Dissertation

Eine Entwicklungsmethodik für sicherheitsrelevante Elektroniksysteme im Automobil

von

Stefan Benz

Universität Karlsruhe (TH) - 2004

Eine Entwicklungsmethodik für sicherheitsrelevante Elektroniksysteme im Automobil

Zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEURS

von der Fakultät für Elektrotechnik und Informationstechnik
der Universität Friedericiana Karlsruhe

genehmigte

DISSERTATION

von

Diplom-Ingenieur Stefan Benz

aus

Böblingen

Tag der mündlichen Prüfung:	20.04.2004
Hauptreferent:	Universitätsprofessor Dr.-Ing. Klaus D. Müller-Glaser
Korreferent:	Universitätsprofessor Dr.-Ing. Winfried Görke

für Corinna

Zusammenfassung

Im Automobilbereich ist ein deutlicher Anstieg des Elektronikanteils bei der Wertschöpfung eines Kraftfahrzeugs zu beobachten. War der intensive Einsatz von Elektronik bisher überwiegend auf Motorsteuerungs- oder Komfortsysteme beschränkt, so werden zukünftig auch sicherheitsrelevante Systeme wie Lenkung und Bremse durch sogenannte X-by-Wire-Systeme kontrolliert werden und sich so die aktive und passive Sicherheit des Fahrzeugs weiter erhöhen. Dies birgt aber die Gefahr, dass bei Fehlern oder gar dem Ausfall derartiger Systeme die Fahrzeuginsassen in höchstem Grad gefährdet werden können.

Zur Entwicklung solcher neuer Elektroniksysteme ist daher eine Anpassung des bestehenden Entwicklungsprozesses notwendig. Mit höheren Anforderungen an Sicherheit und Zuverlässigkeit der verwendeten Systeme ändern sich auch die Anforderungen an den eingesetzten Prozess. Wurde bisher die Funktion des Systems in den Mittelpunkt der Entwicklung gestellt und andere, nicht-funktionale Randbedingungen zusätzlich berücksichtigt, so muss zukünftig Sicherheit und Zuverlässigkeit des betrachteten Systems gleichwertig mit seiner Funktion betrachtet werden.

Heute orientiert sich das prinzipielle Vorgehen bei der Entwicklung elektronischer Systeme im Automobilbereich am V-Modell ’97. Vor allem die Möglichkeit der Generierung von Testfällen wird sehr intensiv genutzt. Allerdings ist das Vorgehen weder standardisiert noch formalisiert, jeder Hersteller und Zulieferer hat hier einen etwas anderen Prozess.

In der Automobilbranche ist man dabei, zusätzliche Erfahrung bei der Entwicklung von sicherheitsrelevanten bzw. hochzuverlässigen Elektroniksystemen aufzubauen. In Bereichen wie der Luftfahrt, wo es standardisierte Prozesse gibt, ist vergleichsweise viel Erfahrung vorhanden. Diese lässt sich aber nicht ohne weiteres auf den Automobilbereich übertragen, da in diesen beiden Bereichen recht unterschiedliche Randbedingungen existieren.

In dieser Arbeit wird ein Ansatz für eine neue Entwicklungsmethodik vorgestellt, die Sicherheit und Zuverlässigkeit verstärkt berücksichtigt. Sie basiert dabei auf dem im Automobilbereich eingesetzten und erprobten V-Modell ’97. Zur

Betrachtung von Sicherheitsanforderungen wird diese bestehende Vorgehensweise um ein zweites V („Sicherheit und Zuverlässigkeit“) erweitert, das mit dem bereits vorhandenen V („Funktion“) an entsprechenden Stellen verknüpft wird. Durch definierten Informationsaustausch zwischen den beiden „Vs“ wird die Durchgängigkeit der Methodik sichergestellt. Basis für die Erweiterungen hinsichtlich Sicherheit und Zuverlässigkeit sind Inhalte aus dem Luftfahrt-Standard SAE ARP 4761, die an Verhältnisse in der Automobilindustrie angepasst wurden.

Kernelemente der so entstandenen Entwicklungsmethodik sind die Verwendung einer Gefährdungsanalyse zur Bewertung des Systems als Basis für die Systementwicklung und der intensive Einsatz von Werkzeugen wie der Fehlerbaumanalyse und der FMEA zum Nachweis der Sicherheitseigenschaften des entwickelten Systems. Die Vorgehensweise wird in dieser Arbeit anhand von zwei Beispielen illustriert.

Englische Kurzfassung

In the automotive industry there is a clear trend to an increase in the number of electronic systems in a vehicle. More and more often mechanical implementations of vehicle functions are replaced by electronics or software systems.

One can find several reasons for this trend. Many functions in today's automobiles cannot be implemented without the extensive use of electronics. Systems that provide more comfort or additional functions such as navigation or "infotainment" rely heavily on electronics in the car. The recent achievements concerning lower fuel consumption, lower emissions, and at the same time greater engine power and performance due to better engine control are also to some extent the result of more and better powertrain control electronics. Other consequences of the extensive usage of electronics are lower costs for the manufacturer and the supplier and also lower running costs for the customer. Finally the most important achievement in the context here is definitely the increase of passive and especially active vehicle safety [Dai02].

It is clear that passive safety systems such as airbags or special body concepts tap their full potential. Further major improvements of vehicle safety through passive safety systems do not seem very likely. Future safety enhancements will have to rely mostly on active safety systems that actively avoid accidents before they can occur. One of the first active safety systems was the antilock braking system (ABS), state of the art today is the electronic stability program (ESP). Additional active safety functions that we will see in the future will culminate eventually in autonomous driving.

Active safety systems of the next generation will incorporate advanced direct interventions in brakes and also in steering. These functions will be implemented by so called x-by-wire systems, vehicle systems in which the transmission of energy and information is done in an electrical way only, i.e. mechanical or hydraulic system parts are replaced by electronics [DD02].

It is obvious that new safety-relevant electronic systems[1] impose strong requirements for safety and reliability. They therefore demand also new methods for system development and design. In the past the function of a system was in the center of focus and other, non-functional conditions such as safety were considered in addition. But for safety-relevant systems the safety of the vehicle system will have to be considered on the same level as its function, thus a thorough adaptation of the existing design process and its methodology is necessary.

The standard process for the development of automotive electronic systems is based on the "V Model ’97", a life cycle process model that is the development standard for IT systems in the Federal Republic of Germany (see [IAB02a]). Originally intended for IT systems it was adapted for other domains, including the automotive domain (see for example [Bor02]). However the methodical procedure is neither standardized nor formalized throughout the automotive domain, almost every manufacturer and supplier uses a slightly different design method.

In the automotive world some efforts are made to gain additional experience in the development of safety-relevant electronic systems. In the aerospace industry (a domain with very standardized and formal development processes) there is a lot of experience with safety-relevant systems. Aerospace development processes have function, and on an equal footing safety of an aerospace system, as their center of focus (see [Kne00] or [Bea00]). However these processes cannot be used directly in the automotive domain, as the boundary conditions and the requirements for safety and reliability in these two domains are quite different.

In this work an approach for a new design methodology is presented. It is based on the V Model ’97 as the established process model for the development of electronic and software systems in the automotive domain. For an advanced consideration of safety and reliability the existing process is extended by a second V (with process elements that have a special focus on safety and reliability) to a "Two V Model". The new elements are interconnected with the existing ones at several points of time during the development process. By a defined information exchange between the two Vs continuity in the methodology is guaranteed. Basis for the extension are contents of the aerospace standard SAE ARP 4761 that were adopted to automotive conditions.

The presented design methodology consists of nine interconnected steps: system requirements analysis, functional design, functional hazard assessment, system

[1] Safety-relevant systems are defined here as systems where failure conditions can lead directly to serious injury or death of one or several persons.

design, design-accompanying system safety assessment, implementation, integration and test, system safety assessment and approval and commissioning.

New core elements are the usage of a functional hazard analysis for an early assessment of the system as a basis for the system design and the intensive usage of tools such as fault tree analysis or FMEA. They are illustrated by two simplified examples.

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit im zentralen Bereich Forschung und Vorausentwicklung der Robert Bosch GmbH. Die wissenschaftliche Betreuung erfolgte durch die Fakultät für Elektrotechnik und Informationstechnik der Universität Karlsruhe (TH).

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. Klaus D. Müller-Glaser, kollegialer Leiter des Instituts für Technik der Informationsverarbeitung, für die Betreuung der Arbeit und die Übernahme des Hauptberichts. Durch sein stetes Interesse am Fortgang der Arbeit und durch viele wissenschaftliche Hilfestellungen machte er den erfolgreichen Abschluss dieser Arbeit erst möglich. Bei Problemen fand ich bei ihm immer ein offenes Ohr.

Herrn Prof. Dr.-Ing. Winfried Görke vom Institut für Rechnerentwurf und Fehlertoleranz an der Fakultät für Informatik danke ich herzlich für die Übernahme des Zweitberichts und seine sorgfältige Begutachtung der Arbeit. Seine stets konstruktive Kritik hat die Qualität dieser Arbeit deutlich gesteigert.

Der Robert Bosch GmbH, die mir die Erstellung dieser Arbeit im Rahmen des Doktorandenprogramms erst ermöglicht hat, bin ich zu besonderem Dank verpflichtet. Allen Kolleginnen und Kollegen aus der Arbeitsgruppe Software- und Systemtechnik sowie der Arbeitsgruppe Angewandte Informationstechnik der Forschungsabteilung Informations- und Systemtechnik möchte ich an dieser Stelle Dank sagen. Sie haben alle auf ihre eigene Art und Weise zum Gelingen dieser Arbeit, v. a. durch die Schaffung einer angenehmen, anregenden und freundschaftlichen Arbeitsatmosphäre, beigetragen.

Besonderer Dank gilt hier Dr. Bernd Müller für zahlreiche Anregungen in der Diskussion, die mit zu den Ergebnisse dieser Arbeit geführt haben. Auch Herrn Dr. Dieter Lienert möchte ich besonders erwähnen.

Nicht zuletzt möchte ich mich ganz besonders bei meiner Frau Corinna und bei meinen Eltern für ihre unermüdliche moralische Unterstützung, ihre besondere Geduld und ihr Verständnis während der Entstehung dieser Arbeit bedanken.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

1. Einführung

1.1. Elektronik im Kraftfahrzeug

Im Automobilbereich ist ein deutlicher Anstieg des Elektronikanteils bei der Wertschöpfung eines Kraftfahrzeugs zu beobachten. Viele Systeme im Automobil verzichten mehr und mehr auf mechanische Komponenten, die dafür durch elektrische bzw. elektronische Pendants ersetzt werden. So wird bei aktuellen Pkws die Kraftstoffzufuhr nicht mehr über eine mechanische Drosselklappe und einen Vergaser geregelt, vielmehr kommen dafür elektronische Sensor-, Regel- und Stellkonzepte zum Einsatz (sogenanntes „E-Gas“ bzw. ein elektronisch geregeltes Einspritzsystem wie z. B. die „Motronic“ von Bosch, vgl. [Bau99]).

Die Anzahl und die Funktionalität elektronisch geregelter Systeme im Kraftfahrzeug nehmen immer weiter zu, der Elektronikanteil im Kfz beträgt heute bis zu 35% der Wertschöpfung. Es ist zu erwarten, dass dieser in den nächsten Jahren noch ansteigen wird (nach [Gor03], [Dai02] und [BTS+02]).

Die dabei entstehenden Systemarchitekturen sind sehr komplex. So beinhaltet der Maybach aus dem Hause DaimlerChrysler mit seinen serienmäßig 76 vernetzten Steuergeräten die heute (Frühjahr 2004) komplexeste Elektronikarchitektur im Kraftfahrzeugbereich (vgl. [WBD+02]). Die Beherrschung einer solch komplexen Architektur stellt große Herausforderungen an Konzeption, Entwicklung, Integration, Test und an die Wartung eines solchen Systems.

Für diesen Trend, der sich nicht alleine auf Fahrzeuge der Oberklasse beschränkt, lassen sich vielfältige Gründe anführen. Zahlreiche Funktionen in heutigen Automobilen können ohne den Einsatz von elektronischen Systemen nicht dargestellt werden. So tragen etliche elektronische Systeme wesentlich zur Komforterhöhung bei, die durch zahlreiche, zum großen Teil eingebettete, elektronische Hilfssysteme erzielt wird.

Ein großes Einsatzgebiet von Elektronik im Automobil ist der Antriebsstrang. Die Senkung des Kraftstoffverbrauchs und auch der verbesserte Schadstoffausstoß werden zu einem großen Teil durch bessere Einspritztechnik und durch Überwachung und Regelung des Schadstoffgehalts im Abgas erreicht. Durch ausgefeiltere

Regelstrategien erreicht man einen Wirkungsgradzuwachs, man erhält mehr Leistung aus weniger Hubraum, andererseits aber auch eine Senkung der Betriebskosten des Fahrzeugs. Auch weitere, zusätzliche Funktionen im Kraftfahrzeug wie beispielsweise Navigation, Kommunikation oder „Infotainment“ sind ohne entsprechende Elektronik nicht denkbar.

In den letzten drei Jahrzehnten konnte die Zahl der Unfalltoten und der Verletzten im deutschen Straßenverkehr signifikant gesenkt werden, obwohl sich die Zahl der Kraftfahrzeuge im selben Zeitraum mehr als verdoppelt hat (siehe Tabelle 1.1 nach Daten des Statistischen Jahrbuchs [Sta03]). Diese Verringerung ist die Folge des intensiven Einsatzes verschiedenster Maßnahmen zur Verbesserung der Fahrzeugsicherheit[1], nicht zuletzt die Folge des Einsatzes entsprechender Elektronik.

Tabelle 1.1.: Unfallstatistik

Jahr:	1972[a]	2002	
Kfz[b]:	20.271.509	54.889.847	+171%
Unfälle[c]:	1.381.526	2.289.474	+66%
Verletzte:	528.527	476.413	-10%
Unfalltote:	18.811	6.842	-64%

Quelle: Statistisches Bundesamt (vgl. [Sta03])

[a] alte BRD
[b] Pkws, Lkws und Motorräder
[c] polizeilich erfasste Unfälle

Ein weiterer Einsatzbereich für Elektroniksysteme im Kraftfahrzeug ist daher im Bereich der aktiven und der passiven Sicherheit zu sehen. Unter aktiver Sicherheit versteht man Sicherheit im Sinne der Unfallvermeidung, entsprechende Fahrzeugsysteme sind beispielsweise das Antiblockiersystem ABS oder die Fahrdynamikregelung ESP. Unter passiver Sicherheit hingegen wird Sicherheit im Sinne einer Abschwächung von Unfallfolgen verstanden, ein Beispielsystem ist der Airbag (vgl. [Bau99]). Die Verbesserung der aktiven Fahrzeugsicherheit beispielsweise durch Einführung des ESP-Systems wird dadurch verdeutlicht, dass nach [Spi02] im Zeitraum von 1999 bis 2001 in Fahrzeugen der DaimlerChrysler AG die Unfallquote dieser Fahrzeuge um etwa 15% gesunken ist.

Weitere Untersuchungen zeigen jedoch, dass gerade das Potenzial von passiven Sicherheitssystemen weitgehend ausgereizt ist. Weitergehende Verbesserungen der

[1] In dieser Arbeit wird Sicherheit grundsätzlich im Sinne der englischen Übersetzung „safety“ verstanden. Für eine exakte Definition des Begriffes Sicherheit sei auf Kapitel 2.1 verwiesen.

Fahrzeugsicherheit durch passive Sicherheitssysteme sind daher nicht sehr wahrscheinlich, zukünftige Fortschritte sind überwiegend von aktiven Sicherheitssystemen zu erwarten. Abbildung 1.1 aus dem Abschlussbericht des EU-Projektes „X-by-Wire“ (siehe [X-B98] bzw. Kapitel 3.8.2.1) zeigt eine Einschätzung der Zukunftspotenziale von Sicherheitssystemen im Automobil. Das weitere Potenzial von aktiven Sicherheitssystemen ist groß, es wird erwartet, dass zusätzliche zukünftige aktive Sicherheitsfunktionen schlussendlich auf Autonomes Fahren hinführen werden.

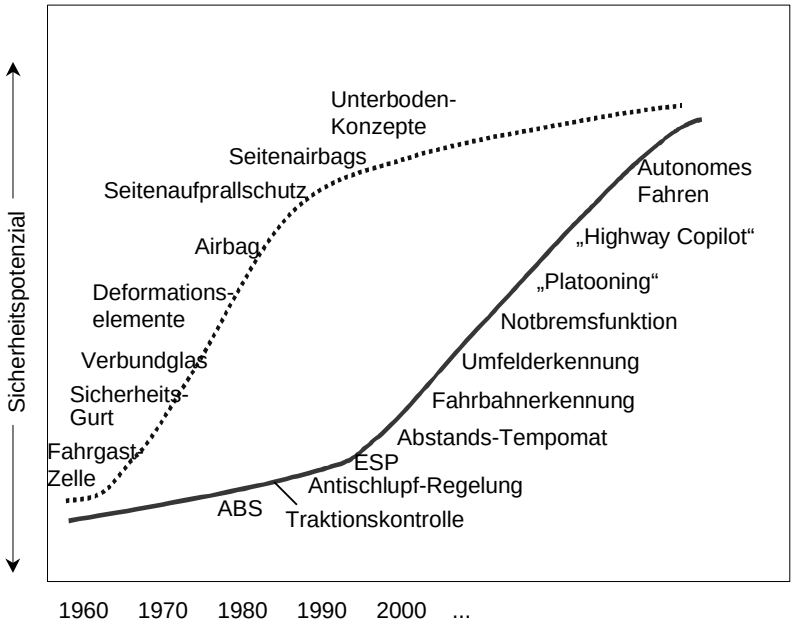

Abbildung 1.1.: Das Potenzial von aktiven bzw. passiven Sicherheitssystemen nach [X-B98].[a]

[a] Die gepunktete Linie zeigt das kumulierte Potenzial von passiven, die durchgezogene Linie das von aktiven Sicherheitssystemen.

Doch durch diese zahlreichen neuen Elektroniksysteme kommen auch neue Problemfelder ins Auto. Die Komplexität dieser Systeme macht sie nur schwer beherrschbar, durch Vernetzung kann sich ein lokales Problem an weit entfernten Systemen im Fahrzeug auswirken (einen beispielhaften Eindruck der Komplexität

einer aktuellen Bordnetzarchitektur vermittelt Abbildung 1.2). Auch das Zusammenspiel zwischen Automobilhersteller und den Zulieferern gewinnt an Komplexität.

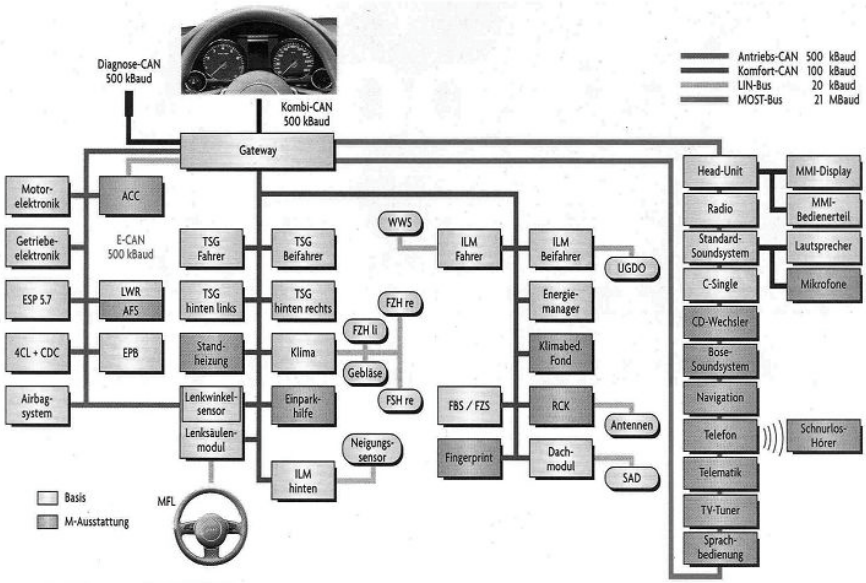

Abbildung 1.2.: Bordnetzarchitektur des Audi A8 nach [MKB03]

Immer mehr Hersteller sind aufgrund von Problemen zu Rückrufaktionen gezwungen. So mussten laut der US-Verkehrssicherheitsbehörde NHTSA[2] alleine im ersten Halbjahr 2002 in den USA 9,4 Millionen Autos aufgrund von Sicherheitsproblemen in die Werkstätten zurückgerufen werden (nach [Rei02]). Dies ist sicherlich nicht alleine ein Elektronikproblem, doch nimmt mittlerweile die Fahrzeugelektrik und -elektronik eine Spitzenstellung auch in der Ausfall-Ursachenliste der ADAC-Pannenstatistik ein (vgl. [Spi03]).

[2]NHTSA: National Highway Traffic Safety Administration

1.2. Sicherheitsrelevante Elektroniksysteme im Kfz

Für die Zukunft ist zu erwarten, dass auch bisher rein mechanische Systeme wie Lenkung und Bremse durch mechatronische, elektronisch geregelte X-by-Wire Systeme substituiert werden. Unter X-by-Wire Systemen versteht man Fahrzeugsysteme, bei denen Erfassung und Umsetzung des Fahrerwunsches hydraulisch bzw. mechanisch entkoppelt sind, die Energie- und Informationsübertragung erfolgt nur noch elektrisch (vgl. z. B. [DD02]).

Ein Serieneinsatz von trockenen X-by-Wire Systemen[3] ist in den angesprochenen Bereichen aktuell aber noch nicht geplant. Im Gegenteil, es wird eher erwartet, dass solche Systeme erst in der nächsten Dekade in Serie eingeführt werden. Hauptursache ist die ungenügende Verfügbarkeit eines entsprechenden Energiebordnetzes (nach [Nel02]).

Mit solchen Systemkonzepten wird in vielerlei Hinsicht Neuland betreten werden. Dem möglichen Sicherheitsgewinn, der durch derartige Systeme erzielt wird, steht aber auch ein erhöhtes Sicherheitsrisiko durch Ausfall oder kritisches Fehlverhalten der Elektronik gegenüber, das nicht gegen den Sicherheitsgewinn aufgerechnet werden darf. Aufgabe ist es, diese Elektroniksysteme so zu beherrschen, dass das Sicherheitsrisiko durch Ausfall oder kritisches Fehlverhalten hinreichend niedrig ist.

In solch einem Zusammenhang verwendet man den Begriff „Sicherheitsrelevantes System“, der aber oft unterschiedlich gehandhabt wird. In dieser Arbeit wird unter einem sicherheitsrelevanten System ein System verstanden, welches bei Fehlern im System oder gar bei einem Ausfall des Systems Leib und Leben der Fahrzeuginsassen unmittelbar gefährden kann.[4]

Im Luftfahrtbereich ist dagegen vergleichsweise viel Erfahrung mit sicherheitsrelevanten Elektroniksystemen vorhanden, X-by-Wire Systeme sind dort schon seit vielen Jahren im Einsatz. Daher drängt sich die Frage auf, warum man die in der Luftfahrt-Elektronik bereits seit langem bewährten Konzepte nicht direkt auf sicherheitsrelevante Elektroniksysteme im Automobilbereich übertragen kann.

Doch gerade diese beiden Domänen Automobil und Luftfahrt unterscheiden sich sehr grundsätzlich in ihren Randbedingungen. Zwar haben die Methoden zur Schaffung sicherer Elektronik im Kraftfahrzeug eine Reihe von Gemeinsamkeiten mit den im Luftfahrtbereich eingesetzten Verfahren. Doch wird den Sicherheitsanforderungen im Luftfahrtbereich mit dem Einsatz massiver Redundanz begegnet, ein

[3] Ein trockenes X-by-Wire System zeichnet sich dadurch aus, dass auch keine mechanische, hydraulische oder pneumatische Rückfallebene mehr vorhanden ist.

[4] Eine Definition der Begriffe Sicherheitsrelevanz, Fehler und Ausfall findet sich in Kapitel 2.

Vorgehen, das im Automobilbereich aufgrund der spezifischen Randbedingungen so nicht umsetzbar ist. Gerade die Forderung nach kostenoptimalen Lösungen begrenzt den Spielraum beim Einsatz von Redundanz und Diversität. Und auch Systemeigenschaften wie Modularisierbarkeit, Variantenvielfalt oder die sehr hohen Stückzahlen fordern eine andere Vorgehensweise als im Luftfahrtbereich. Außerdem sind die Sicherheitsanforderungen an Automobilsysteme v. a. aufgrund der sehr viel höheren Stückzahlen in der Regel als mindestens so kritisch zu bewerten wir die an entsprechende Systeme im Flugzeug (siehe dazu beispielsweise [Rei98] oder [Kif03]).

Das Hauptproblem ist nun nicht die Verfügbarkeit von Lösungen für sicherheitsrelevante Systeme, sondern ihre Anwendbarkeit auf die speziellen Randbedingungen der Kraftfahrzeugindustrie. Lösungen, welche komplexe mechanische Rückfallebenen verwenden, werden nicht die Kostenvorgaben erreichen. Aber ohne mechanische Rückfallebenen wird der Nachweis einer ausreichenden Zuverlässigkeit und Sicherheit dieser neuen Systeme eine große Herausforderung.

Die Folge ist, dass Lösungen für sicherheitsrelevante Elektroniksysteme nicht ohne weiteres vom Luftfahrtbereich auf den Automobilbereich übertragbar sind. Damit müssen für sicherheitsrelevante Elektroniksysteme im Automobil neue Lösungen gefunden werden.

1.3. Entwicklungsmethoden für sicherheitsrelevante Systeme

Da sicherheitsrelevante Elektroniksysteme strenge Anforderungen an Sicherheit und auch Zuverlässigkeit mit sich bringen, verlangen sie auch nach entsprechenden Methoden für die Systementwicklung und den Systementwurf.

In der Vergangenheit war im Automobilbereich die Funktion eines Systems der Haupttreiber bei der Systementwicklung. Andere nicht-funktionale Randbedingungen wurden in der Regel zusätzlich berücksichtigt. Bei sicherheitsrelevanten Systemen muss aber die Sicherheit des Fahrzeugs gleichermaßen wie die Systemfunktion berücksichtigt werden, aus diesem Grund ist eine grundlegende Anpassung des verwendeten Entwicklungsprozesses und der Entwicklungsmethodik notwendig.

Der Standardprozess bei der Entwicklung von elektronischen Systemen im Automobilbereich basiert auf dem „V-Modell ’97“, einem Lebenszyklusmodell, das der Entwicklungsstandard bei der Entwicklung von IT Systemen in der

Bundesrepublik Deutschland ist. Ursprünglich für IT Systeme gedacht, wurde es in der Zwischenzeit für andere Industriebereiche angepasst, so auch für den Automobilbereich. Jedoch ist die methodische Vorgehensweise weder formalisiert noch in der Automobilbranche standardisiert, fast jeder Hersteller oder Zulieferer verwendet einen etwas anderen Prozess.

Im Augenblick ist man in der Automobilbranche dabei, zusätzliche Erfahrung bei der Entwicklung von sicherheitsrelevanten und von hochzuverlässigen Elektroniksystemen aufzubauen.

Genauso wie im Bereich der Elektroniksysteme ist im Luftfahrtbereich auch bzgl. der eingesetzten Prozesse und Entwicklungsmethoden sehr viel Erfahrung vorhanden, denn dort gibt es eine Anzahl standardisierter Prozesse (dazu z. B. [Kne00]). Allerdings lassen sich die dort eingesetzten Prozesse - ähnlich wie die Elektroniksysteme - nicht direkt auf den Automobilbereich übertragen, da in diesen beiden Bereichen sehr unterschiedliche Randbedingungen existieren. Einige Grundkonzepte und Ideen lassen sich aber übernehmen, darauf baut diese Arbeit auf.

1.4. Motivation und Ziele der Arbeit

Bei der Systementwicklung im Automobilbereich wird Sicherheit als Systemeigenschaft heute sehr wohl schon berücksichtigt. Allerdings ist es zukünftig notwendig, dass die Sicherheitseigenschaften eines Systems schon früher bei der Systementwicklung und v. a. gleichwertig mit der Funktion des Systems betrachtet werden. Nur so ist es möglich, die zukünftig höheren Sicherheitsanforderungen an entsprechende Systeme zu erfüllen.

Der Entwicklungsstandard DIN EN 61508 (vgl. [DIN02]) aus dem Prozessautomatisierungsbereich wird aktuell sehr intensiv auch im Automobilbereich diskutiert. Im Augenblick ist es noch nicht absehbar, ob er in der Zukunft auch hier eine Rolle spielen wird, da er ursprünglich nicht für den Automobilbereich entworfen wurde.

In dieser Arbeit wird ein Ansatz für eine Entwicklungsmethodik für die Entwicklung von sicherheitsrelevanten Elektroniksystemen im Automobil vorgestellt. Die Methodik berücksichtigt gleichermaßen Sicherheit und Funktion und kann in Form eines „Zwei-V-Modells“ beschrieben werden. Sie basiert auf dem Grundprinzip des V-Modells, ein zweites V mit Elementen, die einen besonderen Schwerpunkt auf Sicherheit und Zuverlässigkeit haben, wurde dem originalen V hinzugefügt und mit ihm an mehreren Zeitpunkten im Entwicklungsprozess verknüpft. Dabei basiert

die beschriebene Vorgehensweise auf Erfahrungen aus der Luftfahrtindustrie, die auf automobile Randbedingungen übertragen wurden.

Ziel der Arbeit ist es nicht, einen umfassenden, neuen Entwicklungsprozess vorzustellen, der alle neuen Aspekte der Systementwicklung von sicherheitsrelevanten Systemen im Detail löst. Es soll vielmehr in Form eines Vorgehensmodells aufgezeigt werden, wie ein grundsätzliches Vorgehen zur Lösung des Problems aussehen kann. Der Fokus liegt dabei auf der methodischen Vorgehensweise, sehr stark prozessorientierte Schritte wie Reviews oder Entwicklungs-Dokumente wurden aus Gründen der Komplexität ausgeklammert.

1.5. Gliederung der Arbeit

Im zweiten Kapitel dieser Arbeit werden die Grundlagen, die im Zusammenhang mit der Thematik nicht zwingend vorausgesetzt werden können, kurz dargelegt und erläutert. Hier finden sich neben Begriffsdefinitionen und Erklärungen im Zusammenhang mit Sicherheit und Zuverlässigkeit auch kurze Einführungen in verwendete Methoden wie die FMEA oder die Fehlerbaumanalyse.

Im darauf folgenden Kapitel wird dann der Stand der Technik bei der Systementwicklung im Automobilbereich, im Luftfahrtbereich und in der Prozessautomatisierung dargelegt. Dabei wird untersucht, inwiefern Erfahrungen aus diesen anderen Industriebereichen für die Automobilindustrie übernommen werden können.

In Kapitel 4 wird ein Vorschlag für eine neue Entwicklungsmethodik für sicherheitsrelevante Elektroniksysteme im Automobil vorgestellt und im darauf folgenden Kapitel 5 im Detail erläutert.

Darauf folgt Kapitel 6 mit zwei Beispielen, anhand derer das konkrete Vorgehen der neuen Entwicklungsmethodik veranschaulicht wird. Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick.

