent injury to one or more persons; death to one person Death to several people Very many people killed	1 The classification system has been developed to deal with injury and death to people. Other classification schemes would need to be developed for environmental or material damage. 2 For the interpretation of C_1, C_2, C_3 and C_4, the consequences of the accident and normal healing shall be taken into account.
Frequency of, and exposure time in, the hazardous zone (F)	F_1 F_2	Rare to more often exposure in the hazardous zone Frequent to permanent exposure in the hazardous zone	3 See comment 1 above.
Possibility of avoiding the hazardous event (P)	P_1 P_2	Possible under certain conditions Almost impossible	4 This parameter takes into account – operation of a process (supervised (i.e. operated by skilled or unskilled persons) or unsupervised); – rate of development of the hazardous event (for example suddenly, quickly or slowly); – ease of recognition of danger (for example seen immediately, detected by technical measures or detected without technical measures); – avoidance of hazardous event (for example escape routes possible, not possible or possible under certain conditions); – actual safety experience (such experience may exist with an identical EUC or a similar EUC or may not exist).
Probability of the unwanted occurrence (W)	W_1 W_2 W_3	A very slight probability that the unwanted occurrences will come to pass and only a few unwanted occurrences are likely A slight probability that the unwanted occurrences will come to pass and few unwanted occurrences are likely A relatively high probability that the unwanted occurrences will come to pass and frequent unwanted occurrences are likely	5 The purpose of the W factor is to estimate the frequency of the unwanted occurrence taking place without the addition of any safety-related systems (E/E/PE or other technology) but including any external risk reduction facilities. 6 If little or no experience exists of the EUC, or the EUC control system, or of a similar EUC and EUC control system, the estimation of the W factor may be made by calculation. In such an event a worst case prediction shall be made.

W_3	W_2	W_1
a	–	–
b	a	–
c	b	a
d	c	b
e	d	c
f	e	d
g	f	e
h	g	f

a, b, c, d, e, f, g, h represent the necessary minimum risk reduction. The link between the necessary minimum risk reduction and the safety integrity level is shown in the table.

Necessary minimum risk reduction	Safety integrity level
–	No safety requirements
a	No special safety requirements
b, c	1
d	2
e, f	3
g	4
h	An E/E/PE SRS is not sufficient

C = Consequence risk parameter

F = Frequency and exposure time risk parameter

P = Possibility of avoiding hazard risk parameter

W = Probability of the unwanted occurrence

a, b, c ... h = Estimates of the required risk reduction for the SRSs

IEC 1 667/98

Figure 2. Risk graph method according to IEC61508-5:1998, Annex D [1,3].

have the top level functions. To each function there must be at least one failure. Each event out of the risk analysis should be equal to one of these failures. The failures are linked together in a failure tree as shown in Figure 3. Most important in this failure tree aspect is that there are as many blocks below the wide orange line as there are subsystems. Instead of System FMEA ECU there should be also a System FMEA calliper, sensors, actuators, wiring harness, and so on.

Out of this failure tree every potential subsystem failure can be traced to the events found in the risk analysis. This way the contribution of every subsystem to avoid this event becomes visible and helps to get an overview for an integrated system design.

3. THE ROLE OF LOGIC IN MECHANICS, HYDRAULICS, ELECTRONICS, AND SOFTWARE

Logical functions can be built in any of the four systems mechanics, hydraulics, electronics, and software. Since the opportunities of electronics and software boosted development in the past years, a coordination between the above mentioned systems becomes more and more important.

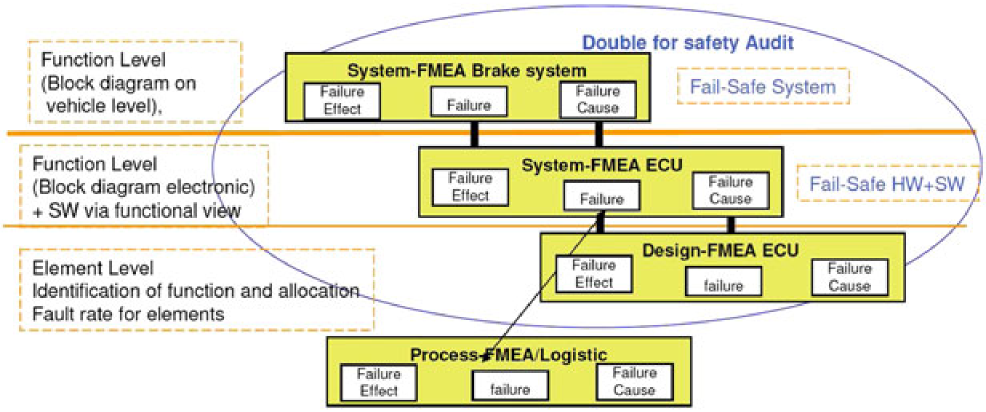


Figure 3. Failure Mode and Effects Analysis (FMEA)