2. Grundlagen

Diese Arbeit ist im Gebiet der Sicherheit und Zuverlässigkeit angesiedelt. Ein häufig auftretendes Problem in diesem Bereich ist ein nicht einheitliches oder gar falsches Verständnis dieser Begriffe Sicherheit und Zuverlässigkeit. Daher werden in diesem Kapitel die wichtigsten in dieser Arbeit verwendeten Begriffe definiert, zudem werden einige der verwendeten Methoden erklärt.

2.1. Sicherheit, Risiko und Gefährdung

Der Begriff „Sicherheit“ ist im Deutschen mehrdeutig. Im englischen Sprachraum gibt es dafür die eindeutigen Entsprechungen „security“ und „safety“, die den Sicherheitsbegriff besser differenzieren.

So wird ein System sicher im Sinne von englisch „security“ bezeichnet, wenn seine Umwelt nur einen streng kontrollierten und überwachten Zugriff auf das System hat. Damit ist also Sicherheit gegen etwas oder gegen jemanden gemeint, beispielsweise gegen ungewünschte Eindringlinge („security“ nach [Oxf89]: „... freedom or protection from danger or worry; measures taken to prevent from spying, attacks, theft“).

Umgekehrt wird ein System als sicher im Sinne des englischen Begriffs „safety“ bezeichnet, wenn von dem System keine bzw. nur eine vertretbare Gefährdung auf seine Umwelt ausgeht („safety“ bedeutet nach [Oxf89] „... being safe; not being dangerous or in danger“).

Die Sicherheit eines Systems, wie sie unter der zweiten Bedeutung („safety“) verstanden wird, ist Kernstück der Betrachtung im Rahmen dieser Arbeit. Im Folgenden wird der Begriff Sicherheit daher stets im Sinne von „safety“ verwandt.

2.1.1. Sicherheit

Nach DIN VDE 31000 (vgl. [DIN87]) und DIN V 19250 (vgl. [DIN95]) ist Sicherheit (engl. safety) wie folgt definiert:

Definition 2.1 (Sicherheit)
Sicherheit ist eine Sachlage, bei der das Risiko nicht größer als das Grenzrisiko ist.

Die hier verwendete Definition von Sicherheit basiert auf den Begriffen Risiko und Grenzrisiko. Daher ist zum Verständnis von Definition 2.1 zunächst eine Festlegung des Risikobegriffs notwendig.

2.1.2. Risiko

In DIN VDE 31000 und DIN V 19250 findet sich folgende Definition für das Risiko (engl. risk):

Definition 2.2 (Risiko)
Das Risiko, das mit einem bestimmten technischen Vorgang oder Zustand verbunden ist, wird zusammenfassend durch eine Wahrscheinlichkeitsaussage beschrieben, die

- *die zu erwartende Wahrscheinlichkeit bzw. Häufigkeit*[1] *des Eintritts eines zum Schaden führenden Ereignisses und*
- *das beim Ereigniseintritt zu erwartende Schadensausmaß*

berücksichtigt.

Mathematisch gesehen ist das Risiko R ein sogenanntes Paar aus Ereignishäufigkeit H und Schadensausmaß S, kurz

$$R = (H, S). \tag{2.1}$$

Im Allgemeinen ist ein Risiko aber nicht quantitativ erfassbar, nur selten lässt es sich als Kombination der beiden Parameter Häufigkeit (H) und Schadensausmaß (S) quantifizieren. Daher sind zwei Risiken auch nicht direkt miteinander vergleichbar. Eine Ausnahme ist, wenn das Schadensausmaß S bei zwei Risiken identisch ist, dann kann ein mathematischer Vergleich von H zum Ziel führen. Dies muss bei der

[1] Anstelle von Wahrscheinlichkeiten wird in der Technik meist mit Häufigkeiten gerechnet.

Definition des Sicherheitsbegriffs beachtet werden, der auf einer Vergleichbarkeit von Risiken beruht.

Ist das Schadensausmaß S quantifizierbar (z. B. in finanzieller Form) und kann man im konkreten Fall von einem „mittleren Schaden“ bei Eintritt des Ereignisses sprechen, so kann für die Quantifizierung des Risikos das Produkt $R = H \cdot S$ verwendet werden.

In DIN V 19250 wird die Häufigkeit H eines gefährlichen Ereignisses weiter in 3 Einflussgrößen aufgegliedert. Dies sind die Aufenthaltsdauer im Gefahrenbereich (A), die Möglichkeit der Gefahrenabwendung (G) und die Wahrscheinlichkeit des Eintritts des unerwünschten Ereignisses (W).

2.1.3. Grenzrisiko

Basierend auf Definition 2.2 kann man nun auch das Grenzrisiko (engl. limiting risk) definieren:

Definition 2.3 (Grenzrisiko)
Das Grenzrisiko, auch vertretbares Risiko genannt, ist das größte noch vertretbare Risiko eines bestimmten technischen Vorgangs oder Zustandes.

Im Allgemeinen lässt sich auch das Grenzrisiko nicht quantitativ erfassen, es wird daher in der Regel durch sicherheitstechnische Festlegungen beschrieben.

Sicherheitstechnische Festlegungen sind Angaben über technische Werte und Maßnahmen sowie Verhaltensanweisungen, deren Einhaltung im Rahmen des jeweiligen technischen Konzeptes sicherstellen sollen, dass das Grenzrisiko nicht überschritten wird. Sie werden sowohl durch Gesetze, Rechtsverordnungen oder sonstige staatliche Maßnahmen erlassen, als auch in Übereinstimmung mit der unter Fachleuten vorherrschenden Meinung getroffen, z. B. durch technische Regelwerke.

Das Grenzrisiko wird durch subjektive und objektive Einflüsse bestimmt und ist für verschiedene Anwendungen teilweise sehr unterschiedlich ausgeprägt. Besondere Einflüsse sind beispielsweise persönliche Gefahrenempfindungen, gesellschaftliche Akzeptanz von Gefahren (sogenannter „sozialer Konsens“) oder der betroffene Personenkreis.

2.1.4. Gefahr

Komplementär zur Definition von Sicherheit ergibt sich die Definition von Gefahr (engl. danger):

Definition 2.4 (Gefahr)
Gefahr ist eine Sachlage, bei der das Risiko größer als das Grenzrisiko ist.

2.1.5. Das Prinzip der Risikominderung

Ist das aktuelle Risiko einer betrachteten Funktion oder eines betrachteten Systems größer als das Grenzrisiko, so muss das Risiko durch geeignete Maßnahmen bis auf mindestens das Grenzrisiko vermindert werden (vgl. Abbildung 2.1). Diese Vorgehensweise der Risikominderung ist die Standardvorgehensweise bei der Entwicklung von sicherheitsrelevanten Systemen in der Prozessautomatisierung (vgl. Kapitel 3.5).

In der Praxis ist das Feststellen des aktuellen Risikos aus den in Kapitel 2.1.2 beschriebenen Gründen nicht trivial. Man behilft sich daher mit Methoden wie beispielsweise einem Vergleich mit dem Stand der Technik, mit einer Kategorisierung unter Verwendung eines Risikographen oder mit der Anwendung der sogenannten ALARP[2]-Methode (siehe dazu z. B. [DIN02]).

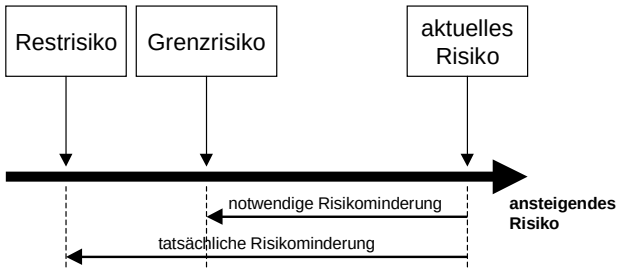

Abbildung 2.1.: Prinzip der Risikominderung (nach [DIN02])

Die Risikominderung wird in der Regel durch sogenannte Sicherheitssysteme mit Hilfe von Sicherheitsfunktionen erbracht. Sie kann aber auch durch Design-

[2] ALARP: as low as reasonably practicable = so niedrig wie vernünftigerweise möglich.
ALARP wird beispielsweise in Großbritannien bei der Festlegung von Regelungen, die die Arbeitssicherheit betreffen, von der zuständigen Behörde Health & Safety Executive HSE verwendet (siehe [Sto96], [HSE04]).

Änderungen im System erfolgen, so dass das System von vornherein ein geringeres Risiko darstellt. Meist werden die notwendigen Maßnahmen durch sicherheitstechnische Festlegungen in den entsprechenden Entwicklungsstandards beschrieben.

Geringe Risiken sind aber auch bei bestehender Sicherheit nie vollständig auszuschließen. Eine „absolute Sicherheit“ ohne jegliches Risiko gibt es weder in der Technik noch in der Natur.

2.1.6. Gefährdung

Ein anderer Begriff im Zusammenhang von Sicherheit und Risiko ist die Gefährdung (engl. hazard, manchmal auch „failure condition“ genannt).

Nach DIN EN 61508 (vgl. [DIN02]) ist sie wie folgt definiert:

Definition 2.5 (Gefährdung)
Eine Gefährdung ist eine potenzielle Schadensquelle.

Damit ist eine Gefährdung eine Situation, von der eine tatsächliche oder mögliche Gefahr für Personen oder die Umwelt des Systems ausgeht.

2.1.7. Sicherheitsrelevanz

In dieser Arbeit wird anstelle des Begriffs „sicherheitskritisch“ durchgängig der Begriff „sicherheitsrelevant“ verwendet. Da der Begriff „sicherheitskritisch“, der in seiner Bedeutung „sicherheitsrelevant“ sehr ähnlich ist, das Vorhandensein eines Sicherheitsproblems impliziert (was aber nicht zwingenderweise im konkreten System der Fall sein muss), wird in dieser Arbeit der Begriff „sicherheitsrelevant“ bevorzugt.

Ein sicherheitsrelevantes System (engl. safety-relevant system) im Automobilbereich wird in dieser Arbeit wie folgt definiert:

Definition 2.6 (sicherheitsrelevantes System)
Ein System ist sicherheitsrelevant, wenn eine Fehlfunktion im System im Regelfall zu einer unmittelbaren Gefahr für Leib und Leben von Verkehrsteilnehmern führen kann. Die Fahrsituation ist durch die Fahrzeuginsassen nicht beherrschbar oder beeinflussbar.

Häufig findet man auch den Begriff „sicherheitsbezogenes System“, der manchmal synonym zu „sicherheitsrelevantes System“ gebraucht wird. Diese Begriffe sind aber grundsätzlich verschieden.

Basierend auf DIN EN 61508 ist ein sicherheitsbezogenes System (engl. safety-related system) wie folgt definiert:

Definition 2.7 (sicherheitsbezogenes System)
Ein sicherheitsbezogenes System ist ein System, das zusammen mit externen Einrichtungen zur Risikominderung dazu vorgesehen ist, die notwendige Risikominderung auf das Restrisiko zu erreichen.

2.1.8. Probleme beim Umgang mit dem Sicherheitsbegriff

Systeme werden somit als sicher bezeichnet, wenn das Risiko, durch den Umgang mit diesen Systemen verletzt oder getötet zu werden, von der Gesellschaft und dem Gesetzgeber akzeptiert wird. Die Definition des Sicherheitsbegriffs zeigt, dass es eine „absolute Sicherheit“ nicht gibt. Restrisiken existieren, insbesondere durch Umgebungseinflüsse und Einbindung des Menschen in Entwicklungsprozesse, als Verkehrsteilnehmer und Nutzer der Technologie.

Das Problem bei der Sicherheitsdefinition nach Definition 2.1 liegt in der Anwendung des Risikobegriffs. Meist werden daher basierend auf der Annahme eines bestimmten Grenzrisikos unter Anwendung der Risikominderung sogenannte Sicherheitsspezifikationen erstellt, deren Erfüllung ein sicheres System garantieren sollen. In besonderen Fällen kann auch die technische Zuverlässigkeit (vgl. Kapitel 2.2) bzgl. der sicherheitsbezogenen Korrektheit eines Systems zur Überprüfung der Sicherheitsanforderungen herangezogen werden.

2.1.9. Wie sicher ist „sicher genug“?

Die Hauptfrage bei der Konzeption von sicherheitsrelevanten Systemen ist, wann das System sicher genug, d. h. wann das Risiko geringer als das Grenzrisiko ist. Diese Frage, wie viele und welche Maßnahmen zur Risikominderung notwendig sind, lässt sich oft nicht eindeutig beantworten.

Zur Lösung dieses Problems sind in der Technik unterschiedliche Verfahren entstanden, die sich in qualitative und quantitative Verfahren einteilen lassen.

2.1.9.1. Qualitative Verfahren

Die Ermittlung des aktuellen Risikos eines Systems erfolgt sehr häufig unter Verwendung eines sogenannten Risikographen, der auf der Anwendung von Risikofaktoren basiert.

Da die exakte Quantifizierung von Risiken oft nicht möglich, auf jeden Fall aber sehr aufwändig und schwierig ist, werden zur Vereinfachung Einflussgrößen eingeführt, die es ermöglichen, Art und Höhe der Gefahrensituation beim Versagen zu beschreiben.

Für die Klassifizierung der Gefährdungen existieren in der Literatur mehrere Vorgehensweisen, die bekannteste und verbreitetste ist die Vorgehensweise nach DIN V 19250 „Grundlegende Sicherheitsbetrachtungen für MSR-Schutzeinrichtungen“ (vgl. [DIN95]). In der aktuell auch im Automobilbereich diskutierten Norm DIN EN 61508 (vgl. [DIN02]) ist diese Vorgehensweise unverbindlich als sogenannter „informativer Anhang“ übernommen worden.

Aus einer Vielzahl möglicher Parameter, die einen Einfluss auf Sicherheitsanforderungen und auf Maßnahmen haben, werden 4 wichtige (Risiko-)Parameter verwendet, die eine sinnvolle Risikoabstufung gestatten und die wesentlichsten Beurteilungsaspekte beinhalten.

Die Risikodefinition folgt der Definition nach Formel 2.1 auf Seite 10, wobei die dort erwähnte Aufgliederung der Häufigkeit H in die Aufenthaltsdauer A, die Gefahrenabwendung G und die Wahrscheinlichkeit des unerwünschten Ereignisses W zur Anwendung kommt.

Dabei wird bei dem Parameter Schadensausmaß die Art des zu schützenden Rechtsguts, die Höhe des Schadens und die Verletzungsschwere von Personen berücksichtigt. In den Parameter Gefahrenabwendung gehen Kriterien wie die Betriebsart eines Prozesses, die zeitliche Entstehung und Entwicklung der Gefahr, mögliches Erkennen der Gefahr, mögliche Abwendung der Gefahr und praktische Sicherheitserfahrung ein (vgl. [DIN95]). Es ergibt sich der Risikograph nach Abbildung 2.2 auf der nächsten Seite.

In DIN V 19250 entsprechen die Zahlen 1 bis 8 den sogenannten Anforderungsklassen AK 1 bis AK 8, denen entsprechende Maßnahmen für Sicherheitsfunktionen zugeordnet sind. Je höher die Ordnungszahl einer Anforderungsklasse, um so größer ist das von einem Sicherheitssystem zu beherrschende Teilrisiko und um so höher sind in der Regel die Anforderungen und die daraus resultierenden Maßnahmen. Systeme in AK 8 können in der Regel durch einfache Sicherheitssysteme nicht beherrscht werden, das aktuelle Risiko ist zu groß.

Den Anforderungsklassen sind in entsprechenden Normen Maßnahmen hinterlegt, die bei der Entwicklung des Systems bzw. beim System selbst berücksichtigt werden müssen.

In der Norm DIN EN 61508, die die DIN V 19250 ablöst, werden die Anforderungsklassen AK 1 bis AK 8 vier sogenannten Sicherheitsintegritätsleveln (SIL) von SIL 1 bis SIL 4 zugeordnet (siehe dazu auch Kapitel 3.5.1).

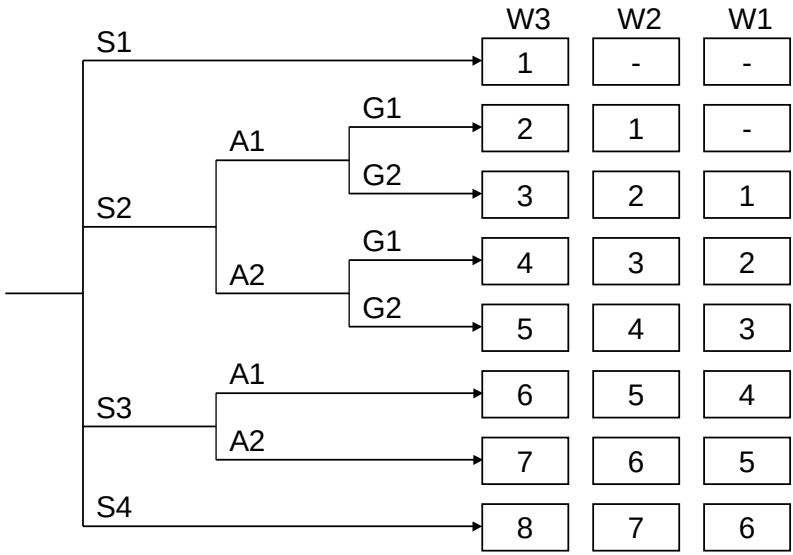

Abbildung 2.2.: Risikograph nach DIN V 19250

Legende:

Schadensausmaß	S1	leichte Verletzung einer Person; kleinere schädliche Umwelteinflüssse
	S2	Schwere irreversible Verletzung einer oder mehrerer Personen oder Tod einer Person; vorübergehende größere schädliche Umwelteinflüsse
	S3	Tod mehrerer Personen; langandauernde größere schädliche Umwelteinflüsse
	S4	Katastrophale Auswirkungen, sehr viele Tote
Aufenthaltsdauer	A1	selten bis öfter
	A2	häufig bis dauernd
Gefahrenabwendung	G1	möglich unter bestimmten Bedingungen
	G2	kaum möglich
Eintritts-wahrscheinlichkeit	W1	sehr gering
	W2	gering
	W3	relativ hoch

Da der Risikograph nach DIN V 19250 und die verwendeten Parameter explizit für Automatisierungssysteme ausgelegt sind, ist er für Automobilsysteme nur bedingt geeignet.

Eine den Randbedingungen der Automobilbranche angepasste Einstufungsmethodik ist „CARTRONIC Safety Analysis“ nach [BDM99]. Interessant ist der an CARTRONIC angelehnte Prozess zur Sicherheitsanalyse und die dabei verwendeten Sicherheitsklassen.

Für unterschiedliche Bereiche der Technik haben sich jeweils spezielle Risikographen mit entsprechenden Kategorisierungen herausgebildet. Tabelle 2.1 zeigt eine Aufstellung der gängigsten Standards.

Tabelle 2.1.: Normen zur Einstufung von Systemen in Sicherheitsklassen

Vorgehensweise	Anwendungsbereich
DIN V 19250	Prozessautomatisierung
DIN V VDE 0801	Rechner im Sicherheitskontext
EN 954	Prozessautomatisierung
MIL-STD 882D	Militärtechnik
JAR 25.1309	Luftfahrt
DIN EN 61508	Prozessautomatisierung
nach CARTRONIC	Automobilbereich

Den Risikoklassen können zusätzlich noch Ausfallwahrscheinlichkeiten zugeordnet werden, der Übergang zu einer quantitativen Analysemethode ist dann fließend.

2.1.9.2. Quantitative Verfahren

Eine andere Vorgehensweise ist ein quantitativer Vergleich mit dem Stand der Technik. Dafür wird die Wahrscheinlichkeit für ein kritisches Ereignis im Zusammenhang mit einem technischen System ermittelt und das Ergebnis dann mit einem akzeptierten Risiko verglichen.

Dies ist nicht immer einfach, da das Grenzrisiko für eine neue Technologie von der Gesellschaft u. U. anders bewertet wird als das für eine eingeführte Technologie. Es ist daher davon auszugehen, dass ein System, das in einer neuen Technologie dargestellt wird, höchstens ein vergleichbares Risiko haben darf, wie ein bestehendes System in einer verwendeten Technologie.

Da die auftretenden Wahrscheinlichkeiten meist sehr kleine Zahlen sind, werden sie nicht in % angegeben, sondern in *ppm*, Ausfallraten auch in *fit*[3].

Zur Bestimmung des Grenzrisikos werden entsprechende Statistiken herangezogen. Tabelle 2.2 nach [KBC+03] zeigt eine Unfallstatistik, die Todesfälle im Verkehr in der EU im Zeitraum 2001/2002 nach Verkehrsmitteln aufschlüsselt. Die Tabelle gibt so einen Eindruck dafür, welches Risiko heutzutage im Verkehr allgemein akzeptiert ist.

Tabelle 2.2.: Unfallstatistik für die EU 2001/2002

Tote pro 100 Mio. Personenkilometer		Tote pro 100 Mio. Personenstunden	
Motorrad	13,8	Motorrad	440
zu Fuß	6,4	Fahrrad	75
Fahrrad	5,4	Straße (insgesamt)	28
Straße (insgesamt)	0,95	zu Fuß	25
Pkw	0,7	Pkw	25
Fähre	0,25	zivile Luftfahrt	16
Bus	0,07	Fähre	8
zivile Luftfahrt	0,035	Bus	2
Eisenbahn	0,035	Eisenbahn	2

Quelle: European Transport Safety Council ETSC (vgl. [KBC+03])

In dieser Arbeit werden für Berechnungen 10000 Betriebsstunden als Lebensdauer eines Pkws zu Grunde gelegt. Daraus folgt unter der Annahme einer durchschnittlichen Lebensdauer eines Automobils von 15 Jahren eine jährliche Betriebsdauer von etwa 667 Betriebsstunden. Wird im Folgenden von Jahren bzw. Stunden gesprochen, so werden darunter in der Regel Betriebsjahre bzw. Betriebsstunden eines Automobils verstanden.

2.1.9.3. Beispiel für eine quantitative Abschätzung

In Tabelle 2.3 auf der nächsten Seite sind Verkehrsunfälle mit Bezug auf technische Mängel an Bremse bzw. Lenkung bei Pkws in Deutschland aus den Jahren 2000 und 2001 aufgeführt. Zur weiteren Betrachtung wurden diejenigen Unfälle addiert, bei denen es zu sicherheitsrelevanten Auswirkungen, also zu Gefährdungen von Leib und Leben der Fahrzeuginsassen kam (vgl. Definition von Sicherheitsrelevanz auf

[3] *ppm* = „parts per million“, d. h. 10^{-6}
fit = „failure in time“, entspricht $10^{-9}/h$

Seite 13), wovon bei Unfällen mit Getöteten, Personenschaden sowie bei schwerem Sachschaden ausgegangen werden kann. Für die Berechnung der jeweiligen Wahrscheinlichkeiten wurde zugrunde gelegt, dass es im Jahre 2000 in Deutschland 42,8 Mio. Pkw und in 2001 43,8 Mio. Pkw gab.

Tabelle 2.3.: Ursachen für Straßenverkehrsunfälle

	Bremse		Lenkung	
Unfälle mit	2000	2001	2000	2001
Getöteten	4	1	2	1
Personenschaden	233	180	85	99
Schwerem Sachschaden	98	87	70	58
Summe	335	268	157	158
rel. in *ppm*	7,8	6,1	3,7	3,6

Quelle: Statistisches Bundesamt (vgl. [Sta01], [Sta02])

Nach Tabelle 2.3 ergeben sich in Summe die auf das jeweilige Jahr bezogenen Wahrscheinlichkeiten P_{2000} bzw. P_{2001} für Unfälle aufgrund von technischen Mängeln an Bremse bzw. Lenkung (ohne die Berücksichtigung von möglichen Mehrfachnennungen) zu

$$P_{2000} = 7,8\ ppm + 3,7\ ppm = 11,5\ ppm \tag{2.2}$$

und

$$P_{2001} = 6,1\ ppm + 3,6\ ppm = 9,7\ ppm. \tag{2.3}$$

Damit ergibt sich in erster Näherung ein Durchschnittswert P_{Mittel} für die Wahrscheinlichkeit für einen Unfall aufgrund technischer Mängel an Bremse oder Lenkung in einem dieser Jahre von

$$P_{Mittel} \approx 10\ ppm. \tag{2.4}$$

Unter der erfahrungsgemäßen Annahme, dass nur jeder fünfte auftretende Fehler auch wirklich zu einem Unfall führt, ergibt sich die auf ein Jahr bezogene Auftretenswahrscheinlichkeit P_{Fehler} für einen kritischen Fehler im Gesamtfahrzeug aufgrund eines Fehlers an Bremse oder Lenkung, der zum Unfall führen kann, zu

$$P_{Fehler} \approx 5 \cdot P_{Mittel} = 50\ ppm. \tag{2.5}$$

Die jährliche Betriebsdauer eines Kraftfahrzeugs beträgt im Durchschnitt in etwa $t_{Betrieb} = 667h$. Daraus folgt, dass unter Verwendung von Formel 2.5 die Fehlerrate für einen kritischen Fehler im Gesamtfahrzeug, der zum Unfall führen kann,

$$\lambda_{krit} = \frac{P_{Fehler}}{t_{Betrieb}} = \frac{50\ ppm}{667\ h} \approx 0,1\ ppm/h \tag{2.6}$$

beträgt. Diese Fehlerrate ist die Summe der Fehlerraten für kritische Fehler in den beiden Systemen Bremse und Lenkung.

Bei nur zwei solchen Systemen pro Fahrzeug (Stand heute) liegen die beiden Einzelfehlerraten in der gleichen Größenordnung wie die Gesamtfehlerrate, d. h. die zulässige Fehlerrate $\lambda_{zul,System}$ für einen kritischen Fehler in einem der beiden Systeme Bremse und Lenkung ist jeweils

$$\lambda_{zul,System} \approx 0,1\ ppm/h = 10^{-7}/h. \tag{2.7}$$

Bei einer Annahme von zukünftig ca. $n_{System} = 10$ sicherheitsrelevanten Systemen pro Fahrzeug folgt für die geforderte Fehlerrate für einen sicherheitskritischen Fehler $\lambda_{Fehler,System}$ in einem System

$$\lambda_{Fehler,System} = \frac{\lambda_{zul,System}}{n_{System}} = 0,01\ ppm/h = 10^{-8}/h. \tag{2.8}$$

Zur Abschätzung des Grenzrisikos einer mit einem kritischen Fehler eines sicherheitsrelevanten Elektroniksystems im Automobil verbundenen Gefährdung kann der Wert $\lambda_{Fehler,System} = 10^{-8}/h$ verwendet werden.

2.1.10. Sicherheit in Sinne von „security“

Anders als bei der Betrachtung der Fahrzeugsicherheit im Sinne von „safety“, die im Prinzip schon seit der Erfindung des Automobils im Jahr 1886 ein Thema ist[4], treten Security-Probleme in der Regel erst durch neue Elektroniksysteme im Fahrzeug auf. Systemkonzepte, wie die zur Zeit immer wichtiger werdende Telematik, erfordern entsprechende Sicherheitsmaßnahmen, um fahrlässige oder böswillige Manipulationen an diesen Systemen zu verhindern. Auch bei der Motorsteuerung sind bedingt durch aufwändige Elektroniksysteme Security-Maßnahmen notwendig geworden, die unter dem Namen „Tuningschutz“ zusammengefasst werden.

[4] beispielsweise musste in den Anfangsjahren des Automobils jedem Fahrzeug aus Sicherheitsgründen eine Person mit einer Fahne vorausgehen

Der weiter zunehmenden Einsatz von Elektronik für neue Systeme im Fahrzeug, besonders für Systeme, die für die Vernetzung im Fahrzeug, aber auch für die teilweise autonome Kommunikation des Fahrzeugs mit seiner Umwelt oder mit anderen Fahrzeugen zuständig sind, wird hier weitere und auch neue Probleme aufwerfen.

Es ist daher auch eine besondere Berücksichtigung der „security“ solcher Systeme notwendig, was sich auch in einem entsprechenden Entwicklungsprozess niederschlagen muss.

Dafür werden aber prinzipiell andere Vorgehensweisen, Werkzeuge und Systemkonzepte verlangt, als diejenigen, die Inhalt dieser Arbeit sind.

2.2. Zuverlässigkeit und Verfügbarkeit

Weitere Kenngrößen eines Systems sind Zuverlässigkeit und Verfügbarkeit, die im Nachfolgenden definiert werden.

2.2.1. Zuverlässigkeit

Die Zuverlässigkeit (engl. reliability) eines Systems ist nach Birolini (vgl. [Bir97]) wie folgt definiert:

Definition 2.8 (Zuverlässigkeit)
Zuverlässigkeit ist die Fähigkeit eines Systems, während einer vorgegebenen Zeitdauer bei zulässigen Betriebsbedingungen ein funktionsgerechtes Verhalten zu erbringen (d. h. korrekt zu arbeiten).

Voraussetzung ist, dass das System zu Anwendungsbeginn korrekt war und nur Ausfälle zu Inkorrektheit führen können.

Mathematisch ist die Zuverlässigkeit eines Systems die Wahrscheinlichkeit dafür, dass das System die geforderte Funktion unter vorgegebenen Arbeitsbedingungen während einer festgelegten Zeitdauer ausfallfrei ausführt.

Das Auftreten eines Ausfalls kann mit der Ausfallwahrscheinlichkeit $F(t)$ beschrieben werden. Für die Ausfalldichte $f(t)$ gilt

$$f(t) = \frac{dF(t)}{dt}. \tag{2.9}$$

Das Komplement zur Ausfallwahrscheinlichkeit $F(t)$ ist die wohl wichtigste Kenngröße der Zuverlässigkeit, die Überlebenswahrscheinlichkeit $R(t)$. Es gilt

$$R(t) = 1 - F(t). \tag{2.10}$$

Die Überlebenswahrscheinlichkeit $R(t)$ gibt die Wahrscheinlichkeit dafür an, ein System nach einer Zeit t in funktionsfähigem Zustand anzutreffen, vorausgesetzt das System war zum Startzeitpunkt t_0 korrekt.

Allgemein lautet die Formel für stetige Überlebenswahrscheinlichkeiten $R(t)$

$$R(t) = e^{-\int_{t_0}^{t} \lambda(\xi)\, d\xi}. \tag{2.11}$$

Die Variable λ wird Ausfallrate genannt. Die Ausfallrate beschreibt die Anzahl der Ausfälle in einem Zeitintervall dt, bezogen auf die Anzahl der zum Zeitpunkt t noch intakten Einheiten. Sie ist allgemein definiert als das Verhältnis von Ausfalldichte zu Überlebenswahrscheinlichkeit:

$$\lambda(t) = \frac{f(t)}{R(t)}. \tag{2.12}$$

Die meistverwendete Verteilungsfunktion für die Überlebenswahrscheinlichkeit $R(t)$ von mechanischen oder elektronischen Komponenten, Geräten und Systemen ist die Weibullverteilung. Die 3-parametrige Weibullverteilung hat die Dichtefunktion

$$f(t) = \frac{b}{T - T_0} \left(\frac{t - T_0}{T - T_0} \right)^{b-1} \cdot e^{-\left(\frac{t - T_0}{T - T_0} \right)^b}. \tag{2.13}$$

Dabei ist b die sogenannte „Ausfallsteilheit“, T die charakteristische Lebensdauer und T_0 die ausfallfreie Zeit.