DOI: 10.1002/smr

3.1. Description of the current situation

3.1.1. Fail safe systems. At present, there are still many systems in production that do have an ECU, but they still rely on mechanical and hydraulic systems as far as safety is concerned. This is especially the case in the automotive industry for example for automatic transmissions and steering systems. Whereas the pure steer-by-wire is still prohibited by law in all major industrialized countries because of the high SIL, shifting by wire is allowed and already included in production by some transmission builders. What quite frequently occurs is the fact that the development departments do not trust the electronics/software department enough to lay full responsibility for failure reactions in their hands. Instead there is often a logic within the hydraulics that actuates for example the park pawl in case of a single failure (a chain of two or more failures is not analyzed in SIL 2 Systems). In steering systems only adding force to the steering rack is allowed. The original mechanical steering column still exists and stands for the fall back basis.

3.1.2. Sensors and diagnostics. Software diagnostics need input signals. These signals can only be provided by sensors. Sensors again cost money. In today's structures of large automotive companies and their suppliers it is merely impossible to think about the vehicle as a system and to find solutions to requirements outside of a subsystem such as the engine, the steering system or the transmission. Even a very simple problem of a transmission builder stating that a better torque signal coming from the engine over the Control Area Network (CAN)-bus could help a great deal to improve shift quality cannot, in today's structures, be solved in a way to analyze which way is best in this case. The idea of a supplier going to his or her original equipment manufacturer and analyze together whether it makes the transmission significantly cheaper if he or she gets a better torque signal can conjure up a smile on every developer's face.

Nevertheless, in today's systems in which it is possible to get information from any sensor in the system this approach needs to be pushed far more than it is today. Only for the past two or three years is software diagnostics consequently represented in the FMEA. This leads to major irritations in the beginning also because the FMEA tools did not provide this functionality. Meanwhile, the FMEA tools are able to handle fault reactions, but the schematic on how to represent software diagnostics in these tools is still at the very beginning as far as the knowhow of the FMEA moderators is concerned. Out of today's point of view, the combination of FMEA and Software Process Improvement and Capability Determination (SPICE) is the best way to handle this problem properly.

3.1.3. The integrated design approach in ISO 15504. The FMEA needs to be worked out thinking in functions and failures to be able to directly compare the results of the FMEA with the requirements formulated for the engineering processes in SPICE. This needs to be done for every subsystem involved. Until recently, this was not the case. Until recently a designer (for mechanics) thought much more in solutions and structures. The designer for example always knew what part he was working at and what attributes he had to take care of. However, it was not always clear what the requirements for the attributes were from a functional point of view. Of course, the designer had to take over many attributes like material, surface roughness, etc. from previous designs to keep efficiency high. Taking over the solutions often helps to answer the question on what the solution looks like but not what problem had to be solved.

Similar to that is the behavior in software development. In software construction it often makes sense to combine modules out of libraries to keep efficiency high. Most important is to fulfill the requirements of the software architectural design. That does not help to answer the question what system requirement is fulfilled with this module.

In the future the developers of each subsystem have to think in functions and requirements to be able to draw links from one level to the next.

3.1.4. System development teams allocate requirements. Figure 4 shows the link between the requirements of the engineering processes and the failures of the FMEA. Also, the measures of the FMEA that are linked to the failures are shown in comparison to the test-cases of the engineering processes. The wording is still exclusively for the software, but as it was rolled out to the whole system, the v-model is similar in all the other subsystems.