Gilt $T_0 = 0$, so spricht man von einer 2-parametrigen Weibull-Verteilung. Es gilt dann für die Überlebenswahrscheinlichkeit

$$R(t) = e^{-(t/T)^b}. \tag{2.14}$$

Zur Beschreibung von Frühausfällen verwendet man $b < 1$, mit $b > 1$ beschreibt man Verschleißverhalten. Für $b = 1$ reduziert sich die Weibullverteilung auf die Exponentialverteilung. Es gilt dann für konstantes $\lambda = 1/T$

$$R(t) = e^{-\lambda \cdot t}, \tag{2.15}$$

eine häufig verwendete erste Näherung für $\lambda \cdot t \ll 1$ ist

$$R(t) \approx 1 - \lambda \cdot t. \tag{2.16}$$

Charakteristisch für die Exponentialverteilung ist, dass ein neues Objekt das gleiche zukünftige Ausfallverhalten hat wie ein altes Objekt, von dem man weiß, dass es gerade noch lebt. Die Exponentialverteilung ist daher für die meisten elektrischen Komponenten gültig. Abbildung 2.3 nach [VDA00] zeigt den typischen Verlauf der Ausfallrate λ über der Lebenszeit für elektrische Komponenten. Mit der sogenannten „Badewannenkurve“ versucht man der Modellierung von Frühausfällen sowie von Verschleiß- und Ermüdungsausfällen Rechnung zu tragen.

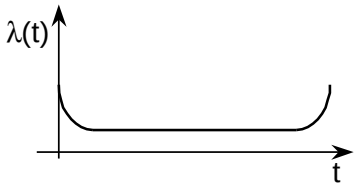

Abbildung 2.3.: Typischer Verlauf der Ausfallrate λ für elektrische Komponenten (nach [Sto96])

Eine weitere Kenngröße für die Zuverlässigkeit ist die mittlere Lebensdauer einer Einheit, meist als MTTF („Mean Time To Failure“) bezeichnet. Die MTTF ist der Erwartungswert $E(\tau)$ der Lebensdauer τ, den man aus dem Integral

$$MTTF = E(\tau) = \int_0^\infty t \cdot f(t)\,dt = \int_0^\infty R(t)\,dt \tag{2.17}$$

erhält. Für eine Exponentialverteilung mit konstanter Ausfallrate λ gilt dann

$$MTTF = \int_0^\infty e^{-\lambda \cdot t}\,dt = \frac{1}{\lambda} = T. \tag{2.18}$$

Nach Ablauf der MTTF gilt bei konstantem λ für eine exponentialverteilte Überlebenswahrscheinlichkeit immer

$$R(t = MTTF) = \frac{1}{e} \approx 36,7\%. \tag{2.19}$$

Bei konstantem λ kann man noch die mittlere Ausfallzeit MTBF („Mean Time Between Failures“) definieren. Wenn in einem gegebenen Zeitraum Δt n von N Komponenten ausfallen, so gilt

$$MTBF = \frac{\Delta t \cdot N}{n}. \tag{2.20}$$

Kann das System repariert werden, so kann man eine mittlere Reparaturzeit MTTR („Mean Time To Repair“) definieren. Damit gilt für die MTBF:

$$MTBF = MTTF + MTTR \tag{2.21}$$

2.2.2. Verfügbarkeit

Verfügbarkeit ist nach Storey (vgl. [Sto96]) wie folgt definiert:

Definition 2.9 (Verfügbarkeit)
Die Verfügbarkeit eines Systems ist die Wahrscheinlichkeit für eine korrekte Funktion im Sinne der Zuverlässigkeit mit Berücksichtigung von Reparatur.

Bei einer ausreichend großen Datenmenge und einem exponentialverteilten Ausfall ergibt sich die mittlere bzw. stationäre Verfügbarkeit zu

$$V_{mittel} = \frac{MTTF}{MTBF} = \frac{MTTF}{MTTF + MTTR}. \tag{2.22}$$

Die momentane oder Punkt-Verfügbarkeit ist wie folgt definiert:

Definition 2.10 (Momentane Verfügbarkeit)
Die momentane Verfügbarkeit ist die Wahrscheinlichkeit, eine Einheit zu einem vorgegebenen Zeitpunkt t der geforderten Anwendungsdauer unter vorgegebenen Arbeitsbedingungen in einem funktionsfähigen Zustand anzutreffen.

Sind die Komponenten eines Systems unabhängig, so lässt sich die Systemverfügbarkeit mit Hilfe des sogenannten Booleschen Modells berechnen (siehe dazu [Str83]).

2.2.3. Bestimmung der Ausfallrate λ

Eine a priori Bestimmung der Ausfallrate λ - z. B. für die Verwendung in einem Fehlerbaum (vgl. Kapitel 2.4.2) - ist erstrebenswert, aber schwierig.

Theoretisch sollte gerade die Automobilindustrie aufgrund der Millionen Fahrzeuge im Feld umfangreiche und genaue Daten über unterschiedlichste Arten von Fehlern und Ausfällen auch von Elektronikkomponenten besitzen. Tatsächlich ist es aber so, dass die Daten, die Herstellern und Zulieferern zur Verfügung stehen, nur sehr eingeschränkt für eine quantitative Analyse verwendbar sind.

Dies liegt darin begründet, dass bei weitem nicht alle Fehler im Feld an den Hersteller bzw. Zulieferer zurückgemeldet werden. Nicht alle Defekte und Ausfälle werden überhaupt in Werkstätten identifiziert, dies gilt insbesondere für ältere Fahrzeuge, deren Garantiezeit abgelaufen ist und die nicht mehr regelmäßig gewartet werden. Selbst wenn Fehler in der Werkstatt exakt festgestellt werden, so werden diese oft weder dem Hersteller noch dem Zulieferer gemeldet. Dies liegt daran, dass Werkstatt, Hersteller und Zulieferer getrennte Organisationen sind; Fehler werden in der Regel nur im Rahmen von Garantiefällen oder von sicherheitsrelevanten Problemen, die zu Rückrufaktionen führen, gemeldet. Innerhalb des Garantiezeitraums kommen zwar Konzepte zum Einsatz, die die Übermittlung von Informationen über Fehler an den Hersteller bzw. Zulieferer ermöglichen, doch sind diese auf die Betrachtung des wirtschaftlichen Aspekts von Garantiekosten optimiert. Technische Details werden in der Regel nicht in einer solchen Qualität bereitgestellt, wie sie für eine quantitative Zuverlässigkeitsanalyse notwendig wären (siehe dazu auch [WJR04]).

Abhilfe schaffen umfangreiche Datenbanken, in denen Ausfallraten von mechanischen, elektrischen oder elektronischen Systemkomponenten aufgeführt sind. Diese basieren auf einer modellbasierten, analytischen Bewertung. Einige allgemein verwendete Standards zur Zuverlässigkeitsvorhersage elektromechanischer und elektronischer Bauteile sind in Tabelle 2.4 auf der nächsten Seite aufgeführt.

Auch wenn der v. a. in der Luftfahrtindustrie recht intensiv eingesetzte Standard MIL-HDBK-217 (vgl. [Dep95]) von Tag zu Tag mehr veraltet, bleibt er die am meisten verwendete Methode für elektronische Komponenten. Neue Methoden, wie sie z. B. vom PRISM-Werkzeug des Reliability Analysis Center (vgl. [RAC04]) verwendet werden, ermöglichen eine verbesserte Modellierung. Doch muss dieses Werkzeug noch um einige Bauteilkategorien erweitert und mit der Industrie weiterentwickelt werden, bevor es eine breite Akzeptanz finden wird. Das neueste Mitglied in der Familie der Zuverlässigkeitsvorhersagemodelle ist RDF 2000 (vgl.

Tabelle 2.4.: Methoden zur Ermittlung der Ausfallrate λ

Standard	Beschreibung
MIL-HDBK-217	Das MIL-Handbook-217 ist die Grundlage der Zuverlässigkeitsvorhersage seit etwa 40 Jahren, wird aber seit 1995 nicht mehr aktualisiert. Es bleibt trotzdem die am meisten verwendete Vorhersage sowohl im zivilen als auch im militärischen Bereich v. a. der Luftfahrt. Das Handbuch beinhaltet eine Reihe von empirisch entwickelten Ausfallraten-Modellen, die auf historischen Bauelemente-Teilausfallraten für sehr viele Bauteiltypen basieren.
Telcordia SR-332	Telcordia SR-332 ist dem MIL-HDBK-217 sehr ähnlich, basiert aber in erster Linie auf Fernmeldedaten.
RDF 2000	RDF 2000 ist ein vergleichsweise neuer Zuverlässigkeitsvorhersage-Standards, der die meisten der MIL-HDBK-217-Bestandteile umfasst. Die Modelle beinhalten zusätzlich die Auswirkung elektrischer Belastungen wie z. B. einen sich periodisch wiederholenden Temperaturverlauf. Dieser Standard wird wohl für viele Anwendungsbereiche der internationale Nachfolger des MIL-HDBK-217 werden.
PRISM	PRISM ist eine neue Vorhersagemethode aus dem Jahr 2000, basierend auf den Datenbanken des „Reliability Analysis Center“ RAC des amerikanischen Verteidigungsministeriums DoD. Eine Besonderheit ist die Möglichkeit, Vorhersagen basierend auf Testdaten zu aktualisieren. Im Augenblick wird es noch durch die Einbindung nur weniger Bauteilfamilien beschränkt, aber es hat Potenzial für eine allgemeine Akzeptanz, wenn die Anzahl der Bauteile ansteigt.
PoF	Physik der Ausfälle (PoF) versucht, das schwächste Glied eines Designs zu identifizieren, um sicherzustellen, dass die geforderte Geräte-Lebensdauer durch den Entwurf überschritten wird. Die Methodik ignoriert im Allgemeinen die Ausfallursachen, die während der Herstellung auftreten, und nimmt an, dass die Produkt-Zuverlässigkeit ausschließlich durch die vorausgesagte Lebensdauer der schwächsten Verbindung bestimmt wird. Diese Familie von Vorhersagen unterscheidet sich wesentlich von den anderen aufgeführten empirischen Methodiken und wird in erster Linie in der Bauelemente-Vorstufe während der Entwicklungsphase verwendet.
IEEE STD 493	Das sogenannte „IEEE Gold Book“ liefert Daten bzgl. der Systemzuverlässigkeit in industriellen und kommerziellen Netzverteilungssystemen. Das Handbuch wurde 1997 aktualisiert; jedoch sind die neuesten Zuverlässigkeitsangaben im Dokument von 1989 und älter.
IEC 61709	Die IEC-Vorhersage ist wenig verbreitet, weil die Umweltbedingungen, unter denen ein System eingesetzt wird, bei der Berechnung nach IEC keine Berücksichtigung finden. Nur die elektrischen Belastungen, wie Strom und Spannung, gehen in die Berechnung ein. Aus diesem Grund wird die Berechnung nach diesem IEC-Standard 61709 international nur als Bauteilzählmethode („parts count method“) eingestuft und hat außerhalb des deutschsprachigen Raums fast keine Bedeutung.

Quelle: RAMS Consult (vgl. [RAM03]) und [SS01]

[Uni00]), welches das Potenzial für den Standard von morgen hat. Ein Beispiel für die Anwendung von RDF 2000 im Automobilbereich findet sich in [SKR+03].

Die Tatsache, dass Ausfalldaten elektronischer Systeme und Komponenten in entsprechendem Umfang und Qualität im Automobilbereich zur Zeit nicht zur Verfügung stehen, ist jedoch für die Entwicklung zukünftiger Systeme nicht akzeptabel.

Sollen quantitative Zuverlässigkeitsanalysen für die Bewertung elektronischer Systeme umfassend zum Einsatz kommen, ist daher der Aufbau und die Pflege automobilspezifischer Datenbanken gerade auch auf Basis von Felddaten unumgänglich. Dazu muss eine entsprechende funktionierende Infrastruktur geschaffen werden.

Dabei sollte man die generelle Art des Umgangs mit solchen Daten überdenken, auch was die Vertraulichkeit der Daten betrifft. Da durch solche Systeme Menschenleben gefährdet werden können, sollten diese Daten dann prinzipiell auch der Gesellschaft zur Verfügung stehen.

Davon unabhängig ist das Problem zu lösen, wie die Sicherheitseigenschaften eines sicherheitsrelevanten Systems, die bei der Inbetriebnahme nachgewiesen wurden, auch über die Systemlebensdauer gewährleistet werden können. Ob dies beispielsweise durch regelmäßige intensive Überprüfung der Systemeigenschaften z. B. im Rahmen der Hauptuntersuchung oder gar durch präventiven Tausch entsprechender Komponenten geschehen kann (wie er im Luftfahrtbereich üblich ist), soll hier nicht weiter vertieft werden.

2.2.4. Sicherheit vs. Verfügbarkeit?

In der Vergangenheit gab es eine prinzipielle Unterscheidung zwischen sicheren Systemen und zuverlässigen Systemen. Mittlerweile ist es aber in vielen Bereichen so, dass diese strikte Trennung nicht mehr möglich ist (vgl. [KG02b]).

Früher wurden in beiden Domänen schon ähnliche Konzepte verwendet (z. B. Redundanz), der Unterschied bestand aber im Wesentlichen darin, dass das Hauptaugenmerk bei der Konzeption von sicheren Systemen darauf lag, dass garantiert keine Beeinträchtigung der Umwelt erfolgen konnte. Dies erreichte man durch Maßnahmen zur Fehlervermeidung, -erkennung und -beherrschung. Im Zweifel wurde bei Feststellung von Inkonsistenzen konsequent abgeschaltet.

Anders bei Systemen mit hohen Anforderungen an Zuverlässigkeit und Verfügbarkeit. Priorität war hier, dass das System die gewünschte Funktion erbrachte. Dafür wurden auch Einschränkungen in Kauf genommen, wie z. B. kurzzeitig fehlerhafte Signale oder Timing-Probleme.

Diese beiden Eigenschaften werden sich bei zukünftigen Elektroniksystemen im Automobil vermischen. Systeme wie Lenkung und Bremse haben höchste Anforderungen an die Systemsicherheit. Diese Sicherheitseigenschaft ist zugleich die Anforderung an die Zuverlässigkeit des Systems, da ein Ausfall eines Lenk- oder Bremssystems ein kritischer Zustand für das Gesamtfahrzeug darstellt. Das System darf prinzipiell nie ausfallen; und wenn Teile des Systems dies doch tun, so muss dies auf eine Art und Weise geschehen, dass die Umwelt des Systems nicht gefährdet wird.

Der Hauptgrund für diese Systemeigenschaft ist die Tatsache, dass für diese Systeme ein allgemein gültiger sicherer Systemzustand nicht für alle Fahrsituationen festgelegt werden kann. Daher hat das betrachtete System gleichzeitig vergleichbare Sicherheits- wie Zuverlässigkeitsanforderungen, d. h. das Fahrzeug ist nur dann sicher, wenn z. B. das Lenksystem hinreichend zuverlässig ist.

Als Folge können für Systeme mit dieser besonderen Eigenschaft im selben Kontext Sicherheitsanforderungen in Zuverlässigkeitsanforderungen überführt werden. Der Sicherheitsprozess in der Luftfahrtindustrie basiert auf dieser Tatsache (vgl. [Vah98]). Sicherheit ist nur schwer quantitativ erfassbar, quantitative Zuverlässigkeitsanalysen sind hingehen Stand der Technik.

2.3. Fehler und Fehlertoleranz

Die englische Sprache hat für den deutschen Begriff Fehler je nach konkreter Bedeutung unterschiedliche Entsprechungen. In dieser Arbeit wird eine Konvention nach Tabelle 2.5 auf der nächsten Seite für den Fehlerbegriff verwendet, die an [Lap92] angelehnt ist (siehe auch Abbildung 2.4).

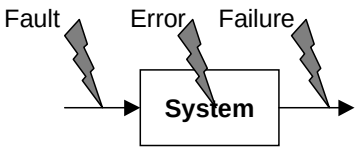

Abbildung 2.4.: Die verschiedenen Arten von Fehlern

Tabelle 2.5.: Konventionen für den Fehlerbegriff

Engl. Begriff	Deutsch. Begriff	Bedeutung
Fault	Fehlerursache	Auslöser für einen Fehlerzustand.
Error	Fehlerzustand	Systemzustand, der dafür verantwortlich ist, dass eine Fehlerauswirkung bzw. ein Ausfall auftritt. Die Fehlerursache wird im System offenbar.
Failure	Fehlerauswirkung oder Ausfall	Abweichung der erbrachten Leistung von der in der System-Spezifikation geforderten Leistung. Ein Ausfall einer Komponente kann die Fehlerursache eines übergeordneten Systems sein.

nach Laprie [Lap92]

Ein weiterer Begriff in diesem Zusammenhang ist Fehlertoleranz.

Definition 2.11 (Fehlertoleranz)
Unter Fehlertoleranz wird die Fähigkeit eines Systems verstanden, auch mit einer begrenzten Zahl fehlerhafter Subsysteme seine spezifizierte Funktion erfüllen zu können (nach [Gör89]).

Ein System wird „fail-operational“ (oft FO abgekürzt) genannt, wenn im Fall eines Fehlers das System eine Grundfunktionalität solange aufrecht erhält, bis ein sicherer Systemzustand erreicht ist. Diese Systemeigenschaft wird verlangt, wenn im Fall eines Fehlers ein solcher sicherer Systemzustand nicht unmittelbar erreicht werden kann (nach [Kop97]).

Als „fail-safe“ bezeichnet man dagegen die Fähigkeit eines technischen Systems, beim Auftreten bestimmter Ausfälle im sicheren Zustand zu bleiben oder unmittelbar in einen (anderen) sicheren Zustand überzugehen (vgl. [VDI00]).

Ein System nennt man „fail-silent“ (kurz FS), wenn es sich nach Feststellen von Abweichungen vom vorgegeben Verlauf nach außen hin sofort still verhält, d. h. sich in einen für die Außenwelt als unkritisch angesehen Zustand versetzt. Dies kann beispielsweise durch spontanes Selbstabschalten der entsprechenden Komponenten oder des jeweiligen Teilsystems erfolgen.

2.4. Methoden

2.4.1. FMEA und FMECA

Die „Failure Modes and Effects Analysis“ FMEA[5] wurde Mitte der sechziger Jahre in den USA von der NASA für das Apollo-Projekt entwickelt. Sie ist eine strukturierte, qualitative Analyse eines Systems, Subsystems oder einer Funktion mit dem Ziel, potenzielle Fehlermöglichkeiten, ihre Ursachen und die möglichen Folgen auf das System zu analysieren. Wenn, was meist der Fall ist, die FMEA um eine Bewertung der Kritikalität erweitert wird, also um eine Bewertung der Schwere einer Fehlerauswirkung und um die Auftretenswahrscheinlichkeit eines Fehlers, dann wird sie auch „Failure Modes, Effects and Criticality Analysis“, kurz FMECA[6] genannt (nach [BEE+96] und [BB94]).

Die FMEA-Methode hat sich auch in der Automobilindustrie bewährt und ist mittlerweile ein fester Baustein in der Qualitätssicherung geworden.

Zur Durchführung der Analyse wird das System in Einheiten aufgeteilt. Jede Einheit besteht aus einer Anzahl von Komponenten, deren Ausfallverhalten bekannt ist bzw. vorhergesagt werden kann. Jedes Ausfallverhalten einer Komponente wird isoliert betrachtet, die Auswirkung auf das gesamte System analysiert und das Fehlerverhalten der betrachteten Komponente quantitativ bewertet. Dabei können aber nur Auswirkungen von Einzelfehlern untersucht werden.

Dies geschieht in der Regel unter Anwendung eines sogenannten FMEA-Formblatts, wie es in Abbildung 2.5 auf der nächsten Seite dargestellt ist. Wichtigste Variable ist die sogenannte Risikoprioritätszahl RPZ, die als Produkt $RPZ = B \cdot A \cdot E$ aus Bedeutung eines Fehlers B, seiner Auftretenswahrscheinlichkeit A und seiner Entdeckungswahrscheinlichkeit E definiert ist. Jeder dieser Parameter wird typischerweise durch einen Wert zwischen 1 und 10 charakterisiert, dabei ist 10 der kritischste Wert. Somit kann die Risikoprioritätszahl zwischen 1 und 1000 schwanken. Werte für RPZ ab 100 werden in der Regel als kritisch angesehen, eine Änderung im System oder eine Gegenmaßnahme wird dann notwendig.

Historisch gesehen gliedert sich die FMEA in 3 Arten: die beiden älteren Methoden Konstruktions-FMEA und Prozess-FMEA und die neuere System-FMEA.

Bei der Konstruktions-FMEA wurde eine Fehlerbetrachtung auf Bauteilebene durchgeführt, der funktionale Zusammenhang sämtlicher Bauteile zueinander wurde dabei nicht systematisch betrachtet. Bei der Prozess-FMEA hingegen wurden mögliche Fehler in einzelnen Prozessschritten betrachtet.

[5] deutsch: Fehler-Möglichkeits- und Einfluss-Analyse
[6] deutsch: Fehler-Möglichkeits-, Einfluss- und Kritikalitäts-Analyse

BOSCH Qualitätssicherung		FMEA Erzeugnis: Stellglied Sach-Nummer: 9 319 150 342								Seite Abt. FMEA-Nr. Datum		10 FVB 75 1289940001 10.10.88
Nummer	Komponente Prozeß	Funktion Zweck	Fehler-art	Fehler-auswirkung	Fehler-ursachen	Fehler-vermeidung	Fehler-entdeckung	S S	- A	E E	S*E RZ	Maßnahme V: / T:
1110	Büchsen-halter montieren	Teile für Lötprozess vorbereiten	Beschädigung der Dichtflächen	Stellglied undicht nach außen => Benzindämpfe im Motorraum	Späne und Flitter in Montage-vorrichtung	Waschen vor der Montage, regelmäßige Reinigung der Werk-zeuge	100%-Sicht-prüfung der Lötung; Oberflächen-prüfung; 100%-Sicht-prüfung vor Verpacken	10	2	1	20	
1180	Büchsen-halter löten	Teile zusammen halten	Teil nicht gelötet	Stellglied undicht nach außen => Benzindämpfe im Motorraum	Lot fehlt	Abfrage Lotvorschub	100%-Sicht-prüfung der Lötung; 100%-Sicht-prüfung der Oberfläche; 100%-Dicht-heitsprüfung	10 (10	2 2	2 1	40 20)	100%-Dicht-heitsprüfung der Bau-gruppe Büchsen-halter V: FVB2 T: 01.89
		Dichtheit gewähr-leisten	Teil undicht (Lunker)	Stellglied undicht nach außen => Benzindämpfe im Motorraum	Zu wenig Lot	Abfrage Lotvorschub	100%-Sicht-prüfung der Lötung	10 (10	4 3	6 2	240 60)	100%-Dicht-heitsprüfung V: FVB2 T: 01.89 + Konstruktive Verbesserung Lötstelle V: EVA3 T: 03.89

S = Schwere des Fehlers A = Auftretenswahrscheinlichkeit E = Entdeckungswahrscheinlichkeit Risikozahl RZ = S x A x E
V = Verantwortlichkeit T = Einführungstermin (C) 1987, 1996 Robert Bosch GmbH Stuttgart

Abbildung 2.5.: Beispiel für eine FMEA. Quelle: [Bau99]

Eine systematische und gegliederte Analyse des gesamten Herstellungsprozesses wurde jedoch nicht durchgeführt. Die Erstellung einer FMEA erfolgte ausschließlich über das FMEA-Formblatt und ließ damit keine strukturierte Beschreibung von Funktions- und möglichen Fehlfunktionszusammenhängen in Systemen zu. Dies machte eine Weiterentwicklung der Methode in Richtung System-FMEA erforderlich.

Die System-FMEA setzt in einem sehr frühen Stadium des Produktentstehungsprozesses an und untersucht den Entwicklungs- und Planungsstand auf mögliche Fehler, um damit vorbeugende Maßnahmen zu deren Vermeidung einzuleiten.

2.4.2. Fehlerbaumanalyse

Die Fehlerbaumanalyse, kurz FTA (nach engl. Fault Tree Analysis), ist eine graphische Methode zur Analyse der Zuverlässigkeit eines Systems.

Ausgehend von einem Systemereignis, dem sogenannten „Top Ereignis“, wird eine Ursachenanalyse entlang einer Baumstruktur durchgeführt. Kombinationen von Ursachen werden durch logische Operatoren (UND, ODER, etc.) beschrieben, die Vorgehensweise wird durch standardisierte Symbole unterstützt.

Die entstehende Baumstruktur, der sogenannte Fehlerbaum, wird immer weiter detailliert, bis an den Ästen des Baumes die sogenannten Elementarereignisse stehen. Werden diesen Elementarereignissen Auftretenswahrscheinlichkeiten hinterlegt, so lässt sich unter Anwendung der Booleschen Algebra die Auftretenswahrscheinlichkeit für das Top Ereignis angeben. Ein Beispiel für einen Fehlerbaum zeigt Abbildung 2.6 auf der nächsten Seite.

Falls eine Fehlerbaumanalyse mangels entsprechender Daten nicht quantitativ durchgeführt werden kann, so kann sie immer noch durch Berechnen der Minimalschnitte des Baumes (d. h. der disjunktiven Normalform) ein wertvolles Werkzeug bei der Systemanalyse sein.

Eine verwandte Analysemethode ist die Ereignisbaumanalyse, kurz ETA (engl. Event Tree Analysis). Hier wird umgekehrt vorgegangen: Ausgehend von Elementarereignissen werden mögliche Auswirkungen ermittelt, der Fehlerbaum wird quasi von unten aufgebaut.

2.4.3. Markov-Analyse

Die Markov-Analyse ist ein mathematisches Vorgehen zur Ermittlung der Wahrscheinlichkeit für einen bestimmten Systemzustand, mit dem dynamische Vorgänge und reparierbare Systeme darstellbar sind.

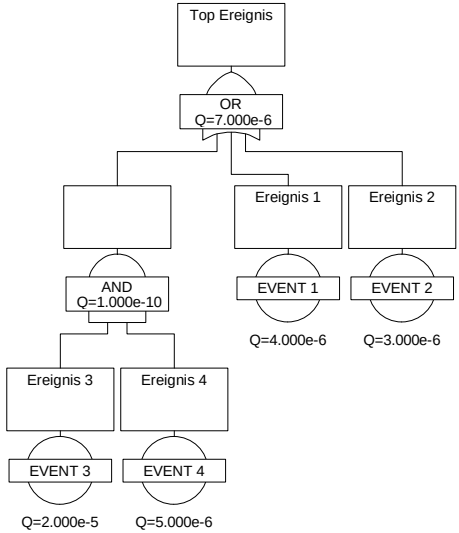

Abbildung 2.6.: Beispiel für einen Fehlerbaum

Die Markov-Analyse basiert auf dem gewöhnlichen Markov-Prozess, einem stochastischen Prozess $Z(t)$ mit endlich vielen Zuständen $Z_1, Z_2, \ldots, Z_n$, für den für jeden beliebigen Zeitpunkt t seine weitere Entwicklung nur vom gegenwärtigen Zustand und von der Zeit t abhängig (d. h. unabhängig von der Vorgeschichte) ist. (vgl. [Fri01])

Im einfachsten Fall werden bei der Markov-Analyse nur Systeme behandelt, deren Elemente konstante Ausfall- und Reparaturraten haben (wie z. B. bei exponentialverteilten Prozessen).

Die konstanten Übergangsraten vom Zustand Z_i in den Zustand Z_j werden mit α_{ij} bezeichnet. Die Wahrscheinlichkeit, zur Zeit t im Zustand Z_i zu sein, wird durch die Zustandswahrscheinlichkeit $P_i(t)$ beschrieben. Für ein System mit m Komponenten ergeben sich im Allgemeinen $n = 2^m$ Zustände und demzufolge n Differentialgleichungen der Form

$$\frac{dP_i(t)}{dt} = -\sum_{j=1, j\neq i}^{n} \alpha_{ij} P_i(t) + \sum_{j=1, j\neq i}^{n} \alpha_{ij} P_j(t) \quad \forall \quad i = 1(1)n \tag{2.23}$$

mit der Normierungsbedingung

$$\sum_{i=1}^{n} P_i(t) = 1. \tag{2.24}$$

Gleichung 2.23 lässt sich einfacher in Matrizenschreibweise darstellen:

$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{A} \cdot \mathbf{P}(t). \tag{2.25}$$

Die Matrix $\mathbf{A}$ besteht aus den Übergangsraten α_{ij} zwischen den einzelnen Zuständen, wobei die Summe einer Spalte der Matrix immer gleich 0 sein muss. Es gilt also:

$$\mathbf{A} = \begin{pmatrix} \beta_1 & \alpha_{21} & \alpha_{31} & \cdots & \alpha_{n1} \\ \alpha_{12} & \beta_2 & \alpha_{32} & \cdots & \alpha_{n2} \\ \alpha_{13} & \alpha_{23} & \beta_3 & \cdots & \alpha_{n3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{1n} & \alpha_{2n} & \alpha_{3n} & \cdots & \beta_n \end{pmatrix} \tag{2.26}$$

mit der Bedingung

$$\beta_i = -\sum_{k=1, k \neq i}^{n} \alpha_{ik}. \tag{2.27}$$

Nun sei Γ_S die Menge aller Zustände, in der die Betrachtungseinheit funktionstüchtig ist. Die Verfügbarkeit ist dann gleich der Summe der Zustandswahrscheinlichkeiten aller Zustände, die einen aktiven Betriebszustand des Systems bzw. dessen Komponenten darstellen:

$$V(t) = \sum_{Z_i \varepsilon \Gamma_S} P_i(t). \tag{2.28}$$

Die Markov-Gleichungen sind in ihren Fundamentalformen nicht zur Berücksichtigung von zeitabhängigen Übergangsraten α_{ij} verwendbar. Dies bedeutet, dass beispielsweise Alterung und Ermüdung von Komponenten nicht mit dem Markov-Prozess modelliert werden können, da diese Effekte zeitabhängige Übergangsraten erfordern.

Liegt im System keine Exponentialverteilung vor bzw. sind die Übergangsraten zeitbehaftet, so können der Semi-Markov-Prozess bzw. die Systemtransporttheorie zum Einsatz kommen, die aber mathematisch sehr viel komplexer sind. Daher sei hier auf entsprechende Literatur verwiesen (z. B. [BGH+86], [Bis90], [Bir97] und [Fri01]).

2.4.3.1. Beispiel

Als Beispiel für eine Markov-Analyse soll das System aus Abbildung 2.7 dienen.

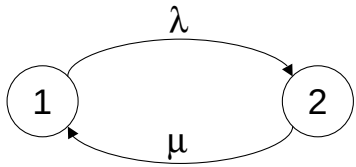

Abbildung 2.7.: Beispielsystem für die Markov-Analyse

Zustand 1 sei der funktionsfähige Zustand der Systems, Zustand 2 das ausgefallene System. λ ist die Ausfallrate des Systems, μ seine Reparaturrate.

Es ergibt sich nach Formel 2.26 und 2.27 folgende Matrix **A**:

$$\mathbf{A} = \begin{pmatrix} -\lambda & \mu \\ \lambda & -\mu \end{pmatrix} \tag{2.29}$$

Diese entspricht nach Formel 2.23 folgenden Differentialgleichungen:

$$\dot{P}_1(t) = -\lambda \cdot P_1(t) + \mu \cdot P_2(t) \tag{2.30}$$

$$\dot{P}_2(t) = \lambda \cdot P_1(t) - \mu \cdot P_2(t) \tag{2.31}$$

Die Lösung dieses linearen Diffentialgleichungssystems unter der Anfangswertbedingung $P_1(0) = 1$ und $P_2(0) = 0$ (d. h. das System ist zu Beginn im funktionsfähigen Zustand) ergibt für $P_1(t)$ und $P_2(t)$:

$$P_1(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} \cdot e^{-(\mu+\lambda)\cdot t} \tag{2.32}$$

$$P_2(t) = \frac{\lambda}{\lambda + \mu} - \frac{\lambda}{\lambda + \mu} \cdot e^{-(\mu+\lambda)\cdot t} \tag{2.33}$$

Somit können die Wahrscheinlichkeiten $P_1(t)$ und $P_2(t)$ dafür, dass sich das System in Zustand 1 bzw. Zustand 2 befindet, für jeden Zeitpunkt t explizit angegeben werden. Komplexere Systeme mit mehr Zuständen lassen sich analog analysieren, die Lösung des entstehenden Differentialgleichungssystems wird in der Regel komplexer und muss u. U. numerisch erfolgen.

Für einen stabilen Systemzustand, also für $t \to \infty$, ergibt sich im konkreten Fall aus Formel 2.32:

$$P_1(t \to \infty) = \frac{\mu}{\lambda + \mu} \tag{2.34}$$

$$P_2(t \to \infty) = \frac{\lambda}{\lambda + \mu} \tag{2.35}$$

$P_1(t \to \infty)$ ist die mittlere Verfügbarkeit des Systems, denn

$$P_1(t \to \infty) = \frac{\mu}{\lambda + \mu} \tag{2.36}$$

$$= \frac{MTTF}{MTTF + MTTR} \tag{2.37}$$

was der Definition der mittleren Verfügbarkeit V_{mittel} aus Formel 2.22 entspricht.

2.4.4. HAZOP-Analyse

Für die Untersuchung von technischen Systemen mit verfahrenstechnischen bzw. allgemein kontinuierlichen Prozessen v. a. in der chemischen Industrie wird häufig das HAZOP-Verfahren (engl. HAZard and OPerability studies) eingesetzt. In Deutschland wird das Verfahren auch PAAG[7]-Verfahren genannt (vgl. [BHR90], [Sto96], [Bie03]).

Hierbei werden von einer Expertengruppe mit Hilfe von Leitwörtern (z. B. „was wäre, wenn...“) systematisch mögliche Abweichungen vom Sollverhalten ermittelt und analysiert. Die Stärken liegen im Aufzeigen von Auswirkungen von Parameteränderungen auf die Sicherheit der betrachteten Anlage. Das Verfahren wird allerdings als sehr zeitaufwändig und kostenintensiv eingeschätzt. Die Qualität der Ergebnisse ist stark vom Brainstorming und somit vom speziellen Fachwissen der Expertengruppe abhängig.

[7] PAAG: **P**rognose von Störungen, **A**uffinden der Ursachen, **A**bschätzen der Wirkungen, **G**egenmaßnahmen

3. Stand der Technik

In diesem Kapitel wird der Stand der Technik von Entwicklungsmethoden für elektronische Systeme dargestellt. Besonderes Augenmerk liegt dabei auf der Berücksichtigung von sicherheitsrelevanten Systemeigenschaften.

Unter einer Entwicklungsmethodik versteht man eine systematische, in der Art des Vorgehens festgelegte Arbeitsweise zur Entwicklung eines Systems. Im Englischen wird dafür der Begriff „design methodology“ verwendet. Die Wortwahl „Design-Methodik“ im Deutschen ist kritisch, da man unter „Design“ in der Regel das Layout von Hardware oder auch die Gestaltgebung eines Gebrauchsgegenstandes versteht. Oft ist auch mit Design nur eine Skizze eines späteren Systems gemeint, eine Entwicklungsmethodik hat aber die Entwicklung eines konkreten Systems zum Ziel.

Eine Methodik wird meist in einer abstrakten Art beschrieben und beinhaltet eine Anzahl von Schritten, die im Laufe der Systementwicklung durchgeführt werden sollen. Die Begriffe Methode und Methodik werden dabei meist synonym gebraucht. Wenn eine Unterscheidung gemacht wird, dann wird unter einer Methodik meist eine Reihe einzelner Methoden verstanden.

Schwieriger ist die sprachliche Abgrenzung der Begriffe Entwicklungsmethodik und Entwicklungsprozess. Meist werden die beiden Begriffe fast synonym gebraucht. Wenn unterschieden wird, so wird unter einem Prozess die natürliche Vergrößerung einer Methodik, das übergeordnete Ganze verstanden. Eine Entwicklungsmethodik beschreibt die konkrete Vorgehensweise zur Entwicklung eines Systems. Ein Entwicklungsprozess beinhaltet darüber hinaus noch organisatorische und Projektmanagement-spezifische Richtlinien, wie z. B. Personal, Anlagen, Einrichtungen, Geldmittel, Techniken und Dokumentation.

3.1. Vorgehensmodelle

Für die Beschreibung einer systematischen Vorgehensweise bei der Systementwicklung verwendet man sogenannte Vorgehensmodelle. So können bestimmte

Eigenarten, Vor- und Nachteile bestehender Prozesse beschrieben werden, aber auch neue Konzepte für Vorgehensweisen sind so darstellbar.

Vorgehensmodelle können nur Teilaspekte des Vorgehens bei der Entwicklung darstellen, hier unterliegen sie allen Einschränkungen von Modellen; sie stellen aber meist ein recht gutes Abbild der Wirklichkeit dar, wobei Details aber oft außen vor bleiben.

Im Folgenden werden die wichtigsten Vorgehensmodelle für die Systementwicklung elektronischer Systeme beschrieben.

3.1.1. Das Wasserfallmodell

Eine der ersten bekannten Strukturierungen des Entwicklungsprozesses wurde von Royce vorgenommen (siehe [Roy70]) und wurde als Wasserfallmodell bekannt (siehe Abbildung 3.1). Das Wasserfallmodell ist einfach zu realisieren, recht einfach in seiner Struktur und unmittelbar auf die Zeitachse abbildbar.

Ein Problem war allerdings, dass in der Darstellung von Royce spätere Änderungen im System nicht berücksichtigt werden konnten. Boehm erweiterte daher das Modell (vgl. [BEP+82]) und fügte explizite Schritte zur Überprüfung der Ergebnisse bei Abschluss des Übergangs in den nächsten Schritt ein. Damit können später gefundene Fehler iterativ zurückverfolgt und an der richtigen Stelle korrigiert werden. Das Wasserfallmodell von Boehm ist das wahrscheinlich am häufigsten verwendete Vorgehensmodell (nach [Chr92], [VDI93]).

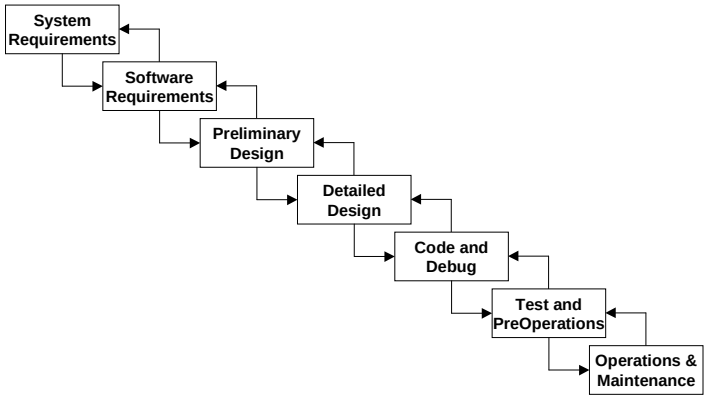

Abbildung 3.1.: Prinzipdarstellung des Wasserfallmodells

Der große Nachteil des Boehmschen Wasserfallmodells ist das Fehlen des Tests. Es ist nicht festgelegt, in welchem Testschritt gegen welche Testreferenz geprüft wird. Symmetrische Vorgehensmodelle (wie z. B. das V-Modell, vgl. Kapitel 3.1.3) dagegen beschreiben Testschritte explizit und heben damit diesen Nachteil des Wasserfallmodells auf.

3.1.2. Das Spiralmodell

Das Spiralmodell wurde ebenfalls von Boehm eingeführt (vgl. [Boe87]), mit dem Ziel, das Wasserfallmodell zu verbessern. Er beschreibt ein iteratives Vorgehen bei der Entwicklung mit mehreren Prototypenphasen, das durch die Darstellung als Spirale verdeutlicht wird (siehe Abbildung 3.2). Jeder Durchlauf der Spirale entspricht dabei der Entstehung eines Zwischenprodukts (Lastenheft, Pflichtenheft, Systementwurf, Funktionsprototyp, Endprodukt, etc.). Eine Besonderheit ist dabei die besondere Berücksichtigung des Risikos.

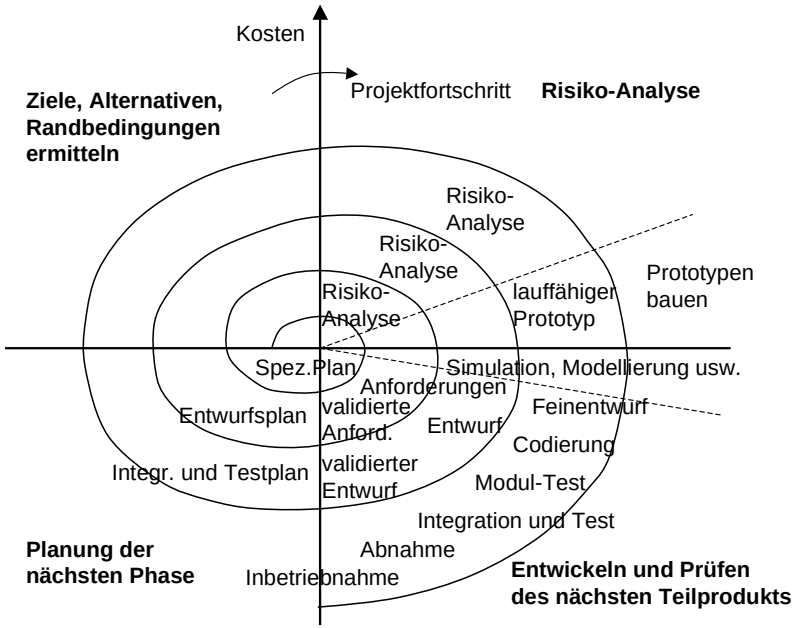

Abbildung 3.2.: Prinzipdarstellung des Spiralmodells

In den vier Quadranten des Modells sind für jedes Zwischenprodukt bestimmte Tätigkeiten vorgesehen. Im nordwestlichen Quadranten sind es die Klärung von Zielen, Anforderungen und Randbedingungen. Im nordöstlichen Quadranten stehen die Abschätzung des jeweiligen Risikos sowie die Untersuchung von Alternativen. Die eigentliche Systementwicklung wie auch Auswertung, Validierung und Verifikation findet sich im südöstlichen Quadranten. Im südwestlichen Quadranten stehen die Planungstätigkeiten für den nächsten Zyklus.

Der Einsatz des Spiralmodells hat im Vergleich zu anderen Vorgehensmodellen zwei wesentliche Vorteile: In jedem Entwicklungszyklus kann bei Bedarf der Projektplan neu angepasst werden, sowohl an den Projektfortschritt, als auch an die Entwicklung der Technologie und des Marktes. Auch das Risikomanagement wird so wesentlich verbessert. Im Spiralmodell ist es beispielsweise erlaubt, bei zeitlichen Verzögerungen bei der Entwicklung ein lauffähiges System mit verringertem Funktionsumfang zum vereinbarten Termin zu liefern. Dies ist beim Wasserfall- oder auch beim V-Modell nicht möglich.

Ein Nachteil gerade im Vergleich zum V-Modell ist, dass im Spiralmodell keine eindeutigen Zwischenergebnisse definiert werden. Dadurch eignet sich das Spiralmodell weniger gut als Vorgehensmodell für Projekte, an denen verschiedene Organisationen beteiligt sind oder für sehr große Entwicklungsprojekte.

Ein Beispiel für die Verwendung des Spiralmodells für sicherheitsrelevante Systeme im Automobil findet sich in [Hen96].

3.1.3. Das V-Modell

Das V-Modell ist ein symmetrisches Vorgehensmodell, welches sich mittlerweile zum Standard-Vorgehensmodell in der Softwareentwicklung entwickelt hat. Auch in anderen Industriebereichen wie dem Automobilsektor ist es in weiten Bereichen etabliert.

Das V-Modell ist eine Art „zusammengeklapptes Wasserfallmodell“, wobei es als Neuerung den jeweiligen Entwicklungsschritten entsprechende Testschritte gegenüberstellt. Dabei macht die Verbindung zwischen dem Entwicklungs- und dem Testschritt auf der gleichen Ebene die Referenz deutlich, gegen die geprüft wird.

Dieser Prüfprozess, der durch das V-Modell dargestellt wird, ist ein Bottom-Up-Prozess. Das heißt, bevor ein System oder Subsystem getestet werden kann, müssen alle Komponentenbeschreibungen erstellt und implementiert worden sein.

Damit fehlt aber im V-Modell die Möglichkeit, eine Anwendung frühzeitig zu validieren, bevor Ergebnisse aus nachfolgenden Entwicklungsschritten vorliegen. Deswegen erhält der Benutzer erst dann eine wirkliche Vorstellung vom Produkt,

wenn das System fast fertig ist. Daher gibt es Ansätze, durch den Einsatz von sogenannten „schnellen Prototypen“ („Rapid Prototyping“, siehe dazu das VP-Modell in Kapitel 3.1.4) diesem Problem Abhilfe zu schaffen.

Abbildung 3.3 nach [BTS+02] zeigt eine vereinfachte, prinzipielle Darstellung des V-Modells.

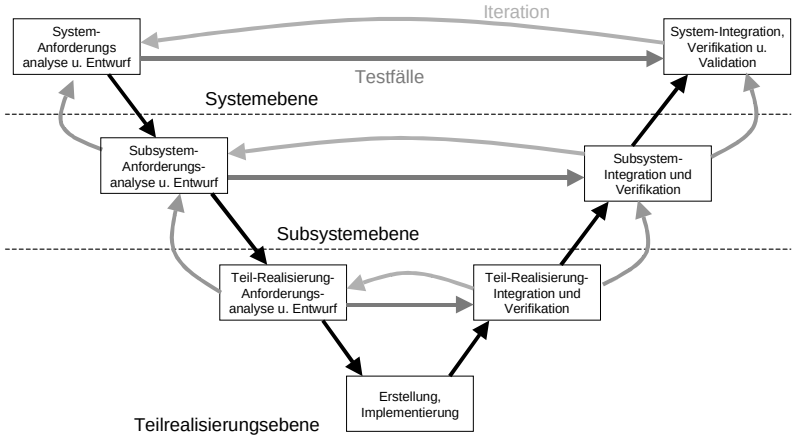

Abbildung 3.3.: Prinzipdarstellung des V-Modells nach [BTS+02]

Das V-Modell hat sich mittlerweile zum Quasi-Standard für die Entwicklung elektronischer Systeme auch im Automobilbereich entwickelt, da es eben in besonderer Weise den im Automobilbereich sehr intensiv genutzten Test berücksichtigt. In Kapitel 3.3.1 wird daher noch einmal besonders auf das V-Modell eingegangen, dann mit besonderem Fokus auf den automobilen Kontext.

Das V-Modell wurde als sogenanntes „V-Modell ’97“ von der IABG[1] für die Entwicklung von IT-Systemen des Bundes in [IAB02a] standardisiert.

Die Dokumentation des V-Modell ’97 umfasst das Vorgehensmodell sowie eine Methodenzuordnung und Funktionale Werkzeuganforderungen. Das Vorgehensmodell besteht aus vier Submodellen Projektmanagement (PM), Qualitätssicherung (QS), Konfigurationsmanagement (KM) und Systemerstellung (SE). Das Zusammenspiel der vier Submodelle zeigt Abbildung 3.4 auf der nächsten Seite.

[1] IABG: Industrieanlagen-Betriebsgesellschaft, 1961 von der Bundesrepublik Deutschland mit den Aufgaben Sicherheits- und Verteidigungsberatung sowie Test-Dienstleistungen für die Luftfahrtindustrie gegründet.

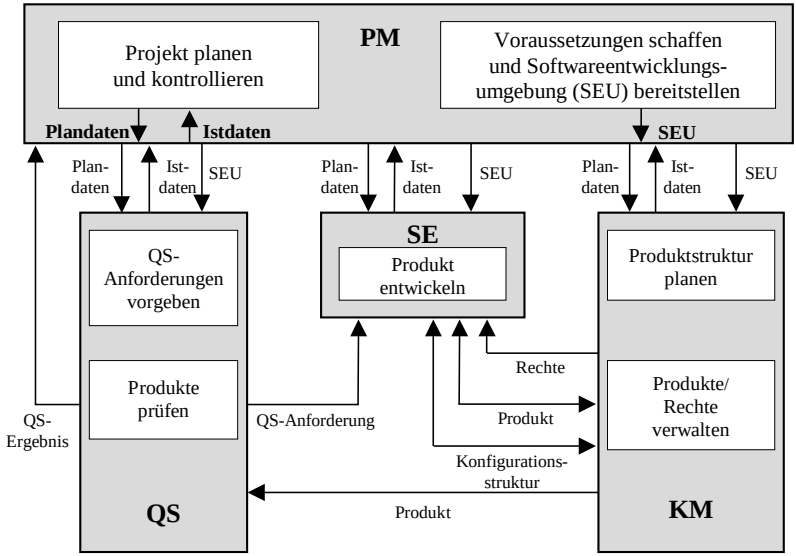

Abbildung 3.4.: Zusammenspiel der 4 Submodelle im V-Modell ’97

Während die drei Submodelle Qualitätssicherung, Projektmanagement und Konfigurationsmanagement die begleitenden Aktivitäten in einem Entwicklungsprojekt beschreiben, wird die Entwicklung selbst im Submodell Systemerstellung durchgeführt. Sein Aussehen wie ein großes V ist der Grund für den Namen „V-Modell“. Abbildung 3.5 auf der nächsten Seite zeigt das Submodell Systemerstellung.

Das Submodell SE untergliedert sich in neun Einzelschritte, die miteinander in Beziehung stehen. Die kleinsten Einheiten dabei sind die sogenannten Aktivitäten, für die keine zeitliche Abfolge definiert ist. Da die Aktivitäten aber voneinander abhängen, ergibt sich eine Reihenfolge implizit. Tabelle 3.1 auf Seite 44 erläutert die Aktivitäten im Submodell Systemerstellung.

Der besondere Schwerpunkt im V-Modell ’97 liegt auf der Entwicklung von Software, auf die Besonderheiten bei der Hardwareerstellung wird in einem Anhang eingegangen. Das V-Modell enthält auch Regelungen, die die Erstellung sicherheitskritischer Systeme beschreiben (dies ist der Anhang SI: „Sicherheit und Kritikalität“). Meist wird im V-Modell ’97 Sicherheit jedoch im Sinne der englischen Bedeutung „security“ verwendet.

Weitere Informationen zum V-Modell ’97 finden sich in Anhang A ab Seite 181.

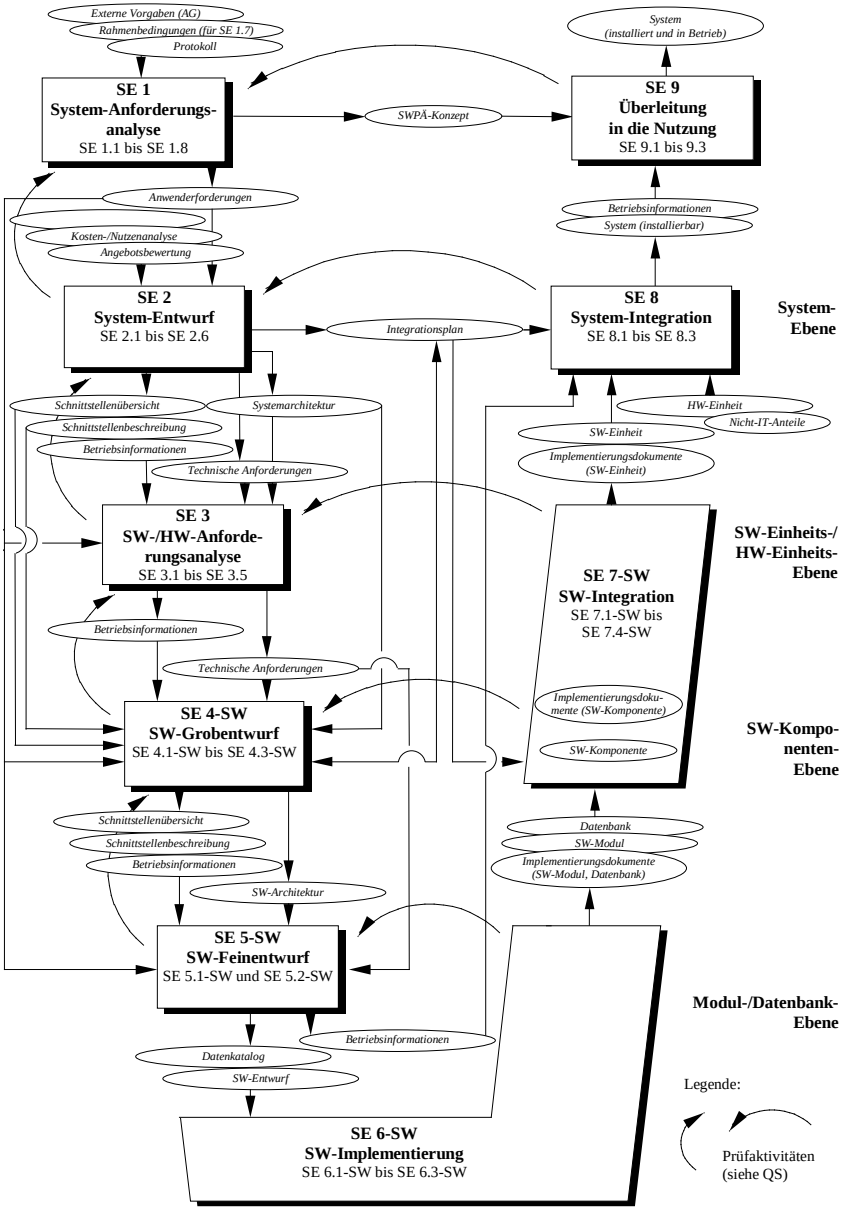

Abbildung 3.5.: Das Submodell Systemerstellung im V-Modell ’97

Tabelle 3.1.: Die Einzelschritte im Submodell Systemerstellung

SE 1	System-Anforderungsanalyse	Erarbeiten der Anwenderforderungen und Beschreibung des Anwendungssystems. Definition von Randbedingungen unter Berücksichtigung von Kritikalität und von Qualitätsanforderungen sowie die Durchführung eines Forderungscontrollings und die Erarbeitung eines Software-Planungs- und Änderungs-Konzepts.
SE 2	System-Entwurf	Erarbeiten einer Systemarchitektur und eines Integrationsplans sowie die Dokumentation bereits erkennbarer Betriebsinformationen und Untersuchung der Realisierbarkeit.
SE 3	Software-/Hardware-Anforderungsanalyse	Präzisierung der technischen Anforderungen an die Software- und Hardware-Einheiten. Es entstehen ein physikalisches Funktionsmodell, funktionale Spezifikationen und Testfälle auf Systemebene. Von hier ab spaltet sich der weitere Fortgang in die Software-Entwicklung und in die Hardware-Entwicklung
SE 4	Software-Grobentwurf	Entwurf der Software-Architektur, Beschreibung der Software-Schnittstellen und Fortführung des Integrationsplans auf Software-Ebene.
SE 5	Software-Feinentwurf	Beschreibung von Software-Komponenten auf Basis der Software-Architektur und der Schnittstellenbeschreibung, außerdem Analyse von Betriebsmittel- und Zeitbedarf.
SE 6	Software-Implementierung	Realisierung der Software-Module durch Codierung, Compilieren und Linken.
SE 7	Software-Integration	Integration der Software-Module zu Software-Einheiten.
SE 8	System-Integration	Integration des Systems aus Software-Einheiten, Hardware-Einheiten und Nicht-IT-Anteilen.
SE 9	Überleitung in die Nutzung	Installation des Systems und Inbetriebnahme.

3.1.4. Das VP-Modell

Das VP-Modell ist ein dreiphasiger Ansatz auf Basis des V-Modells. Er geht auf Ratcliffe (vgl. [Rat88]) zurück und wurde von Burst in [Bur00] verfeinert. Der traditionelle Entwurfsablauf nach dem V-Modell wird durch den Einsatz von Rapid Prototyping auf mehreren Ebenen der Systementwicklung unterstützt und beschleunigt (siehe Abbildung 3.6 auf der nächsten Seite).

In der Arbeit von Burst werden drei Phasen des Rapid Prototyping unterschieden: Konzept-orientiertes, Architektur-orientiertes und Implementierung-orientiertes Rapid Prototyping. Da Zielsetzung und zeitliches Auftreten der drei Ansätze unterschiedlich sind, wird jede Form des Rapid Prototypings in bestimmten Phasen

des Systementwurfs eingesetzt. Die drei Formen stehen somit nicht in Konkurrenz zueinander, sondern ergänzen sich.

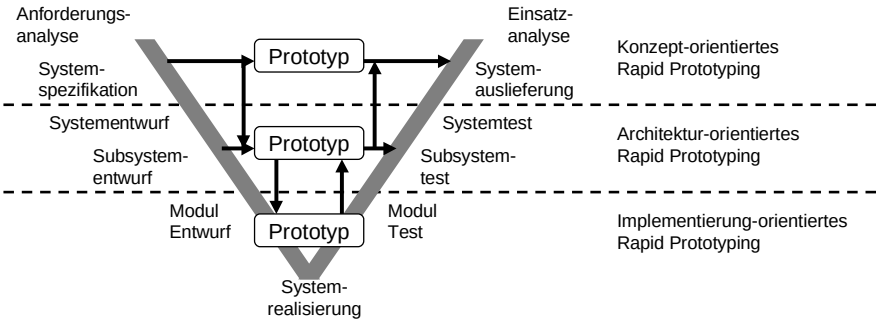

Abbildung 3.6.: Prinzipdarstellung des VP-Modells (nach Burst [Bur00])

3.2. Systementwicklung in verschiedenen Industriebereichen

Im Automobilbereich wird bzgl. der besonderen Berücksichtigung von Sicherheit und Zuverlässigkeit bei der Entwicklung sicherheitsrelevanter Elektroniksysteme zur Zeit Wissen und Erfahrung aufgebaut. Gerade X-by-Wire Systeme sind vielfältig noch in Forschung oder Entwicklung.

Anders sieht dies in anderen Industriezweigen aus. Im Luftfahrtbereich, in dem speziell X-by-Wire Systeme im militärischen und auch im zivilen Bereich schon länger im Einsatz sind[2], kann hier auf langjährige Erfahrung bei der Systementwicklung zurückgegriffen werden.

In den folgenden Kapiteln wird ein Überblick über Entwicklungsmethoden und Entwicklungsprozesse im Automobilbereich, in der Luftfahrt und bei der Prozessautomatisierung gegeben. Besonders wird dabei die Behandlung von sicherheitsrelevanten Systemen betrachtet.

Vereinfacht liegt der Schwerpunkt bei der Systementwicklung im Luftfahrtbereich auf einer entsprechenden Verfügbarkeit und Zuverlässigkeit der eingesetzten

[2] Das erste Fly-by-Wire-Militärflugzeug war 1972 die F-8 DFBW, in Serie ging Fly-by-Wire im Jahr 1978 mit der F-18 „Hornet“; das erste zivile Fly-by-Wire-Flugzeug war 1987 der Airbus A320.

Elektroniksysteme, bei der Prozessautomatisierung auf der Sicherheit des Technischen Systems. Für sicherheitsrelevante Elektroniksysteme im Automobilbereich wird sich wohl zukünftig eine Mischung aus allen Anforderungen ergeben.

3.3. Systementwicklung im Automobilbereich

Bei der Entwicklung elektronischer Systeme im Automobil sind besondere Randbedingungen zu beachten.

Im Vergleich zu anderen Bereichen der Industrie - insbesondere zu solchen mit ähnlichen Anforderungen an Sicherheit und Zuverlässigkeit - werden im Automobilbereich Systeme mit relativ großen Stückzahlen gebaut. Dies bedeutet aber im Gegenzug, dass sich Anpassungen an Besonderheiten eines speziellen Systems eher rentieren als bei kleinen Stückzahlen. Daraus und aus kunden- oder länderspezifischen Anpassungen resultiert eine große Variantenvielfalt elektronischer Systeme.

Ein weiterer besonderer Aspekt im Automobilbereich ist die Forderung nach der Zusammensetzbarkeit elektronischer Systeme (sog. „composability“), die v. a. bei der Zusammenarbeit zwischen Fahrzeughersteller und mehreren Zulieferern eine immer größer werdende Rolle spielt.

Da die optimale Wartung von Kraftfahrzeugen durch den Kunden nicht unbedingt gewährleistet ist, laufen v. a. die mechanischen Komponenten Gefahr, frühzeitig Probleme zu bereiten. Zudem erhalten Autofahrer im Gegensatz beispielsweise zu Piloten oder Bedienern von Prozessautomatisierungssystemen wenig bis gar keine besondere Schulung für ihr Kraftfahrzeug.

Eine Besonderheit bei der Entwicklung von Fahrzeugen und Fahrzeugsystemen im Automobilbereich ist der große Fokus auf den System-Test. Über mehrere Monate hinweg werden zunächst die Teilsysteme und anschließend das Gesamtsystem „Fahrzeug“ intensiv erprobt und getestet.

3.3.1. Automotive V-Modell

Die Vorgehensweisen bei der Systementwicklung im Automobilbereich sind so vielfältig, wie es Hersteller und Zulieferer auf dem Markt gibt. Vor allem die Prozesslastigkeit schwankt dabei sehr stark.

Als ein sehr häufig verwendetes Vorgehensmodell für die Systementwicklung im Automobil hat sich das V-Modell (vgl. Kapitel 3.1.3) herauskristallisiert. Die grundsätzliche Vorgehensweise entspricht dabei dem V-Modell ’97, meist mit v. a.

Entwicklungswerkzeug-getriebenen Anpassungen, die aber auch in [IAB02a] explizit vorgesehen sind (sog. „Tailoring“). Weite Teile der Entwicklung werden durch besondere Entwicklungswerkzeuge unterstützt, teilweise wird „Rapid Prototyping“ verwendet. Allerdings gibt es im Automobilbereich zur Zeit keinen allgemein gültigen Entwicklungsstandard, im Gegensatz zum Luftfahrtbereich oder zur Prozessautomatisierung.