DOI: 10.1002/smr

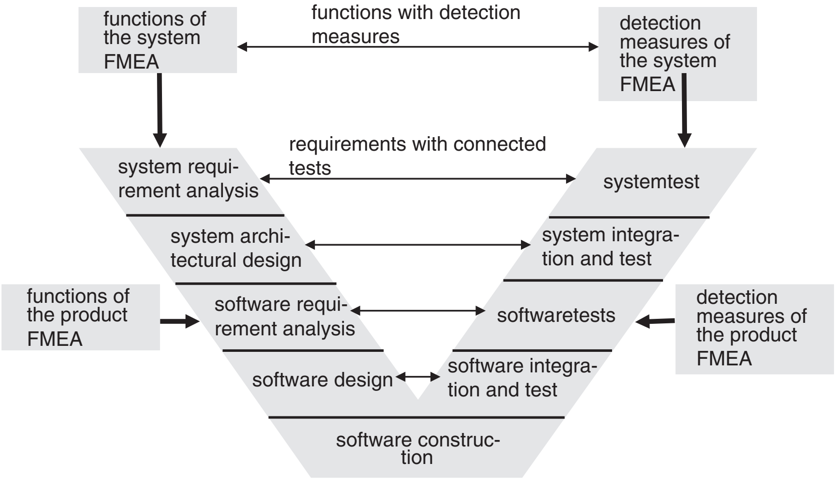

functions of the system FMEA
functions with detection measures
detection measures of the system FMEA
system requirement analysis
requirements with connected tests
systemtest
system architectural design
system integration and test
functions of the product FMEA
software requirement analysis
softwaretests
detection measures of the product FMEA
software design
software integration and test
software construction

Figure 4. Engineering processes of the V-Model in comparison to functions and detection measures in the FMEA.

The demands of the standard (ISO 15504) do not only help the customers of these systems (e.g., the original equipment manufacturers) but also the developers of these complex systems, as it enables them to run analysis concerning the project progress in many fields. Such analysis can be, for example: How many of my system-requirements are linked to the subsystem-requirements (which answers indirectly the question on how many design-decisions already exist), or how many of the subsystem-requirements are already positively tested.

Taking this description as a basis, it becomes clear that the complete system description of the requirements of a mechatronical product need to be described. The point of view is different compared with the FMEA. It is the attempt to get an overview over complex systems by subdividing them into ever smaller parts. Since the requirements have to be tested down to the lowest subsystem-level, the connection to the FMEA is obvious for mechanical systems. As described in the previous chapter in mechatronical systems, it is no longer the mechanics alone that is responsible for safety functions. However, diagnostics-software in particular provides the possibility to avoid hazardous situations in case of a mechanical failure or go into a so called limp home mode to raise the availability of a system.

Going from this example to an overall understanding, the following Figure 5 explains what most frequently happens.

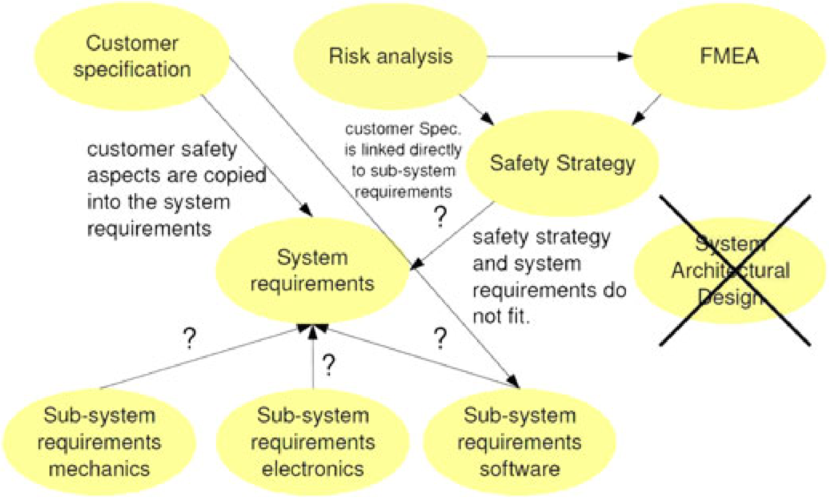

Customer specification
Risk analysis
FMEA
customer safety aspects are copied into the system requirements
customer Spec. is linked directly to sub-system requirements
?
Safety Strategy
safety strategy and system requirements do not fit.
System requirements
System Architectural Design
?
?
?
Sub-system requirements mechanics
Sub-system requirements electronics
Sub-system requirements software

Figure 5. Normal way of trying to fulfill customer requirements and problems arising out of this method.

DOI: 10.1002/smr

Usually the customer requirements concerning safety are written in a detail that reaches down to every subsystem [6]. Usually the fastest way to get a good coverage on links to the customer specification is to link directly to the subsystem. This most certainly gives excellent metrics over development time in the beginning of the project. The problems arise later in the project. A risk analysis needs to be carried out to fulfill the IEC 61508 as described in the previous chapters. The outcomes are usually taken as an input for the FMEA. The FMEA makes all the dependencies between the subsystems clear and thus leads to a safety strategy for the product. If the safety relevant aspects coming from the customer are directly linked into the system and subsystem requirements, it is not possible to adapt to the safety strategy anymore. Problems in this field usually inhibit working out a system architectural design because it does not fit the system requirements at all. In the end there are major problems in linking the subsystems to the system since some of the links are drawn directly from the customer spec. All these problems lead to a zero in an assessment for the processes Engineering 2 and 3 and usually also Engineering 9 and 10.