Detaillierte Darstellungen von Entwicklungsprozessen im Automobilbereich finden sich in [BRI99], [Bor02] oder [Hed01]. In Abbildung 3.7 nach [BRI99] ist eine am V-Modell orientierte Vorgehensweise dargestellt. Man erkennt neben Testfällen auch Rapid Prototyping.

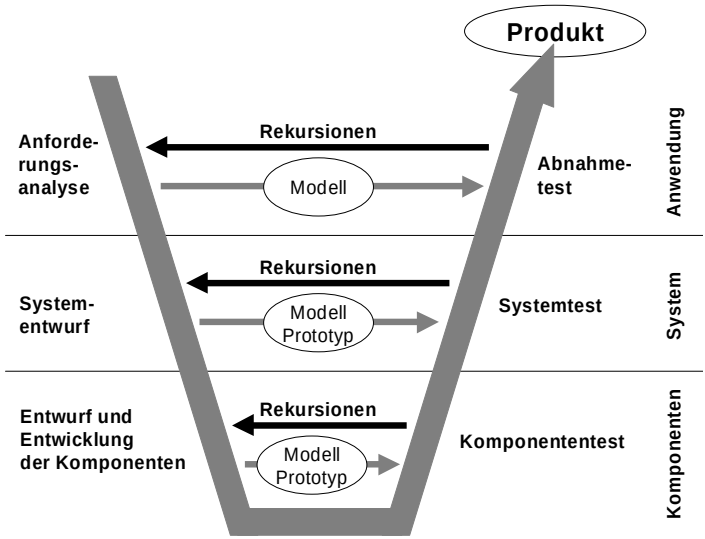

Abbildung 3.7.: Vorgehen bei der Systementwicklung im Automobil

Eine sehr viel stärker detaillierte Darstellung findet sich in [Bor02]. Sie ist in Abbildung 3.8 auf der nächsten Seite wiedergegeben. In der Regel findet die untere Hälfte der Systementwicklung beim Zulieferer statt, der Hersteller liefert die Systemspezifikation und integriert dann das fertige System (links und rechts oben in Abbildung 3.8).

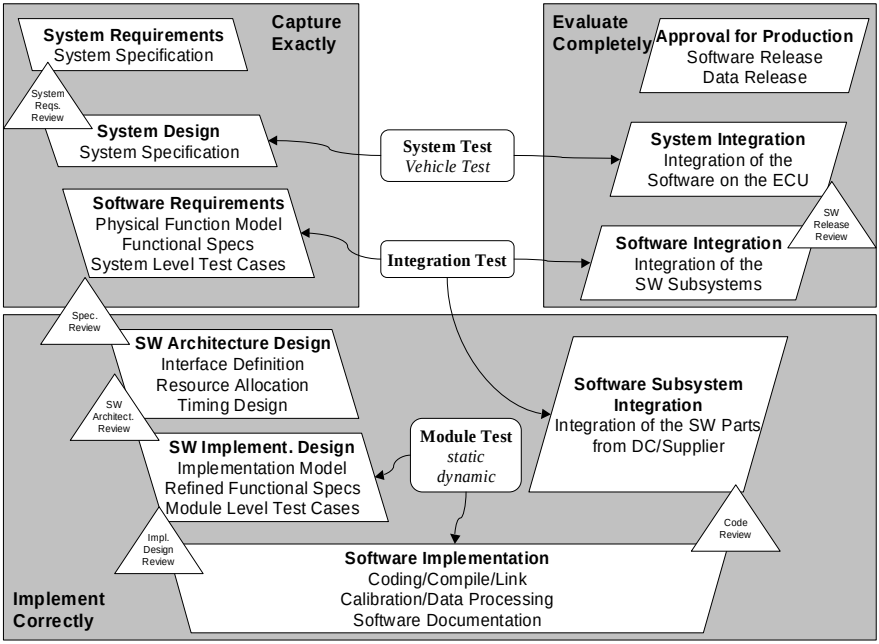

Abbildung 3.8.: Das Automotive V-Modell nach [Bor02]

Im Rahmen dieser Arbeit wird das V-Modell ’97 nach [IAB02a] in einem entsprechend dem Automobilbereich angepassten Verständnis als formale Referenz für Vorgehensmodelle bei der Systementwicklung im Automobil verwendet.

3.3.2. Qualität, Zuverlässigkeit und Sicherheit

Ein Käufer eines Kraftfahrzeugs möchte vor allem ein preiswertes Fahrzeug mit hoher Qualität und Zuverlässigkeit, damit für ihn möglichst wenige Werkstattbesuche anfallen. Der Kunde erwartet die ständige Funktionsbereitschaft seines Wagens.

Doch nicht nur der Kunde hat bestimmte Vorstellungen von Zuverlässigkeit, auch die jeweiligen Gesetzgeber stellen aus Gründen der Verkehrssicherheit, des

Umweltschutzes und des Verbraucherschutzes im Rahmen der Produkthaftung[3] Forderungen, denen entsprochen werden muss. Die zu realisierenden Zuverlässigkeitsanforderungen müssen schon während der Systementwicklung berücksichtigt werden (nach [VDA00]).

Eine hohe Produktzuverlässigkeit kann aber nicht alleine über ausgereifte Konstruktionsmethoden und -verfahren sichergestellt werden, daher kommen verschiedene analytische Zuverlässigkeitsmethoden zum Einsatz. Um die Erfüllung von strengen Zuverlässigkeitsanforderungen zu gewährleisten ist eine nur qualitative Aussage über die Produktzuverlässigkeit (wie sie durch Werkzeuge wie FMEA oder Design-Reviews gewonnen werden kann) nicht ausreichend, daher werden mehr und mehr auch quantitative Methoden eingesetzt (Fehlerbaumanalyse, Markov-Analyse, Monte-Carlo-Simulation etc.).

Zuverlässigkeitsbetrachtungen im Rahmen der Qualitätssicherung sind ein im Automobilbereich wichtiger und daher auch breit eingeführter Bestandteil der Systementwicklung, der durch entsprechende Standards wie DIN EN ISO 9000 „Qualitätsmanagementsysteme“, DIN EN 60300 „Zuverlässigkeitsmanagement“oder firmeninterne Richtlinien unterstützt wird.

Auch im V-Modell ’97 (vgl. Kapitel 3.1.3) ist das Qualitätsmanagement geregelt, das Submodell Qualitätssicherung (QS) ist in Abbildung 3.9 auf der nächsten Seite dargestellt. Der Schwerpunkt liegt hierbei auf der Produktprüfung.

Betrachtungen der Systemsicherheit wurden bisher auch schon durchgeführt, v. a. unter Anwendung von FMEA und FMECA. Allerdings ist, auch mangels einer zwingenden Notwendigkeit bedingt durch entsprechende Sicherheitsanforderungen, ein systematischer, durchgängiger Sicherheitsprozess in der Automobilindustrie nicht vorhanden (nach [MP03]).

Betrachtungen im Sinne eines Sicherheitslebenszyklus (engl. „safety life cycle“), wie sie in [ADM+00], [GR02] oder insbesondere in [JW03] angestellt werden, sind im Automobilbereich recht neu und werden daher auch entsprechend kontrovers diskutiert. Die Folgen gerade für eine zukünftige Vorgehensweise bei der Systementwicklung im Automobil sind noch nicht absehbar.

[3] Produkthaftung ist ein Sammelbegriff für die Verpflichtung eines Herstellers zum Schadensersatz aufgrund fehlerhafter Produkte. Dabei dient als Maßstab, dass die Entwicklung nach dem Stand der Wissenschaft und Technik und nach bestem Wissen und Gewissen durchgeführt wurde.

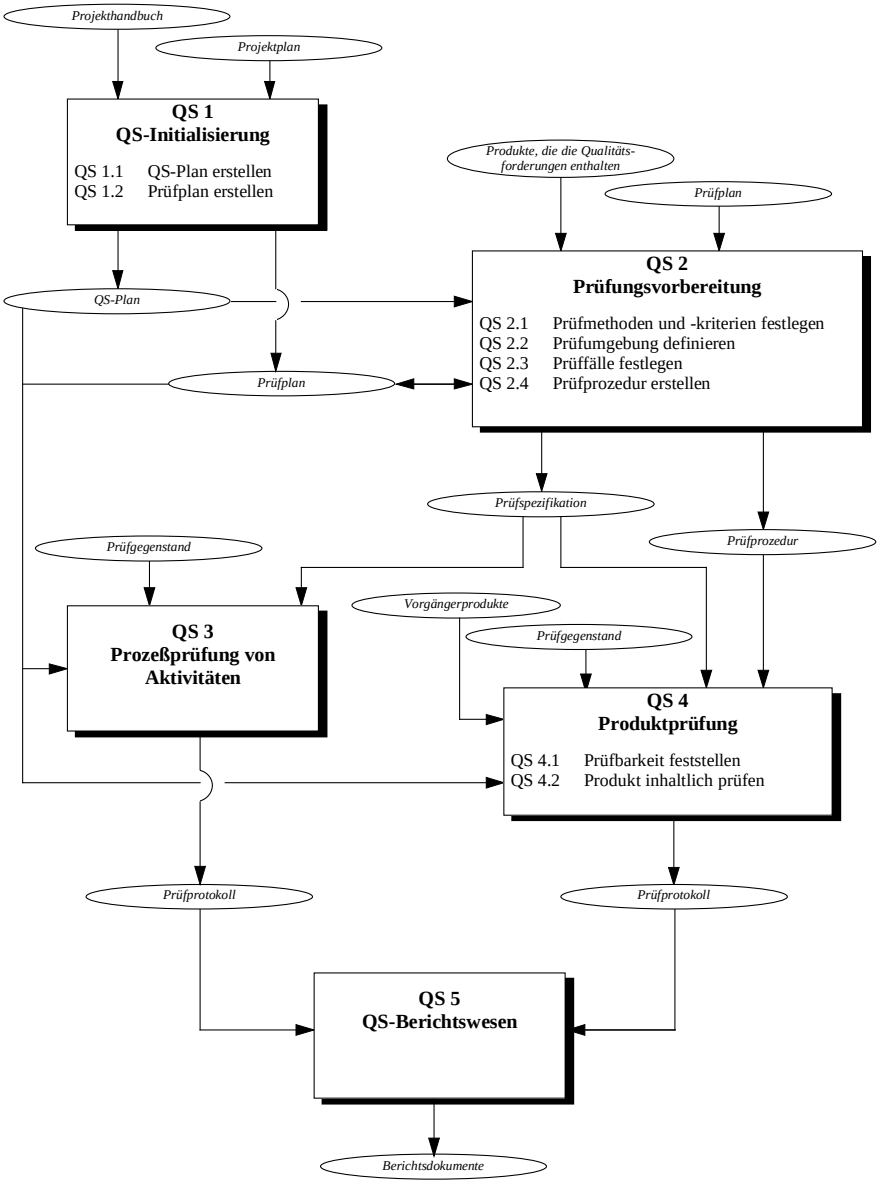

Abbildung 3.9.: Das Submodell Qualitätssicherung im V-Modell ’97

3.4. Systementwicklung im Luftfahrtbereich

Wie schon angedeutet unterscheiden sich die Randbedingungen für die Entwicklung von elektronischen Systemen im Luftfahrtbereich von denen im Automobilbereich.

Von Flugzeugelektronik wird in erster Linie eine hohe Verfügbarkeit verlangt. Ausfälle von Komponenten und Systemen müssen zumindest bis zum nächsten Flughafen toleriert werden können, dort muss mindestens noch eine sichere Landung möglich sein. Systeme, deren Ausfall die Sicherheit des Flugzeuges beeinträchtigen kann, wie z. B. die Fluglageregelung oder das System für die automatisierte Landung, müssen darüber hinaus eine sehr hohe Zuverlässigkeit aufweisen (vgl. [KG02a]).

Zusätzlich muss im Luftfahrtbereich zwischen zivilen und militärischen Anwendungen unterschieden werden, für die jeweils andere Randbedingungen gelten. Es erscheint auf den ersten Blick paradox, dass viele Anforderungen an Sicherheit und Zuverlässigkeit im militärischen Bereich niedriger sind als im zivilen Bereich. Allerdings muss man in Betracht ziehen, dass beim Einsatz militärischer Systeme die Wahrscheinlichkeit, durch Feindeinwirkung Schaden zu erleiden, um Größenordnungen höher ist, als durch einen technischen Fehler einen Defekt zu bekommen. Zudem besitzen Kampfflugzeuge in der Regel noch die „Rückfallebene Schleudersitz“. Die geforderten maximalen Ausfallwahrscheinlichkeiten für den Verlust eines Flugzeuges aufgrund eines Elektronikdefektes liegen laut EADS[4] im militärischen Bereich bei ca. $10^{-5}/h$, bei zivilen Flugzeugen bei etwa $10^{-6}/h$.

Daher kommt als Referenz für die weitere Betrachtung nur die zivile Luftfahrt in Frage.

Ein wesentlicher Unterschied zum Automobilbereich ist die sehr viel geringere Stückzahl, in der elektronische Systeme im Luftfahrtbereich hergestellt werden. Dadurch fallen Kosten für ein Mehr an elektronischen Komponenten im Aerospacebereich sehr viel weniger ins Gewicht als eine personalintensive und langwierige Entwicklung und Erprobung eines neuen Konzepts. Deshalb wird Sicherheits- und Zuverlässigkeitsanforderungen meist durch die Verwendung von aufwändigen Redundanzkonzepten Rechnung getragen. Andererseits rechnen sich spezielle Lösungen oftmals nicht, so dass mehr und mehr auf Standardkomponenten zurückgegriffen wird.

Abbildung 3.10 auf der nächsten Seite nach [Yeh96] zeigt das Redundanzkonzept für den Fluglageregler (auch „Primary Flight Control PFC“ genannt) in der

[4] EADS: European Aeronautic Defence and Space Company

Boeing 777. Der Rechner ist dreifach redundant ausgeführt, ebenso das Bussystem. Jeder redundante Knoten ist wiederum dreifach diversitär redundant, woraus sich eine sogenannte „Triple-Triple-Architektur“ ergibt.

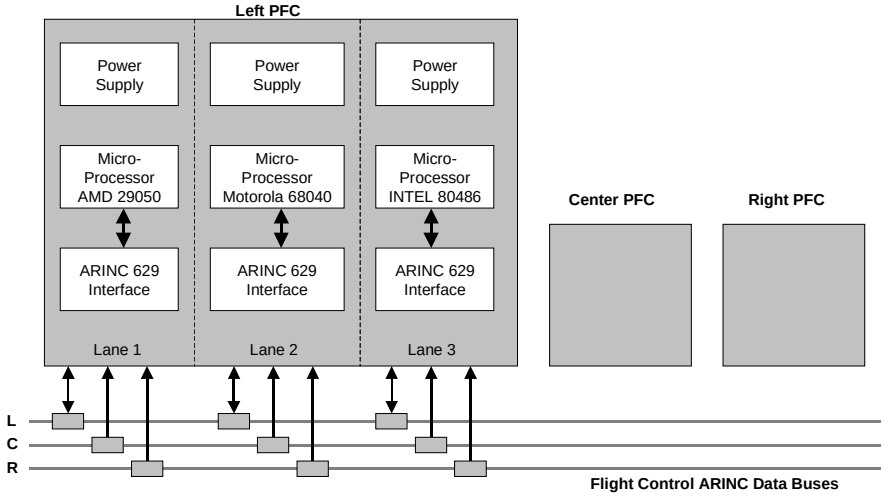

Abbildung 3.10.: Triple-Triple-Redundanzstruktur für den Fluglageregler einer Boeing 777

Ein Grundsatz bei der Entwicklung von Systemen im Luftfahrtbereich ist die Verwendung von Entwicklungsstandards sowohl bei der Entwicklung als auch für die zu entwickelnden Komponenten. Die Vorgehensweise ist dabei genau festgelegt und geregelt, es gibt regelmäßige Reviews mit allen beteiligten Partnern. Wesentlich ist, dass das zu entwickelnde System am Ende von einer unabhängigen Instanz zertifiziert wird. Werden für die Konzeption des Systems bereits zertifizierte Komponenten verwendet, so können diese bei der Systembetrachtung als bereits zugelassen angesehen werden. Eine erneute Betrachtung und Zertifizierung dieser Systemteile ist dann nicht mehr notwendig.

In Europa ist für die Zulassung die „Joint Aviation Authority“, kurz JAA, zuständig, in den USA die „Federal Aviation Association“, kurz FAA. Die relevanten Richtlinien für die Zulassung sind die Joint Aviation Regulations, kurz JAR, bzw. Federal Aviation Regulations, kurz FAR. Dabei kommt für die Systementwicklung von Avioniksystemen in der Regel der Standard SAE ARP 4754[5] (vgl. [SAE96a])

[5] ARP: Aerospace Recommended Practice

zum Einsatz, das europäische Pendant von Eurocontrol (vgl. [Bea00]) ist mit diesen im Wesentlichen identisch (nach [Kne00]).

Abbildung 3.11 zeigt die Vorgehensweise bei der Systementwicklung gemäß SAE ARP 4754 und die dabei verwendeten weiteren Standards.

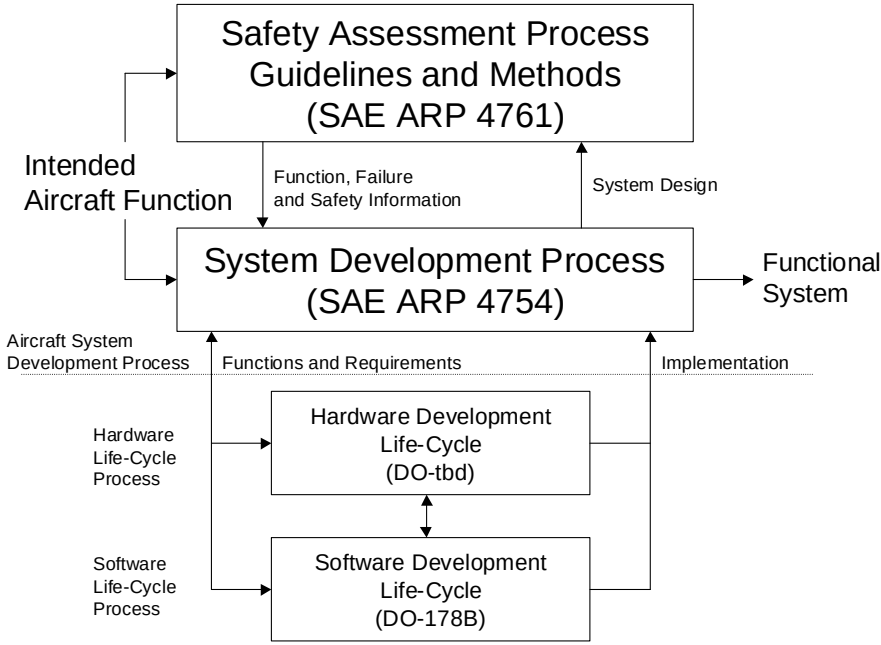

Abbildung 3.11.: Systementwicklung im Luftfahrtbereich gemäß SAE ARP 4754

Die Sicherheitsphilosophie in der Luftfahrt basiert dabei auf der quantitativen Definition von Sicherheitszielen, für die im Rahmen eines entsprechenden Prozesses ein Nachweis mit entsprechenden Analysewerkzeugen zu erbringen ist. Dabei wird davon ausgegangen, dass Ausfälle lebenswichtiger Systeme innerhalb eines bestimmten Intervalls der Lebensdauer eines Flugzeugs extrem unwahrscheinlich sein müssen. Je schwerer die Auswirkungen eines Fehlers sind, umso geringer muss die Wahrscheinlichkeit für das Auftreten eines derartigen Fehlers sein.

Die verwendeten Sicherheits- und Zuverlässigkeitsmethoden werden über alle Phasen des Entwicklungsprozesses angewandt. Hierbei handelt es sich v. a. um quantitative Methoden. So erfolgt der Nachweis der Sicherheitseigenschaften durch

sehr intensive Nutzung von Fehlerbäumen mit meist etlichen Tausend Elementen. Die Vorgehensweise ist dabei dahingehend ausgerichtet, die Anforderungen der Luftfahrtbehörden (v. a. Sicherheitsanforderungen) und der Luftfahrtgesellschaften als Kunden (Sicherheits- und v. a. Zuverlässigkeitsanforderungen) zu erfüllen.

In SAE ARP 4754 wird der Systementwicklungsprozess nur allgemein beschrieben, für weitere Details werden andere Dokumente referenziert. So kommt bei der Softwareentwicklung RTCA/DO-178B (vgl. [DO 92]) zum Einsatz. DO-178B „Software Considerations in Airborne Systems and Equipment Certification“ stellt Richtlinien zur Verfügung, deren Anwendung sicherstellt, dass Software in Luftfahrtsystemen die Luftfahrt-spezifischen Anforderungen erfüllt. Der Standard beschreibt dafür einen Software-Planungs-, Software-Entwicklungs- und einen Software-Integrations-Prozess. Zur Betrachtung von Sicherheitsanforderungen für Software wird diese in fünf Sicherheitsklassen von Level A (Katastrophal) bis E (keine Auswirkungen) eingeteilt.

Für die Berücksichtigung von Sicherheit bei der gesamten Systementwicklung wird von SAE ARP 4754 auf SAE ARP 4761 (vgl. [SAE96b]) verwiesen. Diese beiden Standards sind eng miteinander verwoben, die Berücksichtigung der Sicherheit des betrachteten Systems in SAE ARP 4761 erfolgt parallel zur Entwicklung der Systemfunktion, die in SAE ARP 4754 beschrieben ist. Die Vorgehensweise bei der Betrachtung der Sicherheit, der sogenannte „Safety Assessment Process“ von SAE ARP 4761, ist in Abbildung 3.12 auf der nächsten Seite dargestellt.

Wesentliche Elemente des „Safety Assessment Process“ sind eine Gefährdungsanalyse („Functional Hazard Assessment“ FHA[6]) zu Beginn der Systementwicklung und entsprechende begleitende Schritte zur Überprüfung der Sicherheitsanforderungen während der Systementwicklung („Preliminary System Safety Assessments“ PSSA und „System Safety Assessments“ SSA), die durch entsprechende qualitative und quantitative Methoden wie die Fehlerbaumanalyse unterstützt werden.

Eine Besonderheit des Vorgehens ist die konsequente Trennung von Funktionsentwicklung (sog. „System Development Process“) und der Berücksichtigung der Sicherheit im Entwicklungsprozess (eben der „Safety Assessment Process“). Dies geht so weit, dass getrennte Entwicklungsteams für Funktion und für Sicherheit zuständig sind und sich - gemeinsam mit der Zertifizierungsbehörde - über die Entwicklung des Systems austauschen. Diese Vorgehensweise ist in Abbildung 3.13 auf Seite 56 dargestellt.

[6] Eine Bewertung der FHA im Luftfahrtbereich findet sich in [WK], ein Beispiel für die Anwendung einer FHA für die Eisenbahnsignaltechnik in [Wer02]

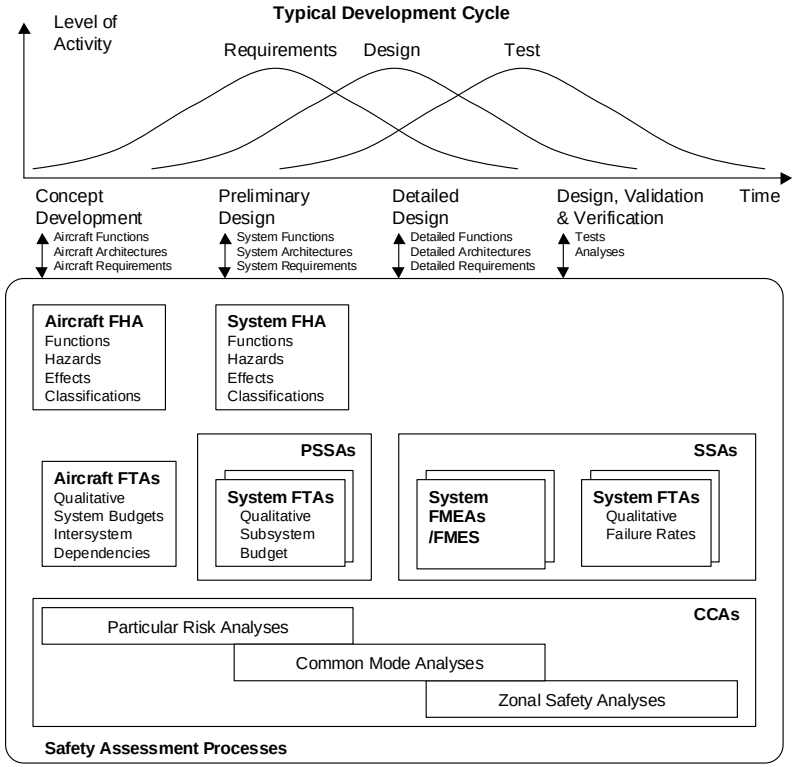

Abbildung 3.12.: Der Safety Assessment Process nach SAE ARP 4761

Da die Randbedingungen und die Anforderungen im Luftfahrbereich sich deutlich von denen im Automobilbereich unterscheiden, können auch die dort verwendeten Prozesse bei der Systementwicklung nicht direkt übertragen werden. Kernelemente können verwendet werden, müssen aber an die Randbedingungen in der Automobilwelt angepasst werden.

Eine interessante Eigenschaft beim Vorgehen zur Entwicklung sicherheitsrelevanter Elektroniksysteme im Luftfahrtbereich, die im Automobilbereich wiederverwendet werden kann, ist die klare Trennung zwischen Funktion und Sicherheit bei der Systementwicklung und die Verwendung entsprechender Methoden auf

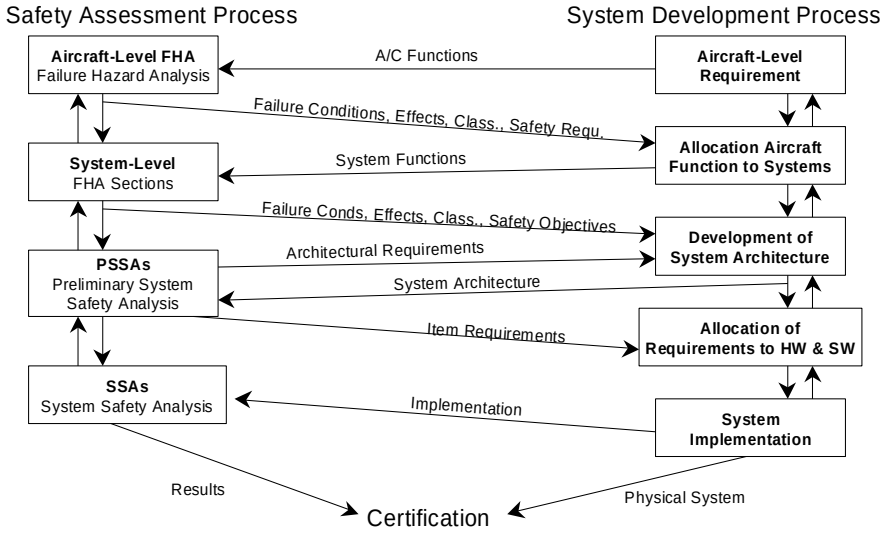

Abbildung 3.13.: Der „Safety Assessment Process“ und der „System Development Process“ nach SAE ARP 4754

quantitativer Basis. So kann eine korrekte Entwicklung sicherheitsrelevanter Elektroniksysteme garantiert und nachgewiesen werden.

Weitere Informationen zu SAE ARP 4754 und SAE ARP 4761 finden sich in Anhang B ab Seite 195.

3.5. Systementwicklung in der Prozessautomatisierung

Ein Prozessautomatisierungssystem unterscheidet sich in seiner Struktur grundsätzlich von Systemen im Automobil- oder Luftfahrtbereich. Es besteht im Allgemeinen aus vier Bestandteilen: dem technischen System, dem Leit- und Steuerungssystem, dem sicherheitsbezogenen System und dem Bedienpersonal (vgl. [Bie03], siehe Abbildung 3.14 auf der nächsten Seite).

Im Technischen System, oft auch „Equipment under Control“ oder kurz EUC genannt, läuft ein technischer Prozess ab, dessen Zustände sich als physikalische Größen mit technischen Mitteln erfassen lassen und dem Leit- und Steuerungs-

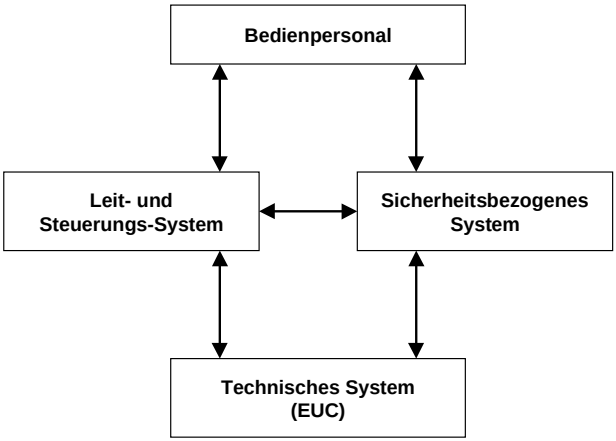

Abbildung 3.14.: Bestandteile eines Prozessautomatisierungssystems

system als Messsignale zur Verfügung stehen. Zustandsänderungen des technischen Prozesses können vom Leit- und Steuerungssystem mit Hilfe von Aktoren hervorgerufen werden.

Das sicherheitsbezogene System (nicht mit einem „sicherheitsrelevanten System“ zu verwechseln) überwacht das Technische System und das Leit- und Steuerungssystem und stellt Sicherheitsfunktionen zur notwendigen Risikominderung zur Verfügung (vgl. Kapitel 2.1.5).

Das Bedienpersonal kann je nach Automatisierungsgrad unterschiedliche Tätigkeiten wahrnehmen und über das Leit- und Steuerungssystem oder das sicherheitsbezogene System in die Vorgänge des technischen Prozesses eingreifen.

Um die funktionale Sicherheit einer Maschine oder Anlage zu erreichen ist es notwendig, dass das sicherheitsbezogene System korrekt funktioniert und sich im Fehlerfall so verhält, dass die Anlage in einen sicheren Zustand gebracht wird bzw. in einem sicheren Zustand bleibt. Dazu ist die Verwendung besonders qualifizierter Technik notwendig, die den in den betreffenden Normen beschriebenen Anforderungen genügt. Die Anforderungen zur Erzielung funktionaler Sicherheit basieren auf den grundlegenden Zielen der Vermeidung und der Beherrschung systematischer und zufälliger Fehler.

Im Bereich der Automatisierungstechnik sind mehrere Standards zur Betrachtung der Sicherheit bei der Systementwicklung etabliert, so beispielsweise der

Standard DIN EN 61508 (vgl. [DIN02]) oder der Standard EN 954 (vgl. [EN 97]). Charakteristisch ist dabei die Zertifizierung von Systemkomponenten, um daraus ein zertifiziertes Gesamtsystem zu erstellen.

Die Standards zielen dabei auf eine entsprechende Ausgestaltung des sicherheitsgerichteten Systems, um die Gefährdung, die vom Technischen System ausgeht, zu minimieren.

Durch den konkreten Zuschnitt auf den Bereich der Prozessautomatisierung sind die angeführten Prozessmodelle aber für die Automobilentwicklung nur bedingt geeignet. Auch ist das Konzept zertifizierter Komponenten im Automobilbereich aus Gründen der Variantenvielfalt, der Modularisierbarkeit und der Zusammensetzbarkeit von Systemen weitestgehend nicht umsetzbar.

3.5.1. DIN EN 61508

Die Norm DIN EN 61508 „Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme“ ist die deutsche Übersetzung der Norm IEC 61508, einer generischen Prozessnorm auf Basis der IEC 1508, die ursprünglich aus dem chemisch-prozesstechnischen Bereich stammt. Sie ist nicht zur direkten Anwendung gedacht, sondern soll als Grundlage für spezifische Ausprägungen für einzelne Industriebereich dienen. DIN EN 61508 ist seit 2002 gültige deutsche Norm, zum 01.08.2004 werden die bisher in Deutschland geltenden Normen DIN V 19250 („Leittechnik - Grundlegende Sicherheitsbetrachtungen für MSR-Schutzeinrichtungen“), DIN V 19251 („Leittechnik - MSR-Schutzeinrichtungen - Anforderungen und Maßnahmen zur gesicherten Funktion“) und DIN V VDE 0801 („Grundsätze für Rechner in Systemen mit Sicherheitsaufgaben“) zurückgezogen, sie gehen in DIN EN 61508 auf.

Die Norm behandelt die Entwicklung sicherheitsbezogener Systeme und deckt dazu alle Aspekte der Systementwicklung von den Systemanforderungen über Architektur- und Hardware-Design-Richtlinien bis hin zu Softwareanforderungen ab. Sie hat aber nicht die Entwicklung des Leit- und Steuerungssystems[7] oder gar des Technischen Systems zum Inhalt.

Hauptbestandteil der Norm DIN EN 61508 ist ein Lebenszyklusmodell für die Systementwicklung eines sicherheitsbezogenen Systems. Darin wird eine Reihe von Phasen beschrieben, die das Erreichen des notwendigen Sicherheitsniveaus

[7] Als Ausnahmefall ist in DIN EN 61508 das Zusammenfallen von Leit- und Steuerungssystem und sicherheitsbezogenem System behandelt.

garantieren sollen. Abbildung 3.15 auf der nächsten Seite zeigt diesen sogenannten Sicherheitslebenszyklus der Norm DIN EN 61508.

Konkret wird ein gegebenes Technisches System (EUC) hinsichtlich möglicher Gefährdungen untersucht. Auf Grundlage dieser Untersuchung wird das Technische System in einen von vier sogenannten „Sicherheitsintegritätslevel“, kurz SIL, eingestuft, um so das Maß der notwendigen Risikominderung zu quantifizieren. Dabei ist die Vorgehensweise zur Ermittlung des jeweiligen SIL nicht vorgeschrieben und kann qualitativ oder quantitativ erfolgen (vgl. Kapitel 2.1.9).

„Sicherheitsintegrität“ wird dabei als die Wahrscheinlichkeit verstanden, dass ein sicherheitsbezogenes System die geforderten Sicherheitsfunktionen unter allen festgelegten Bedingungen innerhalb eines festgelegten Zeitraumes anforderungsgemäß ausführt.

Basierend auf dieser Einstufung in einen Sicherheitsintegritätslevel ergeben sich Anforderungen an die Eigenschaften des sicherheitsbezogenen Systems. Dies reicht von Anforderungen an die maximale Ausfallwahrscheinlichkeit (vgl. Tabelle 3.2) bis hin zu einzusetzenden Programmiersprachen. Darüberhinaus wird die Vorgehensweise bei der Entwicklung basierend auf dieser Einstufung angepasst.

Tabelle 3.2.: Ausfallgrenzwerte für Sicherheitsfunktionen

SIL	Betriebsart mit niedriger Anforderungsrate	Betriebsart mit hoher Anforderungsrate bzw. kontinuierlicher Anforderung
4	$\geq 10^{-5}$ bis $< 10^{-4}$	$\geq 10^{-9}/h$ bis $< 10^{-8}/h$
3	$\geq 10^{-4}$ bis $< 10^{-3}$	$\geq 10^{-8}/h$ bis $< 10^{-7}/h$
2	$\geq 10^{-3}$ bis $< 10^{-2}$	$\geq 10^{-7}/h$ bis $< 10^{-6}/h$
1	$\geq 10^{-2}$ bis $< 10^{-1}$	$\geq 10^{-6}/h$ bis $< 10^{-5}/h$

Quelle: DIN EN 61508

In [Sto96] ist die Vorgehensweise nach DIN EN 61508 bei der Entwicklung von Speicherprogrammierbaren Steuerungen (kurz SPS) beispielhaft dargestellt. In [SS01] sind zusätzliche Erläuterungen zu finden.

Weitere Informationen zur DIN EN 61508 finden sich in Anhang C ab Seite 203.

3.5.2. EN 954

Die Norm EN 954 „Sicherheitsbezogene Teile von Steuerungen“ (vgl. [EN 97]) hat ähnlich wie DIN EN 61508 die Entwicklung von sicherheitsbezogenen Systemen

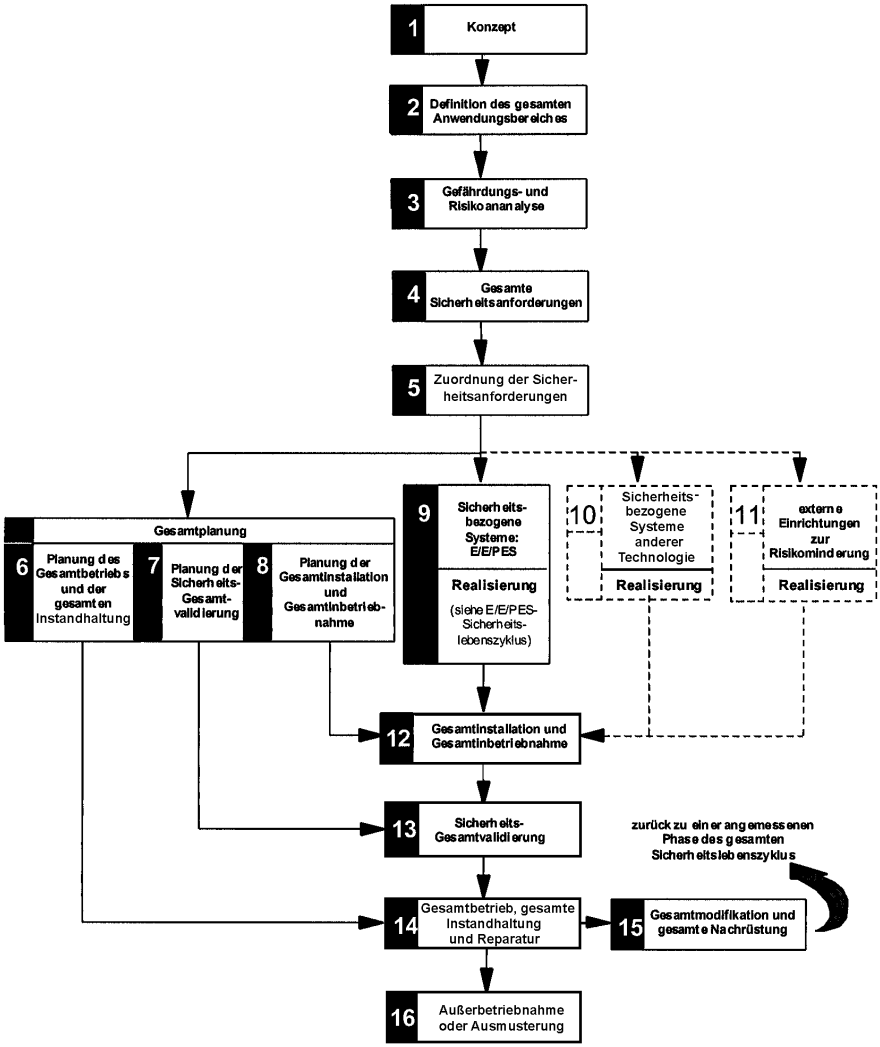

Abbildung 3.15.: Der Sicherheitslebenszyklus in DIN EN 61508

zum Inhalt. Die Leistungsfähigkeit eines sicherheitsbezogenen Teils einer Steuerung beim Auftreten von Fehlern wird dabei in fünf Kategorien unterteilt (Kategorie B, 1, 2, 3 und 4, mit zunehmender Gefährdung). Diese Kategorien können für Steuerungen aller Maschinenarten und Steuerungen von Schutzeinrichtungen angewandt werden. Die Risikobeurteilung erfolgt dabei gemäß EN 1050 (vgl. [EN 96]).

Basierend auf der Kategorieneinstufung ergeben sich grundsätzliche Architekturvorschriften, wie das entsprechende System zu gestalten ist. In [BIA97] sind zahlreiche Beispiele für die Ausgestaltung von Systemen basierend auf der Kategorieneinstufung nach EN 954 enthalten.

Eine allgemeine Darstellung zur Gestaltung sicherheitsbezogener Systeme im Prozessautomatisierungsbereich findet sich in [Mon99].

3.5.3. MIL-STD-882C/D

Einer der einflussreichsten Entwicklungsstandards in den USA ist der Standard MIL-STD-882 „System Safety Program Requirements“(vgl. [MIL99]).

MIL-STD-882 regelt detailliert einen Systemsicherheitsprozesses für alle elektronischen Systeme, die vom amerikanischen Verteidigungsministerium „Department of Defense“ entwickelt oder in Auftrag gegeben werden.

Im Rahmen des Standards werden Anforderungen für die Entwicklung und Implementierung eines Sicherheitsprogramms festgelegt. Hauptziel des Sicherheitsprogramms ist die Definition eines systematischen Vorgehens, um sicherzustellen, dass die Sicherheitsanforderungen des betreffenden Systems zusammen mit seinen funktionalen Anforderungen in einer zeit- und kostenoptimalen Art und Weise umgesetzt werden. Dazu werden mögliche Gefährdungen werden identifiziert, bewertet und beseitigt bzw. das verbundene Risiko wird auf ein akzeptables Maß reduziert. Der Standard beschreibt dazu 22 einzelne Werkzeuge und Methoden, wie z. B. einen System-Sicherheitsprogrammplan oder Gefährdungsanalysen, zusätzlich stellt er auch Anforderungen an die beteiligten Organisationen. Die Aktivitäten des Standards beziehen sich auf den gesamten Systemlebenszyklus.

Die Version MIL-STD-882D hat im Jahre 1999 MIL-STD-882C ersetzt. Allerdings ist die aktuelle Version in ihren Ausführungen und Vorgaben sehr viel allgemeiner und weniger detailliert als ihr Vorgänger, so dass in der Regel immer noch die Version MIL-STD-882C zum Einsatz kommt (vgl. [CDJM03]).

3.6. Reifegradmodelle

Zur Beurteilung des Reifegrades eines Entwicklungsprozesses - ursprünglich nur für Softwaresysteme - sind sogenannte Prozess-Verbesserungs-Modelle entstanden. Mit Hilfe der dort beschriebenen Methoden ist es möglich, einen Entwicklungsprozess bewerten zu können.

Das bekannteste Referenzmodell zur Bewertung von Organisationen bezüglich der Reife des Softwareentwicklungsprozesses ist CMM. Das „Capability Maturity Model“ CMM entstand am Systems Engineering Institute SEI der Carnegie Mellon University in den USA (vgl. [SEI04]). Es beinhaltet ein fünfstufiges Modell, das die Stufen „initial“, „repeatable“, „defined“, „managed“ und „optimizing“ kennt.

Ein Prozess, der in Stufe „initial“ eingestuft wurde, ist durch eine chaotisch und nach außen undurchsichtige Vorgehensweise geprägt. Ein Prozess der Stufe zwei „repeatable“ ist wiederholbar und hat Meilensteine. Auf Stufe drei „defined“ ist ein organisationsweit gültiger Software-Prozess definiert und eingeführt. Bei einem Prozess nach Stufe vier „managed“ wird die Qualität der Produkte und die Produktivität der Prozesse durch ein organisationsweites Metrikprogramm quantitativ gemessen. Das Management hat dadurch eine objektive Basis, um Entscheidungen treffen zu können. In Stufe fünf „optimizing“ schließlich werden ständig neue und verbesserte Wege der Softwareentwicklung erprobt. Die gesamte Organisation ist auf kontinuierliche Prozessverbesserung eingestellt.

Neben dem ursprünglichen Software-CMM entwickelten sich seit 1991 noch weitere CMM-Derivate wie „Systems Engineering CMM“ oder „People-CMM“ und andere mehr. Da diese Derivate sich überlappen und z. T. sogar widersprechen entstand CMMI. CMMI steht für „Capability Maturity Model Integration“ und integriert „Software Engineering CMM“, „Systems Engineering CMM“ und „Integrated Product and Process Development“ zu einem Reifegradmodell. Zusätzliches Ziel ist die Kompatibilität mit ISO/IEC 15504 „SPICE“.

SPICE steht für „Software Process Improvement and Capability Determination“ und ist in ISO/IEC 15504 festgelegt. Im Gegensatz zu CMM ist es eine internationale Initiative zur Förderung der Entwicklung eines internationalen Standards für Software-Prozess-Bewertung.

Sowohl SPICE als auch CMM basieren auf dem Softwareprozesmodell nach ISO 12207 „Information technology — Software life cycle processes“ (vgl. [ISO95]).

3.7. Beschreibungssprachen

Zur Beschreibung von Funktionen, Systemen und konkret von Software haben sich verschiedene Methoden und Beschreibungssprachen entwickelt. Die wichtigsten im Zusammenhang mit Echtzeitsystemen und dem Automobilbereich werden im Folgenden vorgestellt.

3.7.1. SA/RT

Die Abkürzung SA/RT steht für „Structured Analysis/Real Time Analysis“, zu deutsch „Strukturierte Analyse/Echtzeitanalyse“. Nach [Bal01] basiert SA/RT dabei auf der Methode „Strukturierte Analyse“ (SA), die 1979 von Tom DeMarco in [DeM79] beschrieben wurde, und wurde von Ward und Mellor 1985 in [WM85] und von Hatley und Pirbal 1987 in [HP87] vorgestellt.

Die Grundelemente der Strukturierten Analyse sind Datenflussdiagramme, die mit Hilfe von hierarchischen Regeln die Struktur eines Systems beschreiben. Auf der obersten Ebene beschreibt das sogenannte Kontextdiagramm (ein Beispiel zeigt Abbildung 3.16) die Schnittstellen des zu modellierenden Systems mit seiner Umwelt, auf der untersten Ebene wird eine Funktionalität mit Hilfe von sogenannten MiniSpecs (textuelle Beschreibungen oder Pseudo-Code) dargestellt.

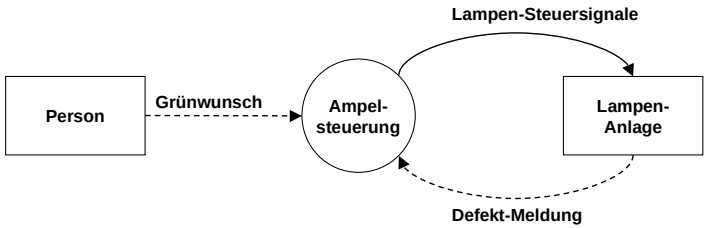

Abbildung 3.16.: Beispiel für ein Kontextdiagramm: Fußgängerampel

SA/RT erweitert diese Datenfluss-orientierte Darstellung um Kontrollflüsse, die Signale darstellen, die im System eine steuernde Funktion übernehmen. In sogenannten Flussdiagrammen (Beispiel in Abbildung 3.17 auf der nächsten Seite), in denen die Daten- und Kontrollflüsse modelliert werden, werden Datenflüsse mit Hilfe von durchgezogenen Linien, Kontrollflüsse durch gestrichelte Linien dargestellt. Neben den aus der SA bekannten MiniSpecs sind auch Zustandsübergangsdiagramme zur Modellierung vorgesehen.

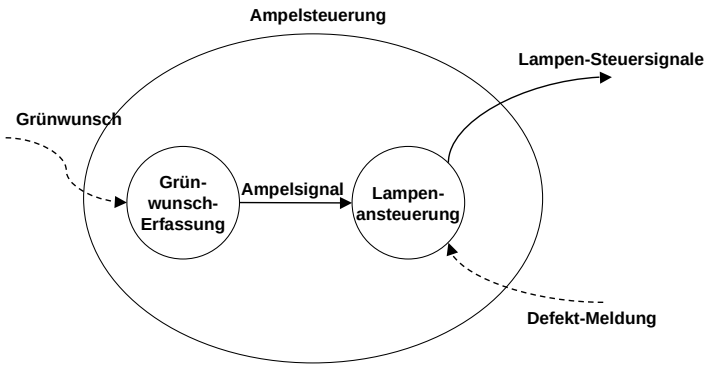

Abbildung 3.17.: Beispiel für Flussdiagramm: Ampelsteuerung

Die SA ist mittlerweile weitgehend durch die Objektorientierte Analyse (OOA) mit ihrer Notationssprache UML[8] verdrängt worden. SA/RT ist heute (Frühjahr 2004) noch immer der Stand der Technik v. a. im Automobilbereich, wird wohl aber in der Zukunft auch durch die UML (dann in der Version 2.0) abgelöst werden.

3.7.2. COMET

Die „Concurrent Object Modelling and Architectural Design Method COMET“ (vgl. [Gom00]) ist eine Weiterentwicklung von SA/RT nach Ward/Mellor und stellt eine Entwicklungsmethodik für verteilte Echtzeitsysteme dar. Sie wurde von Prof. Hassan Gomaa an der George Mason University, in den USA entwickelt.

Zur Darstellung wird dabei die UML verwendet. COMET teilt sich auf in eine Anforderungsphase und eine Designphase, dabei wird das System bis auf Taskebene entworfen und dargestellt. Es fehlt aber der Übergang zur Implementierung.

3.7.3. UML-RT

Zur Zeit wird eine Erweiterung der objektorientierten Modellierungssprache „Unified Modeling Language UML“ diskutiert, um Echtzeitsysteme darstellen zu können. „UML for Real-Time“ (UML-RT) basiert auf Entwicklungen wie ROOM

[8] UML: Unified Modeling Language

(„Real-Time Object-Oriented Modeling“, vgl. [SGW94]) und stellt bessere Notationsmittel für die Darstellung von Echtzeitsystemen zur Verfügung. Es wird erwartet, dass sie in den neuen Standard UML 2.0 (siehe dazu [OMG04]) einfließen wird.

3.7.4. CARTRONIC

CARTRONIC ist ein Strukturierungskonzept für alle Steuerungs- und Regelungssysteme eines Kraftfahrzeugs zur Erstellung einer abgestimmten, fahrzeugweiten Funktionsstruktur. Dabei wird das betrachtete System nach funktionalen Gesichtspunkten gegliedert, um eine realisierungsunabhängige Basis für die Systementwicklung zu schaffen. Ausgehend von dieser Gliederung kann unter Verwendung der UML-Notation ein Entwurf des Systems erfolgen. Das Strukturierungskonzept legt fest, aus welchen Strukturelementen eine CARTRONIC-Funktionsstruktur bestehen kann und enthält Regeln, wie die Strukturelemente miteinander in Beziehung gesetzt werden können (vgl. [LTS+01]).

Dabei wird das System auf rein funktionaler Ebene beschrieben. Es gibt Informationsgeber, Koordinatoren und sogenannte operative Komponenten. Darüber hinaus wird das System hierarchisch gegliedert. Auch die Kommunikation der funktionalen Bestandteile untereinander wird dargestellt, es gibt Abfragen, Anforderungen und Aufträge.

In [BDM99] wurden Elemente von CARTRONIC um die Betrachtung der Sicherheit erweitert. Unter anderem findet sich dabei ein Konzept zur Gefährdungsanalyse von Automobilsystemen.

Abbildung 3.18 auf der nächsten Seite zeigt als Beispielsystem einen Fahrzeugantrieb in CARTRONIC-Darstellung.

3.8. Aktuelle Arbeiten

In diesem Abschnitt werden aktuelle Arbeiten im Zusammenhang mit der Berücksichtigung von Sicherheit und Zuverlässigkeit bei der Systementwicklung im Automobilbereich vorgestellt.

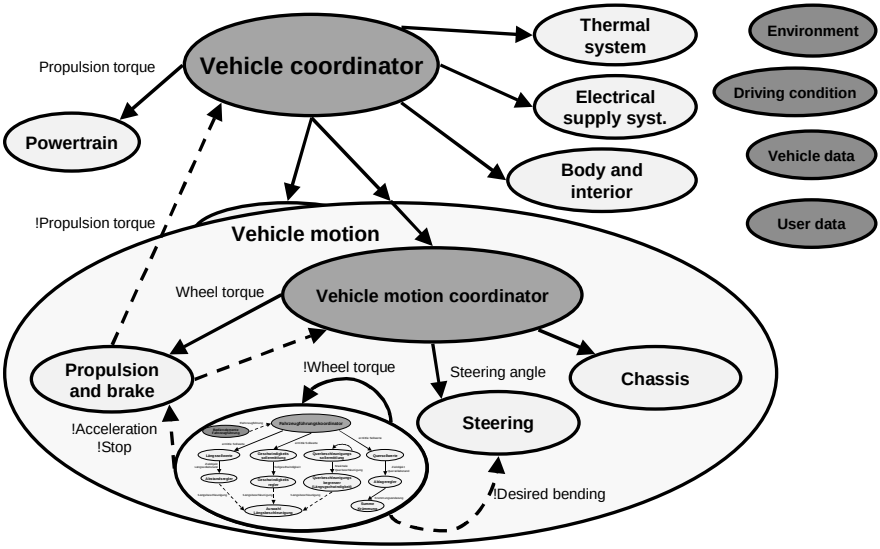

Abbildung 3.18.: Ein Fahrzeugantrieb, nach CARTRONIC strukturiert. Quelle: Bosch

3.8.1. Aktuelle Forschungsarbeiten

3.8.1.1. Forschungsarbeit von Lenk

In seiner Dissertation mit dem Titel „Zuverlässigkeitsanalyse von komplexen Systemen am Beispiel Pkw-Automatikgetriebe“ (siehe [Len95]), entstanden bei der BMW AG in München in Kooperation mit der Universität Stuttgart, stellt Robert Lenk eine Methodik zur Zuverlässigkeitsanalyse vor. Der Fokus liegt dabei allein auf der Betrachtung von Zuverlässigkeit, Sicherheit bleibt außen vor. Als Beispielsystem dient eine elektronische Getriebesteuerung.

3.8.1.2. Forschungsarbeit von Pauli

Bernhard Pauli stellt in seiner Dissertation „Zuverlässigkeitsprognosen für elektrische Steuergeräte im Kraftfahrzeug: Modellbildungen und deren praktische Anwendungen“ (vgl. [Pau97]) einige neue Verfahren zur Ermittlung von Zuverlässigkeitskennzahlen vor. Die Arbeit entstand an der Bergischen Universität-Gesamthochschule Wuppertal in Kooperation mit der Robert Bosch GmbH.

3.8.1.3. Forschungsarbeit von Mahmoud

Rachad Mahmoud promovierte an der Universität Siegen in Kooperation mit der Daimler-Benz Forschung in Stuttgart. In seiner Arbeit „Sicherheits- und Verfügbarkeitsanalyse komplexer Kfz-Systeme“ (vgl. [Mah00]) stellt er die Verbindung von Fehlerbaum- und Markov-Analyse zu einem verketteten Konzept vor, um die Vorzüge beider Konzepte nutzen zu können.

3.8.1.4. Forschungsarbeit von Stölzl

Stefan Stölzl stellte in seiner Dissertation „Fehlertolerante Pedaleinheit für ein elektromechanisches Bremssystem (Brake-by-Wire)“ (vgl. [Stö00]) eine Vorgehensweise zur Systementwicklung basierend auf fünf Entwurfskriterien Zuverlässigkeit, Verfügbarkeit, Instandhaltbarkeit, Sicherheit und Kosten vor. Die Vorgehensweise ist in Abbildung 3.19 dargestellt.

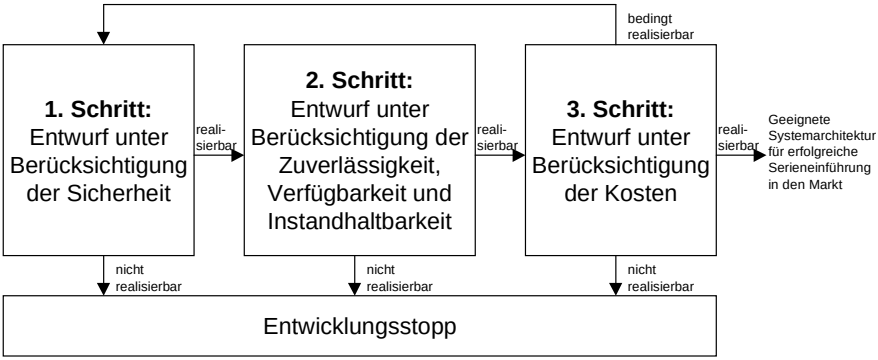

Abbildung 3.19.: Vorgehensweise beim Entwicklungsprinzip nach Stölzl

3.8.1.5. Forschungsarbeit von Hedenetz

Bernd Hedenetz beschreibt in seiner Dissertation „Entwurf von verteilten, fehlertoleranten Elektronikarchitekturen in Kraftfahrzeugen“ (vgl. [Hed01]) einen Entwicklungszyklus, bestehend aus mehreren iterativen V-Prozessen: Funktionsprototyping, Funktionsentwicklung, Funktionsintegration und Reifegradentwicklung. Diese V-Prozesse basieren auf einem Referenz-V-Prozess mit den Phasen

Definition der Anforderungen, Globaler Entwurf, Lokaler Entwurf, Realisierung und Verifikation/Validierung.

In [Hed01] werden zudem vier Erweiterungen für den Entwicklungsprozess im Automobil vorgestellt, die das V-Modell zu einem geschlossenen Ansatz erweitern. Im einzelnen sind dies Simulationsmodelle für die Entwicklung von verteilten Regelungen unter Berücksichtigung der Kommunikationsbeziehungen, Integration der WCET[9]-Analyse in den Entwicklungsprozess, um Aussagen über Prozesslaufzeiten für ein optimiertes Prozessscheduling treffen zu können und die Synchronisation der verteilten Prozesse zu gewährleisten. Dann eine Validierung der Fehlertoleranzeigenschaften des Architekturentwurfs durch Fehlerinjektion in Modelle des Systems und das reale System und die Integration der Zuverlässigkeitsanalyse in den Entwicklungsprozess durch Generierung von Fehlerbäumen aus Simulationsmodellen.

3.8.1.6. Forschungsarbeit von Binfet-Kull

Maria Binfet-Kull promovierte an der Bergischen Universität-Gesamthochschule Wuppertal, ihre Arbeit entstand aber größtenteils bei der Volkswagen AG. Sie untersucht in ihrer Arbeit „Entwicklung einer Steer-by-Wire-Architektur nach zuverlässigkeits- und sicherheitstechnischen Vorgaben“ (vgl. [Bin01]) verschiedene Alternativen für eine Steer-by-Wire Systemarchitektur.

Dabei betrachtet sie auch Elemente aus dem Luftfahrtbereich, konkret eine Gefährdungsanalyse nach SAE ARP 4761.

3.8.2. Forschungsprojekte

3.8.2.1. Das EU-Projekt „X-by-Wire“

Ziel dieses europäischen Forschungsprojekts aus den Jahren 1996 bis 1999 war, einen ersten Rahmen für die Einführung von Systemen mit sehr hohen Sicherheitsanforderungen, namentlich „X-by-Wire Systemen“, in Fahrzeugen abzustecken. Mehrere europäische Firmen und Universitäten arbeiteten an diesem Projekt mit dem Ziel, eine Aufsplitterung in verschiedene konkurrierende Standards zu vermeiden, (vgl. [DMF], [Kop97] und [LTS+01]).

[9] WCET: Worst Case Execution Time

3.8.2.2. Das EU-Projekt „SETTA“

Das Ziel des EU-Projektes SETTA „Systems Engineering for Time-Triggered Architectures“ (vgl. [SET04]) aus den Jahren 2000 bis 2002 war, den Systementwicklungsprozess der Time Triggered Architecture mit dem zeitgesteuerten Bussystem TTP/C als Grundlage voranzutreiben.

Dazu wurde im Rahmen des Projektes ein „Drei-V-Modell“ verwendet, das in einzelnen Arbeiten im Rahmen dieses Projektes ausgefüllt wurde. Abbildung 3.20 zeigt dieses Drei-V-Modell, das auf [MB97] zurückgeht.

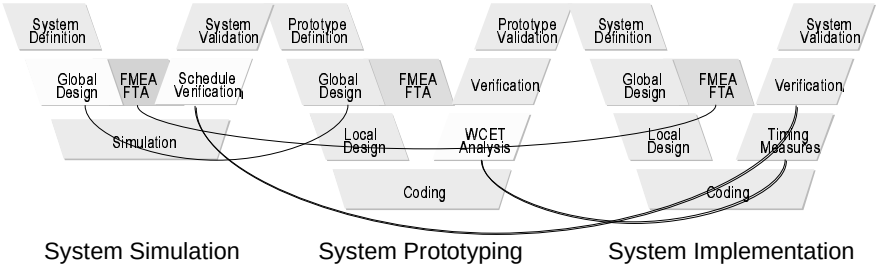

Abbildung 3.20.: Das Drei-V-Modell im SETTA-Projekt

Das Drei-V-Modell basiert darauf, dass der Entwicklungsprozess in drei große Teile untergliedert wird, die zunächst nicht miteinander verknüpft sind. Zuerst wird das zu entwickelnde System in einem ersten V simuliert, dies ist ganz links in Abbildung 3.20 dargestellt. Basierend auf der Erfahrungen aus diesem Prozessschritt folgt die Erstellung eines Systemprototypen (Mitte von Abbildung 3.20), erst im dritten V ist der eigentliche Entwicklungsprozess mit der Implementierung des Zielsystems dargestellt.

Im Rahmen des SETTA-Projektes wurden durch Werkzeuge gezielt methodische Verbindungen zwischen diesen drei Vs gezogen, diese sind in Abbildung 3.20 durch Verbindungslinien dargestellt.

3.8.2.3. Das Projekt „STEP-X“

STEP-X („Strukturierter Entwicklungsprozess für eingebettete Systeme im Automobilbereich“, Informationen z. B. unter [STE03]) ist ein Forschungsprojekt der TU Braunschweig in Kooperation mit der Volkswagen AG. Ziel des Projektes ist

es, einen durchgängigen Entwicklungsprozess von der Definition der Anforderungen bis zur automatischen Codegenerierung aufzuzeigen.

Dabei soll untersucht werden, mit welchen Methoden, Vorgehensmodellen und Werkzeugen ein durchgängiger Entwicklungsprozess für zeitgesteuerte Anwendungen geschaffen werden kann. Dazu werden beispielhaft - basierend auf dem Bussystem TTP/C - zeitgesteuerte Anwendungen im Automobil betrachtet, die aber bei bisherigen Betrachtungen ausschließlich auf die Komfortelektronik abzielen.

4. Das Konzept für eine neue Entwicklungsmethodik

In diesem Kapitel wird die Grundidee für eine Entwicklungsmethodik für sicherheitsrelevante Elektroniksysteme im Automobil skizziert, die den zukünftigen, besonderen Anforderungen Rechnung trägt.