The above figure already somewhat contains the solution that is shown in Figure 6.

Using this approach the risk analysis is carried out for the product just as in the previous example with the only difference that the customer requirements may have an influence on the cases which are considered in the risk analysis. That means that every new customer that asks for the product may enlarge the risk analysis by points or aspects that were not considered until then.

The outcome is taken as an input into the FMEA, which again is used to create a fitting system architectural design. This design is the basis of every decision on how to fulfill system requirements in subsystem requirements. It is a need to take these design decisions to be able to draw the links from the subsystem to the system requirements. The requirements tree that ends in the system requirements is now used to prove that all the customer specification requirements are met. Work in an interdisciplinary team is essential to be able to do this. In this team there is at least one specialist for every subsystem needed just as much as the system designer. It is not possible to come to good design decisions without these interdisciplinary teams. Engineering teams have to get interdisciplinary, and thus demand for a mutual understanding and collaboration between domain expert team members is necessary. For instance the customer (see Safety Team in collaboration with customer figure 7) will closely collaborate to transform safety requirements into system requirements which avoid hazards and risks. The internal development team will communicate and collaborate (see Internal Team developing safe system and Safety Team in collaboration with customer figure 7) to implement the system and different components fulfilling the safety requirements. And a core issue is the role of a Systems Architectural Team in close collaboration between the systems

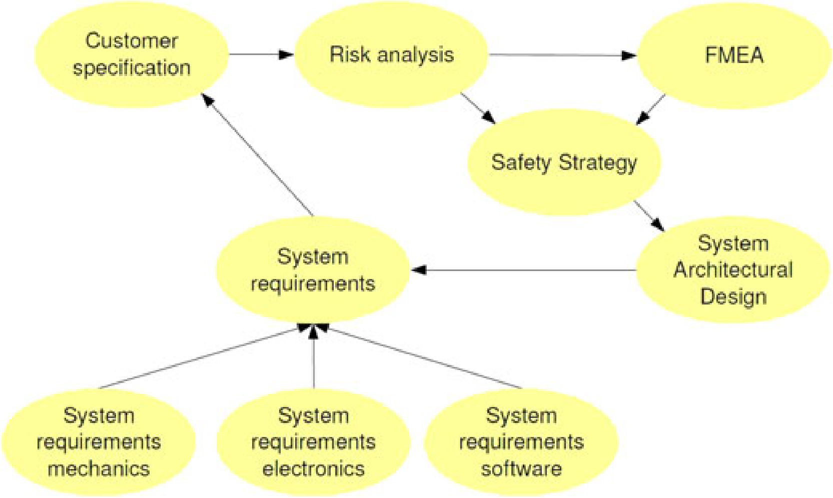

Customer specification
Risk analysis
FMEA
Safety Strategy
System requirements
System Architectural Design
System requirements mechanics
System requirements electronics
System requirements software

Figure 6. Integrated design approach to fulfill customer requirements.

DOI: 10.1002/smr

architect, the systems requirements manager, and the safety concept responsible. If these three core people have problems in communication the whole project is endangered. If these three form a coherent team and have good level of understanding with the software department, the safety concepts work out and budget overruns will not take place.

4. SAFETY AND THE INFLUENCE ON TESTING ON SYSTEM LEVEL

Considering functional safety, the conflict of goals within the FMEA is solved by changing the severity of the consequence in case of a failure as long as there is a diagnostics software existent that can lead to a different consequence. Using the example of the transmission in which the failure cause for the wrong starting direction can be found in the assembly group level and the consequence is on the level of the vehicle, the severity for this consequence is 10. The consequence can now be changed within the FMEA to a different case taking the exact same failure cause to an eight since starting in the wrong direction was changed by the diagnostic software to a stranded vehicle. This is not relevant to safety anymore. Of course, a detection measure has to be formulated that proves that the diagnostic software and the according actuators and mechanics work properly on system level. This can best be proved in the requirements tree as shown in Figures 4 and 6.

Working with the SPICE processes in the example of the transmission, there should be a requirement on system level that asks for a detection when starting off in the wrong direction. This would lead to requirements on subsystems such as in the software asking for diagnostics software and in electronics asking for a sensor for the shift lever position and a sensor measuring the direction of rotation of the output shaft. Each subsystem requirement and the system requirement must be linked to a test. The system architectural design is influenced by a great deal and since this is also very much a question of money these decisions need to be based on consequences for the vehicle.

Solutions must be found to integrate the tools used today to avoid double work in the future. In a first step there must be a description of the interface, for instance, of the IQ-FMEA and DOORS or MKS. Today these problems must still be solved through an indirect way via an html export with an import into the next tool.

Working according to this approach does not only help to get proper design decisions but also leads to a level two in an assessment for Engineering 2 and 3, and in case all the tests are existing and linked, also Engineering 9 and 10. If this is achieved the same way for different customers and different products as described in Figure 6, even a level three is possible.

5. CONCLUSIONS

Most engineers today agree that the largest potential for improving systems in the near future can be found in mechatronical development. A large misunderstanding is still on the way on what belongs to a mechatronical system. For most mechanical engineers this is the combination of actuators, sensors, electronics, and software. This paper showed that the mechanics is also an essential part of a mechatronical system. Since there is a clear tool chain break in documentation systems between mechanics and electronics/software, the processes do not work hand in hand. This paper showed how an integrated design approach can lead to an improvement of the product system stability and thus to a higher level of safety and availability.

This paper is meant to be part of a strategy for a tool-based development by describing the interfaces of tools that are commonly known and used in the different faculties. The tools with their interfaces described are the standards ISO 15504, IEC 61508, and the FMEA.

Working according to the approach of an integrated design demands engineering teams that work in an interdisciplinary manner. There is a need for a mutual understanding and collaboration between domain expert team members.

DOI: 10.1002/smr

REFERENCES

1. http://www.pdf-search-engine.com/iec-61508-pdf.html
2. http://en.wikipedia.org/wiki/Anti-lock_braking_system
3. IEC 61508:1998
4. Poth A. SPI of the Requirements-Engineering-Process for Embedded Systems Using SPICE. *Proceedings of the EuroSPI 2006 Conference.*
5. Spork G. Establishment of a Performance Driven Improvement Program. *Proceedings of the EuroSPI 2007 Conference*.
6. Bachmann V, Messnarz R. Improving the Software-Development for multiple Projects by applying a Platform Strategy for Mechatronic Systems. *Proceedings of the EuroSPI 2009 Conference*.

AUTHORS' BIOGRAPHIES:

Volker Ovi Bachmann is the managing director of SIBAC GmbH. He studied at Darmstadt University from 1988 until 1993 and received a degree in mechanical engineering. From 1993 to 1998 he worked as an assistant at the automotive department of Darmstadt University as researcher and lecturer and received a degree as Dr.- Ing. Since 1998 he has been working in the automotive industry for ZF Friedrichshafen AG in several positions the latest being team manager in the quality management department. He was in charge of seven employees and the quality management of five large transmission development projects (with more than 60 engineers per project involved). During his time in the industry he received the qualification as a Spice Assessor. He has an experience of 120 assessment hours. In 2008 he become a competent assessor. In January 2008 he founded SIBAC GmbH. Since then he has been working as a process consultant at various companies. E-mail: info@sibac.de

Richard Messnarz is the Executive Director of ISCN LTD. He studied at the University of Technology Graz and he worked as a researcher and lecturer at this University from 1991 to 1996. In two European mobility projects (1993 and 1994) he was involved in the foundation of ISCN, and he became the director of ISCN in 1997. He is/has been the technical director of many European projects: PICO–Process Improvement Combined Approach, 1995–1998; Bestregit–Best RegionalTechnology Transfer, 1996–1999; TEAMWORK–Strategic Eworking Platform Development and Trial, 2001–2002; MediaISF–Eworking of media organisation for strategic collaboration on EU integration, 2001–2002. He is the editor of a book 'Better Software Practice for Business Benefit', which was published by IEEE (www.ieee.org) in 1999. He is the chairman of the EuroSPI initiative and chair of the programme committee of the EuroSPI conference series. He is author of many publications in e-working and new methods of work in conferences of the European Commission (E-2001 in Venice, E-2002 in Prague), and in the magazine for software quality (Software Quality Professional) of the ASQ (American Society for Quality). He is a lead ISO 15504 assessor. He has worked as a consultant for many automotive firms, such as BOSCH, ZF TE, ZF N, Continental TEMIC, Audi/VW, etc. He is a founding member of the INTACS (International Assessor Certification Scheme) accreditation board, a founding member of the Austrian Testing Board, a founding member of the Configuration Management Board, and he is the technical moderator of the SOQRATES initiative (www.soqrates.de). E-mail: rmess@iscn.com

DOI: 10.1002/smr