Dafür werden in diesem Kapitel zunächst die Anforderungen an eine solche Entwicklungsmethodik dargestellt, wie sie sich aufgrund der heutigen und für die nächste Zukunft absehbaren Randbedingungen ergeben. An diesen Anforderungen wird die heutige Vorgehensweise bei der Systementwicklung im Automobilbereich gespiegelt und zusätzlich notwendige Schritte bei der Systementwicklung ermittelt. Basierend darauf wird dann eine Entwicklungsmethodik dargestellt, die diese zusätzlichen Schritte beinhaltet: das sogenannte „Zwei-V-Modell“.

Die besondere Herausforderung bei der Entwicklung sicherheitsrelevanter Elektroniksysteme im Automobilbereich ist das Erbringen eines Sicherheitsnachweises für das entwickelte System. Zusätzlich ist es wünschenswert, bereits bei der Auswahl einer Systemarchitektur a priori eine Abschätzung treffen zu können, ob die Sicherheitsanforderungen des Systems durch die entsprechende Systemarchitektur erfüllt werden können.

4.1. Anforderungen an eine Entwicklungsmethodik

Mit sicherheitsrelevanten Systemen, wie sie im Rahmen dieser Arbeit verstanden werden (vgl. Kapitel 2.1.7), wird im Automobilbereich Neuland betreten. Es ergeben sich dabei neue Anforderungen und Randbedingungen hinsichtlich der Sicherheit eines solchen Systems, der Zuverlässigkeit seiner Komponenten, des Nachweises dieser Systemeigenschaften und daher auch hinsichtlich des methodischen Vorgehens bei der Systementwicklung. Bei der Konzeption einer Vorgehensweise für die Systementwicklung für zukünftige sicherheitsrelevante Systeme im Automobil müssen diese Anforderungen berücksichtigt werden.

Zu Beginn einer jeden Systementwicklung steht die Analyse der Systemanforderungen. Hierbei werden neben der gewünschten Systemfunktion auch alle nicht-funktionalen Randbedingungen festgelegt, wozu auch besondere Anforderungen an die Sicherheit des Systems gehören. Oft werden in dieser Phase im Rahmen von Machbarkeitsstudien mögliche Konzepte aufgezeigt, auch Rapid Prototyping kommt zum Einsatz. Wichtig ist dabei das Erfassen aller relevanten Aspekte des Systems.

Grundsätzlich muss sichergestellt sein, dass die Entwicklung des Systems so erfolgt, dass die gestellten Sicherheitsanforderungen dann auch vom fertigen Produkt erfüllt werden. Dazu müssen zu Beginn der Systementwicklung die konkreten Sicherheitsanforderungen für das System und seine Architektur festgelegt und für die weitere Verwendung aufbereitet werden. Dies kann beispielsweise im Rahmen einer Risiko- oder Gefährdungsanalyse geschehen. Zusätzlich muss definiert werden, wie die Erfüllung der Sicherheitsanforderungen im Laufe der Entwicklung und später am fertigen Produkt verifiziert werden kann.

In der Regel erfolgt nun als erster Schritt bei der Systemkonzeption ein funktionaler Entwurf des Systems, bei dem die Gesamtfunktionalität definiert und strukturiert wird.

Basierend auf dieser funktionalen Gliederung erfolgt der Systementwurf. Hier wird eine geeignete Systemarchitektur so ausgewählt, dass sie den Anforderungen des Systems Rechnung trägt. Dann erfolgt der Entwurf der Architektur und der Topologie mit der Aufteilung des Systems in Subsysteme, dann die Verteilung der Funktion in Hard- und Software.

Grundsätzlich sollte eine Entwicklungsmethodik eine Unterstützung geben, um bei der Konzeption und Ausgestaltung der Architektur des Systems sowie seiner Subsysteme, Komponenten, Hardware und Software eine optimale Lösung zu finden, so dass das System seine geforderten funktionalen und nicht-funktionalen Anforderungen erfüllt. Dazu ist eine Unterstützung bei der Auswahl entsprechender Architekturalternativen unter besonderer Berücksichtigung der Sicherheitsanforderungen notwendig. Der Entwickler sollte „a priori“ dazu in der Lage sein, eine Abschätzung treffen zu können, ob die gewählte Systemarchitektur die geforderten Sicherheitsanforderungen erfüllen wird. Dies ermöglicht eine zeit- und kostenoptimale Systementwicklung. Dafür sollten Fehlertoleranzeigenschaften des gesamten Systems, aber auch einzelner Komponenten im Voraus planbar und vorhersagbar sein, auch die Konzepte zum Erreichen des geforderten Sicherheitsniveaus wie Redundanzen oder Diversität müssen dabei berücksichtigt werden. Wünschenswert ist die Möglichkeit einer Evaluierung und Vergleichs verschiedener Entwurfsalternativen unter dem Gesichtspunkt Sicherheit.

Oberstes Ziel muss dabei bleiben, ein sicherheitsrelevantes System so zu entwickeln, dass von ihm eine möglichst kleine Gefährdung ausgeht. Dabei kommt der Vermeidung von Fehlern die größte Bedeutung zu. Falls Fehlervermeidung nicht möglich ist, muss durch Begrenzung der Fehlerauswirkung oder durch Erhöhung der Zuverlässigkeit von Bauteilen bzw. von Software erreicht werden, dass die Produkte mindestens die Sicherheit bieten, die unter Berücksichtigung aller Umstände berechtigterweise erwartet werden kann. Die Konstruktion muss die geltenden technischen Normen erfüllen und darüber hinaus dem laufend fortschreitenden Stand der Technik entsprechen. Erkenntnisse aus der Beobachtung des Marktes und der Wettbewerber sind ebenfalls zu berücksichtigen.

Der nächste Schritt der Systementwicklung, der sich an den Systementwurf anschließt, ist die Implementierung des entworfenen Systems mit Konstruktion und Codierung.

Anschließend werden die einzelnen Komponenten des Systems zu größeren Einheiten integriert und dabei verifiziert und validiert. Dabei muss der Nachweis erbracht werden, dass die Sicherheitsanforderungen vom implementierten und integrierten System auch erfüllt werden (sog. Sicherheitsnachweis, d. h. die Verifikation und Validierung der Sicherheitsanforderungen). Dies wird auch für die Typzulassung benötigt, auch hier ist entsprechende Unterstützung notwendig.

Grundsätzlich sollte sich auch eine neuartige Vorgehensweise an der bisherigen Art der Systementwicklung im Automobilbereich orientieren. Dies heißt konkret, dass das V-Modell ’97 als Grundlage dienen sollte, da dieses im Automobilbereich eingeführt und erprobt ist. Besondere automobilspezifische Randbedingungen (vgl. Kapitel 3.3) müssen u. U. zusätzlich berücksichtigt werden. Allerdings werden wahrscheinlich Systemeigenschaften wie z. B. Variantenvielfalt bei zukünftigen sicherheitsrelevanten Systemen aus Sicherheitsgründen so nicht mehr zu erwarten sein.

4.2. Vergleich mit dem Stand der Technik

Die Systementwicklung im Automobilbereich ist heute hauptsächlich von der Darstellung einer bestimmten Systemfunktion getrieben. Andere Randbedingungen, die nicht direkt die Systemfunktion betreffen, spielen dabei auch eine wichtige, allerdings untergeordnete Rolle (wie beispielsweise Diagnose). Ein großer Schwerpunkt bei der Systementwicklung liegt auf der Zuverlässigkeit und Qualität des zu entwickelnden Systems, die Vorgehensweisen hier sind größtenteils sehr ausgefeilt und erprobt.

Die systematische Aufbereitung von Sicherheitsanforderungen und ihre methodische Berücksichtigung in einer Systemarchitektur findet sich dagegen seltener. Wohl finden sich auch im V-Modell ’97 etliche Elemente bzgl. Sicherheit (vgl. z. B. [IAB02b]), doch waren bisher aufwändige Sicherheitsbetrachtungen meist nicht notwendig, weswegen hier im Automobilbereich noch nicht sehr viel Erfahrung vorhanden ist. Gerade bei der Festlegung von Sicherheitsanforderungen an das zu entwickelnde System („wie sicher muss es sein?“) gibt es noch kein systematisches Konzept. Auch Methoden zur Ermittlung dieser Anforderungen waren bisher nicht gefordert.

Ein weiteres großes Potenzial ist bei der Auswahl von Architekturalternativen unter Berücksichtigung der Sicherheitsanforderungen zu sehen. Gerade die Möglichkeit, a priori eine Abschätzung treffen zu können, ob die gewählte Systemarchitektur die gestellten Anforderungen erfüllen wird, kann sehr wertvoll sein. Diese Abschätzung kann mit Hilfe von quantitativen oder qualitativen Maßnahmen erfolgen. Während der gesamten Systementwicklung ist eine Überprüfung der Erfüllung aller Sicherheitsanforderungen notwendig.

Allerdings sind gerade bei neuen Systemen oftmals nicht ausreichend Daten für eine entsprechende quantitative Analyse vorhanden. Hier müssen entsprechende Datenbanken aufgebaut werden, eine Abhilfe können auch Konzepte wie Entwurfsmuster sein.

Die Problemstellung, wie man die Sicherheits- und Zuverlässigkeitseigenschaften von Software beurteilen kann, ist aktuell noch ungelöst (siehe dazu auch [Lev95]). Gerade auch im Automobilbereich, wo der Anteil an Software bei der Darstellung einer Funktion stetig wächst, muss daher ein geeignetes Konzept gefunden werden.

Der Sicherheitsnachweis gegen Ende der Systementwicklung ist ein sehr wichtiger, ebenfalls teilweise noch ungelöster Schritt. Die Verifikation und Validierung der Erfüllung der Sicherheitsanforderungen ist gerade für eine Unterstützung der neugeregelten Typzulassung nach UN ECE R 79 oder UN ECE R 13 (vgl. [UN 01b] bzw. [UN 01a]) unabdingbar.

Die Notwendigkeit für eine entsprechende Anpassung der Systementwicklung wird auch durch mehrere weitere Aktivitäten auf diesem Gebiet im Automobilbereich unterstrichen. So veröffentlichte eine Arbeitsgruppe der Delphi Corporation Vorschläge für einen entsprechenden Sicherheitsprozess im Automobilbereich auf Basis des MIL-STD-882 (vgl. [ADM+00] und [CDJM03] sowie Kapitel 3.5.3). Auch bei der BMW AG gibt es Aktivitäten, welche die funktionale Sicherheit von Automobilelektronik basierend auf der Norm DIN EN 61508 im Fokus haben (vgl. [JW03]). Diese Aktivitäten sind v. a. dadurch getrieben, dass die Norm

DIN EN 61508 in Ermangelung eines verbindlichen, allgemeingültigen Entwicklungsstandards im Automobilbereich einen normativen Status erlangt hat.

Die Arbeiten bei Delphi bzw. bei der BMW AG orientieren sich an Konzepten aus dem militärischen Bereich bzw. aus der Prozessautomatisierung. Die vorliegende Arbeit greift dagegen Konzepte aus der Systementwicklung im Luftfahrtbereich auf.

Nachfolgend wird eine Entwicklungsmethodik vorgestellt, die auf einer Verflechtung von V-Modell ’97 und SAE ARP 4761 beruht.

4.3. Die Grundidee: Das „Zwei-V-Modell“

Wie schon erwähnt bildet das V-Modell ’97 die Grundlage für die in dieser Arbeit vorgestellte Methodik. Sie wird durch einzelne Elemente aus SAE ARP 4761 erweitert, die entsprechend an die Vorgehensweise nach dem V-Modell ’97 angepasst wurden. Im Einzelnen sind dies eine Erweiterung der Anforderungsanalyse um die systematische Erfassung der Sicherheitsanforderungen im Rahmen einer Risiko- und Gefährdungsanalyse, entsprechende Schritte bei der Systementwicklung, die eine a-priori-Bewertung möglicher Architekturalternativen ermöglichen und dann auch eine entsprechende Hilfestellung beim Sicherheitsnachweis geben.

Grundsätzlich kann man die entstandene Methodik als „Zwei-V-Modell“ verstehen. Anschaulich gesehen besteht es aus zwei verschränkten Vs, von denen das erste V die Betrachtung der Funktion und das zweite V die Betrachtung der Systemsicherheit zum Kern hat. Die Grundlage für die Elemente im Funktions-V bildet das V-Modell ’97, die Basis für die Elemente des Sicherheits-V der Standard SAE ARP 4761. Die Betrachtung von Sicherheit und Funktion ist analog zur Vorgehensweise aus SAE ARP 4754 getrennt (vgl. Abbildung 3.13 auf Seite 56). Die Grundidee für das Zwei-V-Modell ist in Abbildung 4.1 auf der nächsten Seite graphisch dargestellt.

Für diese Art der Darstellung als zwei getrennte, aber verflochtene Vs gibt es mehrere Gründe: Die Vorgehensweise, wie sie im V-Modell beschrieben ist und im Automobilbereich entsprechend mit zusätzlichen Ergänzungen hinsichtlich Qualität und Zuverlässigkeit umgesetzt ist, wird als Grundlage für die Entwicklung zukünftiger sicherheitsrelevanter Elektroniksysteme so nicht mehr ausreichen.

Die Systemsicherheit muss stärker als bisher bei der Entwicklung des entsprechenden Systems im Vordergrund stehen und daher im Entwicklungsprozess entsprechend berücksichtigt werden. Charakteristisch ist dabei, dass die zusätzlichen Elemente eine gleichwertige Bedeutung wie die bisherigen Elemente auf Basis

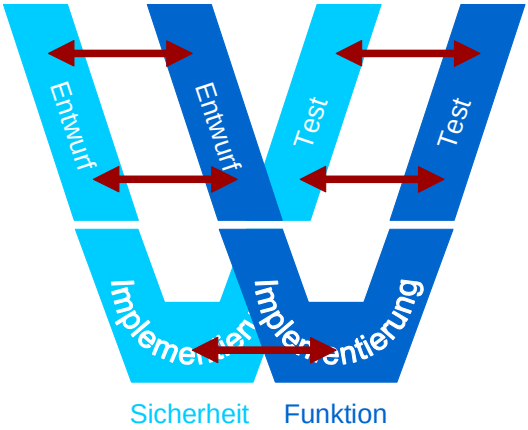

Abbildung 4.1.: Grundidee für das Zwei-V-Modell

des V-Modell ’97 haben müssen. Lag der bisherige Fokus bei der Entwicklung im Wesentlichen auf der Funktion des Systems, so muss zukünftig Funktion und Sicherheit gleichwertig betrachtet werden. Man könnte daher auch den Begriff „Funktion-Sicherheit Co-Design“ verwenden. Sicherheit ist eine Eigenschaft des gesamten Systems und muss deswegen auch im gesamten System berücksichtigt werden. Daher muss es auch im gesamten Entwicklungsprozess eine entsprechende Rolle spielen. Dies soll durch die zwei gleichwertigen Vs verdeutlicht werden.

Andererseits spricht für die Beibehaltung der Grundstruktur des V-Modells, dass dieses im Automobilbereich eingeführt, erprobt und bewährt ist (z. B. in Form von Unterstützung durch entsprechende Software-Werkzeuge). Es eignet sich gut für große Projekte wie elektronische Systeme im Automobil und hat in besonderer Art und Weise den im Automobilbereich sehr intensiv verwandten Schwerpunkt auf dem System-Test.

Die Darstellung der in dieser Arbeit entstandenen Entwicklungsmethodik in Abbildung 4.2 auf der nächsten Seite zeigt neun miteinander verzahnte Prozess-Schritte. Dabei sind die Schritte des äußeren V (Fokus „Funktion“) an das V-Modell ’97 angelehnt, die des inneren V (Fokus „Sicherheit“) an die Norm SAE ARP 4761.

Im ersten Schritt *System-Anforderungsanalyse* werden die Systemanforderungen erfasst, gebündelt und aufbereitet. Basierend darauf wird das System im *Funktionalen Entwurf* auf rein funktionaler Ebene skizziert, parallel erfolgt basierend darauf in der *Funktionalen Gefährdungsanalyse* eine Ermittlung und Bewertung der

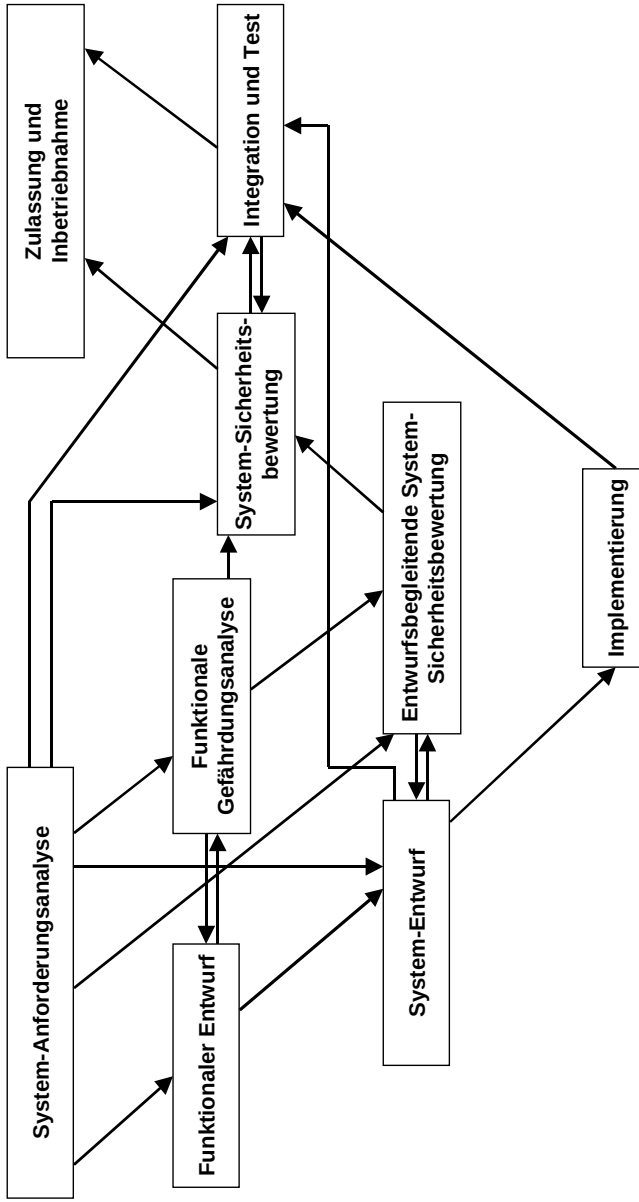

Abbildung 4.2.: Die vorgeschlagene Entwicklungsmethodik im Überblick

Sicherheitsanforderungen. Diese werden den funktionalen Einheiten entsprechend zugeordnet.

Auf Basis der entstandenen funktionalen Beschreibung wird das System inkrementell im Rahmen des *System-Entwurfs* entwickelt und entworfen, parallel erfolgt in der *Entwurfsbegleitenden Systems-Sicherheitsbewertung* eine Abschätzung, ob das System die Sicherheitsanforderungen erfüllen wird. Grundlage bei der Auswahl von Architekturalternativen ist dabei nicht die Vorgabe von konkreten Architekturkonzepten durch die Methodik selbst (wie sie z. B. in [Rei98] vorgeschlagen werden), sondern die Bereitstellung von entsprechenden Methoden zur Bewertung unterschiedlicher Architekturalternativen.

Die entstandenen Komponenten des Systems werden dann im darauf folgenden Schritt *Implementierung* erstellt und in *Integration und Test* zu größeren Einheiten integriert, diese dann getestet und verifiziert. Parallel erfolgt bei der *System-Sicherheitsbewertung* der Nachweis der Erfüllung der Sicherheitsanforderungen an das System. Am Ende steht dann die *Zulassung und Inbetriebnahme* des Systems.

Die Verbindungspfeile zwischen den Prozess-Schritten in Abbildung 4.2 stellen einen Fluss von Informationen dar, parallele doppelte Pfeile stehen für einen interaktiven Informationsaustausch zwischen zwei parallel durchgeführten Schritten der Entwicklungsmethodik.

Prinzipiell könnte man einwenden, dass etliche Elemente der vorgestellten Entwicklungsmethodik im Automobilbereich heute bereits in gewisser Form bei der Systementwicklung eingesetzt werden. Dies geschieht allerdings noch nicht durchgängig und nicht immer systematisch. In dieser Arbeit werden die einzelnen Schritte durchgängig und vernetzt zu einer umfassenden systematischen Entwicklungsmethodik zusammengefasst.

Im nächsten Kapitel findet sich eine detaillierte Darstellung aller neun Prozess-Schritte.

5. Die Entwicklungsmethodik im Detail

In diesem Kapitel werden die Elemente der in Kapitel 4 skizzierten Entwicklungsmethodik detailliert vorgestellt und erläutert.

In Abbildung 4.2 auf Seite 77 ist die Entwicklungsmethodik in ihrer Übersicht dargestellt. Diese Darstellung wird nun im Weiteren verfeinert, die Abstraktionsebene der Darstellung orientiert sich dabei an der des Submodells Systemerstellung im V-Modell ’97 nach [IAB02a].

5.1. SA - System-Anforderungsanalyse

Im Folgenden wird eine Vorgehensweise für die *System-Anforderungsanalyse* vorgestellt. Dabei sollte beachtet werden, dass diese Arbeit nicht alle Aspekte einer Anforderungsanalyse im Detail beleuchten kann. Für weitergehende Informationen sei auf die Literatur, beispielsweise [Bis90], verwiesen.

Ausgangspunkt einer jeden Systementwicklung sind Vorstellungen über eine bestimmte Funktion oder auch konkrete Gedanken über die Struktur eines Systems. So ist die Idee für eine neue Systemfunktionalität oft schon ausformuliert oder es liegen bereits Prototypen des Systems vor, aus denen es entwickelt werden soll. In anderen Fällen ist ein neues System eine Weiterentwicklung eines bestehenden Systems, es kommen Funktionen hinzu oder fallen weg, oder es ändern sich die Randbedingungen des Systems.

Die Systemanforderungen kommen dabei vom Kunden, basieren auf eigenen Wünschen oder auf Anforderungen einer weiteren, dritten Kategorie. Diese weiteren Anforderungen sind beispielsweise technische Randbedingungen (wie z. B. Baugröße, Schnittstellen, Plattform, etc.), die durch das Zusammenspiel mit anderen Systemen bedingt sind, oder externe, regulative Vorgaben durch Gesetze, Normen oder Zulassungsvorschriften.

Sämtliche Systemanforderungen lassen sich in sogenannte funktionale Anforderungen (alle Anforderungen, die direkt die Funktion des Systems betreffen)

und nicht-funktionale Anforderungen (im Wesentlichen alle anderen Anforderungen) aufgliedern. Zu den nicht-funktionalen Anforderungen gehören beispielsweise Kostenvorgaben, Randbedingungen bzgl. Sicherheit oder Zuverlässigkeit, Anforderungen an die Wartbarkeit des Systems, System-Randbedingungen wie die Verwendung einer Plattform, Peripherie, Schnittstellen oder auch mechanische Vorgaben.

In der *System-Anforderungsanalyse* erfolgt die Erfassung dieser Systemanforderungen und deren systematische Aufbereitung. Dabei werden noch keine Ansätze für konkrete Implementierungen des Systems berücksichtigt.

Mögliche Bedrohungen und Risiken durch das System und die sich daraus ergebenden Sicherheitsanforderungen werden nicht hier in der *System-Anforderungsanalyse* analysiert, dies erfolgt in der *Funktionalen Gefährdungsanalyse* (vgl. Kapitel 5.3).

In einem ersten Schritt beinhaltet die *System-Anforderungsanalyse* die Aufnahme und Analyse des Ist-Zustands. Hierbei wird der Stand der Technik im Zusammenhang mit dem zu entwickelnden System untersucht, gleiches gilt für die Randbedingungen des Systems. Es wird v. a. der Frage nachgegangen, ob besondere Randbedingungen existieren, die im speziellen Fall zu berücksichtigen sind. Falls es bereits Vorgänger-Systeme oder Systeme in einem ähnlichen Einsatzgebiet gibt, wird untersucht, ob und wie sich Elemente daraus im neuen System weiterverwenden lassen.

Der nächste Schritt ist die Spezifikation des Systems im Rahmen einer Systembeschreibung. Hier wird die prinzipielle Systemfunktion, die Randbedingungen sowie die nicht-funktionalen Anforderungen erfasst, festgelegt und dokumentiert. Diese Spezifikation bildet die Grundlage für alle weiteren Verfeinerungen und Ergänzungen der Anwenderforderungen in nachfolgenden Entwicklungsschritten. Hauptgrundlage sind dabei die Kundenanforderungen, in der Regel wird die Systemspezifikation auch in Kooperation mit dem Kunden erstellt. In weiteren Schritten sind die technischen, organisatorischen und auch weitere Randbedingungen wie Qualitätsanforderungen bei der Systemerstellung und dem Betrieb des Systems festzuhalten, soweit dies zu diesem Zeitpunkt bereits möglich ist.

Ein wichtiger, darauf aufbauender Schritt ist die Machbarkeitsanalyse. Hier wird untersucht, ob auf Basis der gestellten Anforderungen das System technisch und auch wirtschaftlich prinzipiell realisiert werden kann. Ist die Entwicklung des Systems unter den gegebenen Randbedingungen nicht möglich, so müssen die Anforderungen überarbeitet werden bzw. notfalls muss das Projekt gestoppt werden.

Der letzte Schritt im Rahmen der *System-Anforderungsanalyse* ist die Festlegung, wie die Validierung der Systemanforderungen im späteren Prozess-Schritt *Integration und Test* (vgl. Kapitel 5.7) zu erfolgen hat.

Alle in den einzelnen Schritten der *System-Anforderungsanalyse* ermittelten Daten sollten in einer Art und Weise erfasst und gespeichert werden, so dass sie in den folgenden Schritten einfach wiederverwendet werden können. Daher bieten sich hier Werkzeuge zum Anforderungsmanagement (wie z. B. DOORS, vgl. [Tel03]) oder die Verwendung eines geeigneten Datenformates (wie z. B. XML[1]) an. Dabei ist besonders der Gefahr zu begegnen, dass eine rein textuelle Erfassung missverständlich sein kann. Die Verwendung von UML-Konstrukten oder von ausführbaren Spezifikationen können eine Alternative sein. Im Rahmen des MSR-Projektes[2] wird eine Standardisierung von Spezifikationen und Simulationsmodellen für Elektroniksysteme in der Automobilindustrie für den Austausch von Daten zwischen Automobilherstellern und Systemlieferanten erarbeitet (vgl. [MSR03]).

Abbildung 5.1 auf der nächsten Seite zeigt als Ausschnitt von Abbildung 4.2 auf Seite 77 die Einordnung der *System-Anforderungsanalyse* in den Gesamtprozess. Im Gegensatz zu Abbildung 4.2 sind hier alle Beziehungen zu anderen Prozess-Schritten dargestellt, die Tabelle darunter erläutert die dabei mit den anderen Prozess-Elementen ausgetauschten Informationen. Abbildung 5.2 auf Seite 83 zeigt die Abwicklung der im Folgenden vorgestellten Einzelschritte der *System-Anforderungsanalyse*.

5.1.1. SA 1 - Durchführung einer Ist-Aufnahme und -Analyse

In diesem ersten Schritt werden Informationen über ein möglicherweise bereits vorhandenes Vorgängersystem beschafft, analysiert und dokumentiert. Die Anforderungen an das Vorgängersystem werden mit denen an das jetzige System verglichen und ausgewertet. Zudem wird analysiert, welche Erfahrungen mit ähnlichen Systemen in der eigenen Organisation vorliegen bzw. wie der Stand der Technik dazu ist.

Auf Basis dieser Erfahrung wird dann im nächsten Schritt eine Systembeschreibung erstellt.

5.1.2. SA 2 - Erstellung einer Systembeschreibung

Die Systembeschreibung ist das zentrale Element der *System-Anforderungsanalyse*. Aus Formulierungen der Anforderungen entsteht hier die Spezifikation eines Grundkonzepts für das gesamte System, welche den Rahmen für alle weiteren Verfeinerungen und Ergänzungen bildet.

[1] XML: Extensible Markup Language
[2] MSR: Manufacturer Supplier Relationship

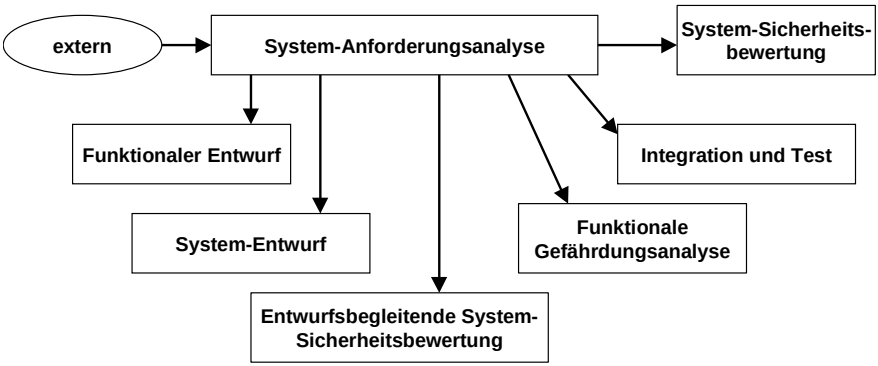

Abbildung 5.1.: Informationsfluss bei der System-Anforderungsanalyse

von	Information	nach
extern	System-Anforderungen (extern, intern, Dritte)	-
-	Funktionale Anforderungen	Funktionaler Entwurf
-	Anforderungen bzgl. Sicherheit	Funktionale Gefährdungsanalyse
-	Sonstige nicht-funktionale Anforderungen	System-Entwurf
-	Anforderungen bzgl. Sicherheit	Entwurfsbegleitende System-Sicherheitsbewertung
-	Spezifikation der Validierung	Integration und Test
-	Anforderungen bzgl. Sicherheit	System-Sicherheitsbewertung

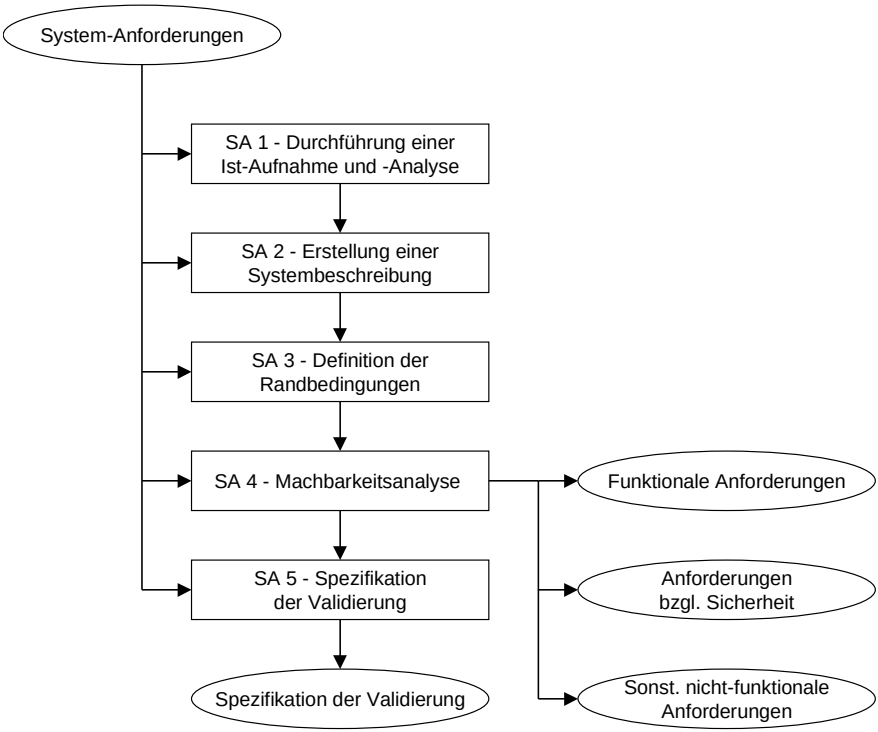

Abbildung 5.2.: Abwicklung der System-Anforderungsanalyse

Zunächst ist der Fokus auf eine funktionale Betrachtung zu legen, dabei muss klar werden, welche fachlichen Aufgaben durch das System erbracht werden sollen. In der Systembeschreibung muss erkennbar sein, welche Gesamtfunktionalität durch das System abgedeckt werden soll und welche Prioritäten dabei im weiteren Verlauf erwartet werden.

Dann sind die technischen sowie organisatorische Randbedingungen abzuleiten und in die jeweilige Organisation und das Umfeld einzuordnen. Sicherheitsbetrachtungen finden gebündelt in der *Funktionalen Gefährdungsanalyse* (vgl. Kapitel 5.3) statt, der Schwerpunkt liegt dort auf der funktionalen Sicherheit. Zusätzliche, nichtfunktionale Sicherheitsanforderungen werden hier bei der Erstellung einer Systembeschreibung erfasst.

Die Darstellung der Systembeschreibung muss so präzise erfolgen, dass im weiteren Verlauf jederzeit entscheidbar ist, wann diese verlassen wird und dementsprechend technische Entscheidungen neu überdacht werden müssen. Dazu sollte die Systembeschreibung in geeigneter elektronischer Art gespeichert werden. Zudem muss die Systembeschreibung quantitative Abschätzungen enthalten, die es ermöglichen, grundsätzliche technische Entscheidungen zu treffen. Die Verwendung erprobter Konzepte oder vordefinierter fachlicher Bausteine ist zu berücksichtigen.

5.1.3. SA 3 - Definition der Randbedingungen

Die hier zu betrachtenden Randbedingungen sind technischer und organisatorischer Art. Die Definition der technischen Randbedingungen betrifft dabei beispielsweise Anforderungen hinsichtlich der beim technischen Entwurf entstehenden Schnittstellen des Systems.

Die organisatorischen Randbedingungen werden beispielsweise durch den Kommunikations-, Kooperations- und Koordinierungsbedarf gesetzt, der bei einer sachgerechten Abwicklung der Prozesse zwischen den einzelnen Aufgabenträgern besteht.

5.1.4. SA 4 - Machbarkeitsanalyse

Die Machbarkeitsanalyse (englisch „feasibility analysis“) dient der Bewertung der aufgestellten Anwenderforderungen hinsichtlich ihrer wirtschaftlichen und technischen Realisierbarkeit und basiert auf ersten Architekturüberlegungen oder bereits auf einer konkreten, angedachten Systemarchitektur. Dabei sind die technischen und organisatorischen Rahmenbedingungen zu beachten.

Konkret werden bei einer Machbarkeitsanalyse die aufgestellten Anforderungen bezüglich ihrer Plausibilität, der Beherrschbarkeit der Komplexität, Möglichkeiten für den Einsatz von Fertigprodukten, Kostenschätzungen sowie die Machbarkeit des Systems in Bezug auf Sicherheit und Zuverlässigkeit überprüft.

Die Ergebnisse der Analyse führen gegebenenfalls zu Vorschlägen für Modifikationen der Anforderungen. Dabei sind die Vorteile, die aus einer Modifikation oder Reduzierung der ursprünglichen Anforderungen erwachsen, aufzuzeigen. Es ist sicherzustellen, dass die Modifikation bzw. Reduzierung der Anwenderforderungen die Erreichbarkeit der Projektziele nicht gefährdet. Die Machbarkeitsanalyse erfordert dabei in der Regel einen intensiven Dialog zwischen Entwicklern und Kunde.

5.1.5. SA 5 - Spezifikation der Validierung

Um die korrekte Validierung, also die Überprüfung der Erfüllung der Systemanforderungen aus der Systembeschreibung im Schritt *Integration und Test* zu gewährleisten, ist diese im Rahmen der *System-Anforderungsanalyse* zu spezifizieren.

Dabei muss nicht nur festgelegt werden, welche Anforderungen zu validieren sind, sondern auch von wem und in welcher konkreten Form die Validierung geschehen soll. Es müssen Kriterien und Methoden festgelegt werden, die schlussendlich eine Aussage darüber geben können, ob ein korrektes System entwickelt worden ist.

5.2. FE - Funktionaler Entwurf

Die Entwicklungsmethodik, die im Rahmen dieser Arbeit beschrieben wird, folgt dem Paradigma, dass die Architektur eines Systems primär von seiner Funktion bestimmt wird (auch „form follows function“ genannt). Die Systemfunktion ist daher auch der Haupttreiber beim Systementwurf. Gleichwertig, aber in Abhängigkeit von der Systemfunktion zu betrachten, ist die Sicherheit des Systems. Andere Systemeigenschaften sind diesen beiden Systemeigenschaften untergeordnet.

Der Entwurf des Systems beginnt mit dem *Funktionalen Entwurf*. Basierend auf den Ergebnissen der *System-Anforderungsanalyse* werden hier systematisch die Systemfunktionen strukturiert und beschrieben und die funktionalen Systemanforderungen festgelegt. Ausgehend von einem ersten Entwurf der Gesamtfunktion wird diese immer weiter detailliert, in Unterfunktionen gegliedert und darauf aufbauend der konzeptionelle Entwurf verfeinert. Die funktionale Architektur

wird festgelegt, es entsteht eine komplette funktionale Darstellung und eine hierarchische Gliederung des Systems.

In Kapitel 6.1 ist ein Beispiel für diese Vorgehensweise beim *Funktionalen Entwurf* und bei der parallel durchgeführten *Funktionalen Gefährdungsanalyse* dargestellt.

Während dieser Phase wird die Systemfunktion losgelöst von den Randbedingungen betrachtet, die von einer möglichen Realisierung herrühren, d. h. die physikalische Architektur und die damit verbundenen Implementierungsaspekte werden zunächst nicht berücksichtigt. Der Schritt *Funktionaler Entwurf* beinhaltet nur die Betrachtung des Ablaufs einer Funktion, von der Aufnahme eines stimulierenden Eingangssignals (z. B. Signal eines Fahrzeugsystems oder Aktion des Fahrers) über die Verarbeitung der Information bis hin zur Erzeugung einer neuen Aktion (z. B. Ansteuerung eines Aktuators oder Ausgabe von Meldungen).

Als erster Schritt werden im Rahmen des *Funktionalen Entwurfs* die Systemfunktionen einer ersten Hierarchieebene bestimmt, die dann in Form einer Funktionsliste festgehalten werden. Diese Funktionsliste dient auch als Eingangs-Information für die *Funktionale Gefährdungsanalyse*, die in enger iterativer Kopplung mit dem *Funktionalen Entwurf* erfolgt. An jeden Detaillierungsschritt schließt sich eine erneute *Funktionale Gefährdungsanalyse* für die nächste Hierarchieebene an.

Ebenso wird die Systemfunktion hierarchisch aufgegliedert, die Bestimmung der Systemfunktionen wiederholt sich auf der nächsten Abstraktionsebene. Dazu wird der *Funktionale Entwurf* auf niedrigeren Ebenen wiederholt, die Ergebnisse werden jedes Mal iterativ mit der *Funktionalen Gefährdungsanalyse* ausgetauscht.

Das Ergebnis dieser Schritte ist eine komplette funktionale Beschreibung des Systems, gepaart mit den Anforderungen an Sicherheit und Zuverlässigkeit seiner funktionalen Einheiten.

Diese Vorgehensweise wird von etlichen, zum großen Teil schon eingeführten Methoden für die funktionale Systembeschreibung unterstützt (vgl. dazu auch Kapitel 3.7). Neben der klassischen SA/RT-Methode ist auch UML, vor allem in der zukünftigen, echtzeittauglichen Spezifikation 2.0, verwendbar (z. B. in Form von „Use Case Diagrammen“). Im Automobilbereich ist gerade für die funktionale Betrachtung von Elektroniksystemen CARTRONIC entstanden (siehe Kapitel 3.7.4). In [BDM99] und auch in der Dissertation von Wolfgang Längst (vgl. [Län03]) sind Erweiterungen von CARTRONIC hinsichtlich der besonderen Berücksichtigung von Sicherheitseigenschaften beschrieben.

Grundsätzlich ist aber die Vorgehensweise beim *Funktionalen Entwurf* unabhängig von der dabei eingesetzten Methodik. Das Ergebnis ist eine detaillierte, hierarchisch gegliederte Funktionsliste, die auch bei der *Funktionalen Gefährdungs-*

analyse und beim darauf folgenden *System-Entwurf* (siehe Kapitel 5.4) verwendet wird.

Im Rahmen der Darstellung im Beispiel in Kapitel 6.1 wird exemplarisch die SA/RT-Methodik verwendet, da sie immer noch die am meisten eingesetzte Methodik für die funktionale Darstellung im Automobilbereich ist.

In Abbildung 5.3 ist die Einordnung des *Funktionalen Entwurfs* in den Gesamtprozess zu sehen, die Tabelle darunter erläutert die mit anderen Schritten ausgetauschten Informationen. In Abbildung 5.4 auf der nächsten Seite sieht man die Abfolge der einzelnen Schritte beim *Funktionalen Entwurf*.

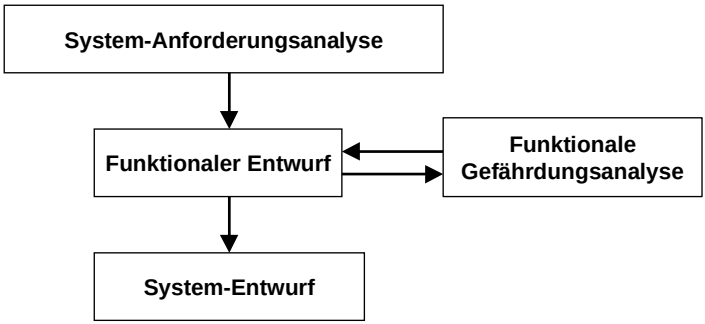

Abbildung 5.3.: Informationsfluss beim Funktionalen Entwurf

von	Information	nach
System-Anforderungsanalyse	Funktionale Anforderungen	-
-	Funktionale Beschreibung bzw. Funktionsliste	Funktionale Gefährdungsanalyse
Funktionale Gefährdungsanalyse	Gefährdungseinstufung mit Zuordnung zu funktionalen Einheiten	-
-	Komplette funktionale Beschreibung des Systems inkl. zugeordneten Sicherheitsanforderungen	System-Entwurf

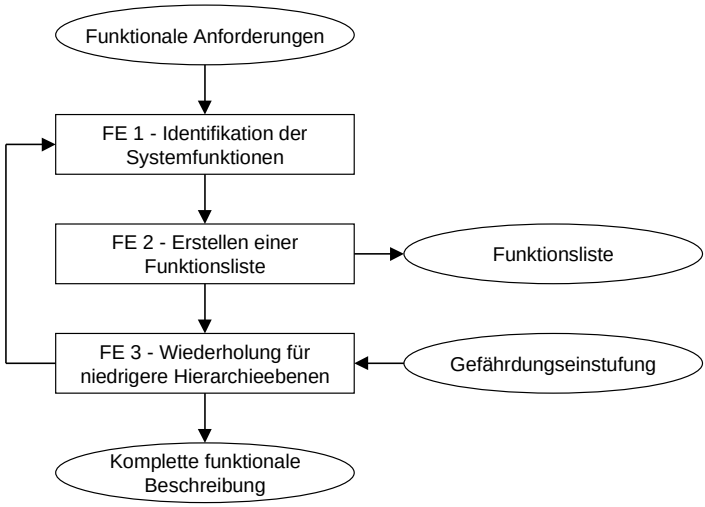

Abbildung 5.4.: Abwicklung des Funktionalen Entwurfs

5.2.1. FE 1 - Identifikation der Systemfunktionen

In einem ersten Schritt werden alle, sowohl interne wie auch die mit der Umgebung ausgetauschten Funktionen der betrachteten Hierarchieebene identifiziert. Dazu sind zunächst alle notwendigen Informationen und Daten zu beschaffen, die dann in einer geeigneten Form (z. B. SA/RT, UML, CARTRONIC, etc.) dargestellt werden.

5.2.2. FE 2 - Erstellen einer Funktionsliste

Die Funktionsliste besteht aus den einzelnen Funktionen, die im vorigen Schritt identifiziert wurden. Sie dient als Eingangsinformation für die *Funktionale Gefährdungsanalyse*. Daneben sind auch noch die verwendeten funktionalen Schnittstellen festzuhalten.

Die in der Funktionsliste dargestellten Funktionen werden später beim *System-Entwurf* (vgl. Kapitel 5.4) Hardware- bzw. Software-Einheiten zugeordnet.

5.2.3. FE 3 - Wiederholung für niedrigere Hierarchieebenen

Die Vorgehensweise in FE 1 und FE 2 wird auf niedrigeren Systemebenen wiederholt. Die identifizierten Funktionen werden weiter verfeinert. Dies geschieht bis zu einem Punkt, an dem eine weitere hierarchische Gliederung nicht mehr möglich ist.

Die Basiselemente einer solchen Gliederung werden dann mit Hilfe von Kontrollspezifikationen (sog. CSpecs) oder Mini-Spezifikationen (sog. MiniSpecs) definiert. Das Endergebnis ist dann eine komplette funktionale Systembeschreibung.

5.3. FGA - Funktionale Gefährdungsanalyse

Bei der *Funktionalen Gefährdungsanalyse* werden die funktionalen Sicherheitsanforderungen an das System ermittelt und detailliert festgelegt. Basierend auf den Ergebnissen der *System-Anforderungsanalyse* werden die Funktionen des Systems in Gefährdungsklassen eingestuft. Dazu wird das Gesamtsystem in seine Umgebung eingeordnet und die Auswirkungen des Systems auf die Umwelt und die der Umwelt auf das System analysiert.

Das Ergebnis der *Funktionalen Gefährdungsanalyse* ist eine Klassifizierung des Systems bezüglich der Gefährdungen seiner Funktionen. Durch eine sehr systematische Vorgehensweise wird eine Durchgängigkeit bei der Sicherheitsbetrachtung für die gesamte Systementwicklung sichergestellt.

Den Einstufungen der Systemfunktionen in Gefährdungsklassen können zusätzliche Zuverlässigkeitsanforderungen hinterlegt werden, deren Überprüfung ein Bestandteil eines späteren Sicherheitsnachweises sein kann. Dafür ist aber Voraussetzung, dass das System im Kontext der betrachteten Systemfunktionen nicht unmittelbar in einen allgemeingültigen sicheren Zustand als Rückfallebene bei Störungen überführbar ist (vgl. Kapitel 2.2.4). Dann und nur dann, wenn diese Voraussetzung erfüllt ist, kann man die Sicherheitsanforderungen der betreffenden Funktion direkt in Zuverlässigkeitsanforderungen umwandeln. Dies ist beispielsweise bei trockenen Brake-by-Wire oder Steer-by-Wire Systemen gegeben, da das Fahrzeug bei diesen Systemen nur dann sicher ist, wenn sie auch zuverlässig funktionieren. Im Fahrbetrieb ist kein allgemeingültiger sicherer Zustand für diese Systeme bestimmbar.

Diese Vorgehensweise hat den Vorteil, dass Zuverlässigkeit im Gegensatz zu Sicherheit quantitativ analysiert werden kann. Sie ist Standard bei der Entwicklung

von Luftfahrtsystemen, da bei einem Flugzeug während des Fluges kein sicherer Zustand bestimmbar ist (nach [Vah98]).

Allerdings muss beachtet werden, dass bei Systemen, die diese Voraussetzung nicht erfüllen, ein anderer Weg für den Nachweis der Erfüllung der Sicherheitsanforderungen gewählt werden muss. In den meisten Fällen kann ein Sicherheitsnachweis dann nicht ausschließlich quantitativ erfolgen.

Wie im parallelen Schritt *Funktionaler Entwurf* konzentriert sich die Vorgehensweise bei der *Funktionalen Gefährdungsanalyse* auf die Funktionen des Systems. Basierend auf der beim *Funktionalen Entwurf* erstellten Funktionsliste werden die zugehörigen Gefährdungen ermittelt. Abhängig von den unterschiedlichen Randbedingungen, in denen sich das System befindet, können jeder Funktion des Systems u. U. mehrere verschiedene Gefährdungen zugeordnet werden. Die Auswirkungen dieser Gefährdungen beinhalten mögliche Risiken. Zur Kategorisierung werden diese Gefährdungen nach ihrer Schwere in Gefährdungs- oder Risikoklassen eingeteilt.

Parallel zu dieser Analyse wird die funktionale Darstellung des Systems im Schritt *Funktionaler Entwurf* inkrementell verfeinert und weiter detailliert. Durch Iterationsschritte zwischen dem *Funktionalen Entwurf* und der *Funktionalen Gefährdungsanalyse* werden so Subfunktionen ebenfalls einer Gefährdungsanalyse unterzogen. Die Gefährdungsklassen werden dabei zunächst auf die Subfunktionen vererbt und dann neu bewertet. So entsteht nach und nach eine umfangreiche Darstellung der Gefährdungseinstufung mit konkreten quantitativen Anforderungen an die Zuverlässigkeit aller Funktionen und Subfunktionen.

Zur Klassifizierung der Gefährdungen in der *Funktionalen Gefährdungsanalyse* können prinzipiell verschiedenste methodische Vorgehensweisen zur Anwendung kommen. Wie in Kapitel 2.1.9 erläutert kommt dazu in der Regel ein Risikograph zur Anwendung, meist wird der Risikograph nach DIN V 19250 verwendet (vgl. Abbildung 2.2 auf Seite 16). Allerdings ist zu beachten, dass aufgrund des besonderen Zuschnitts der Norm DIN V 19250 (und auch ihrer Nachfolgenorm DIN EN 61508) auf Systeme für die Prozessautomatisierung der Risikograph dort für Automobilsysteme nur sehr bedingt geeignet ist. Sinnvoller ist eine Einstufungsmethodik, deren Parameter die Randbedingungen im Automobilbereich besonders berücksichtigen. In [BDM99] wurde im Rahmen der CARTRONIC Sicherheitsanalyse (kurz CSA) ein solcher Risikograph vorgestellt, er ist in Abbildung 5.5 auf der nächsten Seite wiedergegeben.

Die Gefährdungsklassen der CSA werden „Sicherheitsstufen“ (kurz SF) genannt. Es ergeben sich fünf Sicherheitsstufen SF 0 bis SF 4, wobei SF 4 die Stufe mit der höchsten Gefährdung ist.

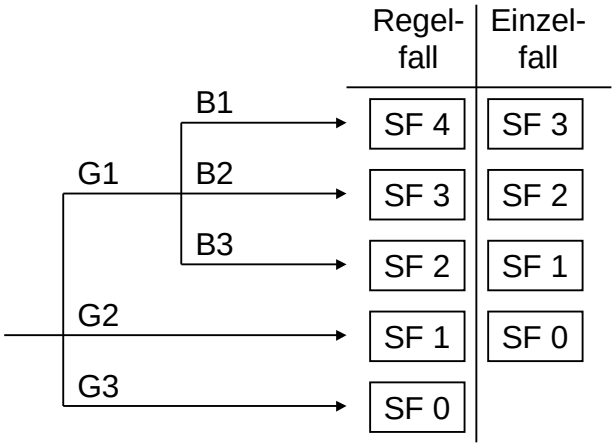

Abbildung 5.5.: Risikograph nach CSA

Legende:	
G1	unmittelbare Gefahr für Leib und Leben
G2	leichte Verletzung
G3	keine unmittelbare Gefahr
B1	nicht beherrschbar
B2	schwer beherrschbar
B3	beherrschbar
Regelfall	im Regelfall kann von einem Auftreten des Fehlers ausgegangen werden
Einzelfall	der Fehler eher selten, Auftreten nur in Einzelfällen

Quelle: [BDM99]

Der Risikodefinition gemäß Definition 2.2 auf Seite 10 wird bei der CSA noch der Parameter „Beherrschbarkeit“ hinzugefügt. Ein Fehler ist dann beherrschbar, wenn sein Einfluss auf das Fahrzeug dem entspricht oder dem vergleichbar ist, was an Situationen im normalen Straßenverkehr im Allgemeinen auch zu beherrschen ist.

Es ergibt sich so eine Risiko-Konvention mit den Parametern Schadensauswirkung (G1, G2 oder G3), Beherrschbarkeit (B1, B2 oder B3) und Häufigkeit des Auftretens (Regelfall oder Einzelfall).

Die dabei betrachteten Gefährdungen sind immer als unmittelbar zu sehen. Eine mittelbare, langfristige Gefährdung, wie sie beispielsweise durch eine Schädigung der Umwelt und eine möglicherweise dadurch verkürzte Lebenserwartung erzeugt wird, soll nicht betrachtet werden.

Eine Unterscheidung nach Anzahl der möglicherweise zu Schaden kommenden Personen erfolgt nicht, da dies aufgrund des weiten Anwendungsbereichs von Fahrzeugen nicht möglich ist. Ein Fehler, der eine schwere Verletzung oder den Tod einer Person herbeiführen kann, kann im Automobil immer auch den Tod mehrerer Personen verursachen.

Auch wird bei der Klassifikation nach CSA keine Aufenthaltsdauer im Gefahrenbereich bewertet. Diese Aufenthaltsdauer ist bei Kraftfahrzeugen immer maximal, da sich immer mindestens eine Person im Kraftfahrzeug befindet, wenn es benutzt wird.

In Tabelle 5.1 auf der nächsten Seite sind die verwendeten Sicherheitsstufen SF 0 bis SF 4 erläutert. Für eine ausführlichere Darstellung sei auf [BDM99] verwiesen.

In Abbildung 5.6 auf Seite 94 ist die Einordnung der *Funktionalen Gefährdungsanalyse* in den Gesamtprozess zu sehen, die Tabelle darunter erläutert die mit anderen Schritten ausgetauschten Informationen. In Abbildung 5.7 auf Seite 95 sieht man die Abfolge der einzelnen Schritte bei der *Funktionalen Gefährdungsanalyse*.

5.3.1. Zuordnung der quantitativen Anforderungen

Den Sicherheitsstufen sind in [BDM99] keine zugehörigen quantitativen Zuverlässigkeitsanforderungen hinterlegt. Wenn eine quantitative Analyse zum Einsatz kommen soll, so müssen diese erst noch ermittelt werden. Dies soll im Folgenden geschehen.

Dazu sei vereinbart, dass die Zuverlässigkeitsanforderungen von einer Sicherheitsstufe zur nächsten sich um jeweils eine Zehnerpotenz verändern. Damit ist für die Ermittlung der Anforderungen aller Sicherheitsklassen die Zuverlässigkeitsanforderung an nur eine Sicherheitsstufe ausreichend. Im Folgenden wird diese

Tabelle 5.1.: Beschreibung der Sicherheitsstufen SF nach CSA (gemäß [BDM99])

SF 4 Ein Fehler einer Funktion, die nach SF 4 eingestuft wurde, führt im Regelfall zu einer unmittelbaren Gefahr für Leib und Leben von Verkehrsteilnehmern. Die Fahrsituation ist nicht beherrschbar oder beeinflussbar durch die Fahrzeuginsassen.
Beispiele: keine Verzögerung des Fahrzeugs möglich; Fahrzeug nicht lenkfähig; Blockieren der Hinterachse.

SF 3 Ein Fehler in der Funktion führt im Regelfall zu einer unmittelbaren Gefahr für Leib und Leben von Verkehrsteilnehmern. Die Fahrsituation ist schwer beherrschbar oder beeinflussbar durch die Fahrzeuginsassen. In Einzelfällen führt ein Fehler in der Funktion zu einer unmittelbaren Gefahr für Leib und Leben von Verkehrsteilnehmern. Die Fahrsituation ist nicht beherrschbar oder beeinflussbar durch die Fahrzeuginsassen.
Beispiele: ungewollte Beschleunigung; keine Beschleunigung; plötzliche, einseitige Verzögerung; plötzliche Änderung der Lenkungscharakteristik; Beschleunigungsausfall bei Überholvorgang mit Gegenverkehr.

SF 2 Ein Fehler in der Funktion führt im Regelfall zu einer unmittelbaren Gefahr für Leib und Leben von Verkehrsteilnehmern. Die Fahrsituation ist beherrschbar und beeinflussbar durch die Fahrzeuginsassen. In Einzelfällen führt ein Fehler in der Funktion zu einer unmittelbaren Gefahr für Leib und Leben von Verkehrsteilnehmern. Die Fahrsituation ist schwer beherrschbar oder beeinflussbar durch die Fahrzeuginsassen.
Beispiele: schwergängige Lenkung; Entriegelung der Lenkradhöhenverstellung; geänderte Bremskraftverstärkung; defekter Scheibeneinklemmschutz.

SF 1 Ein Fehler einer nach SF 1 eingestuften Funktion führt im Regelfall maximal zu leichten Verletzungen der Verkehrsteilnehmer. In Einzelfällen führt ein Fehler in der Funktion zu einer unmittelbaren Gefahr für Leib und Leben von Verkehrsteilnehmern. Die Fahrsituation ist beherrschbar oder beeinflussbar durch die Fahrzeuginsassen.
Beispiele: defekte Sitzheizung; defekte Lenkradheizung.

SF 0 Eine unmittelbare Gefahr für Verkehrsteilnehmer kann ausgeschlossen werden.
Beispiele: Ausfall der Abgasreinigung; Ausfall von Komfortelektronikfunktionen.

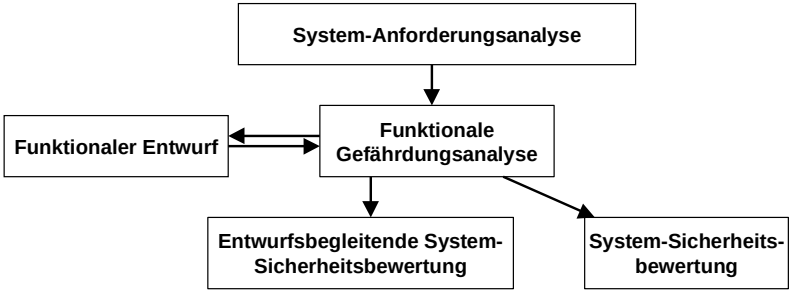

Abbildung 5.6.: Informationsfluss bei der Funktionalen Gefährdungsanalyse

von	Information	nach
System-Anforderungsanalyse	Anforderungen bzgl. Sicherheit	-
Funktionaler Entwurf	Funktionale Beschreibung bzw. Funktionsliste	-
-	Gefährdungseinstufung mit Zuordnung zu funktionalen Einheiten	Funktionaler Entwurf
-	Zugeordnete Anforderungen an Sicherheit und Zuverlässigkeit	Entwurfsbegleitende System-Sicherheitsbewertung
-	Vorgehensweise zur Verifikation dieser Anforderungen	Entwurfsbegleitende System-Sicherheitsbewertung
-	Zugeordnete Anforderungen an Sicherheit und Zuverlässigkeit	System-Sicherheitsbewertung
-	Vorgehensweise zur Verifikation dieser Anforderungen	System-Sicherheitsbewertung

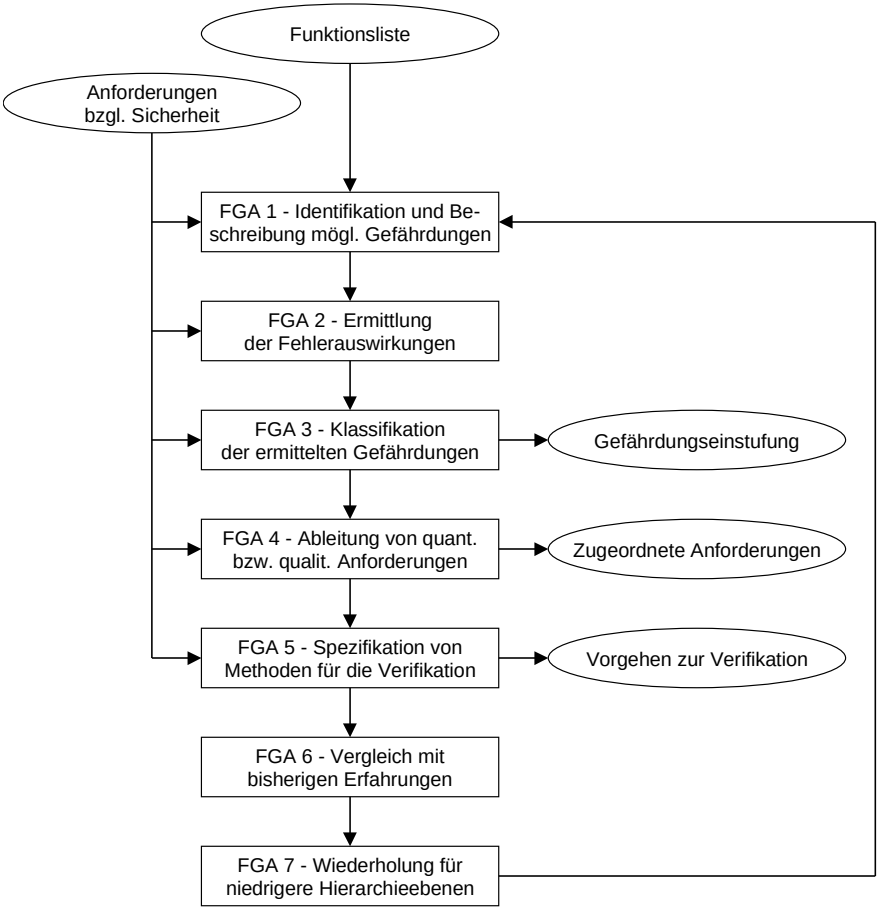

Abbildung 5.7.: Abwicklung der Funktionalen Gefährdungsanalyse

Ermittlung durch eine Abschätzung des Grenzrisikos für Systemfunktionen in der höchsten Sicherheitsstufe SF 4 durchgeführt.

Nach [Nel02] ist eine typische Anforderung eines Automobilherstellers maximal ein gefährlicher Fehler in einem sicherheitsrelevanten System in 15 Jahren Betriebsdauer bei 10 Mio. Fahrzeugen. Bei angenommenen 667 Betriebsstunden im Jahr ergibt sich so eine geforderte Rate für SF 4 von ca. $\lambda \leq 10^{-11}/h$.

In [KG02a] schlagen die Autoren aufgrund von Analogien zwischen verschiedenen Industriebereichen eine geforderte Rate von $\lambda \leq 10^{-10}/h$ vor.

Nach DIN EN 61508 sollten sicherheitsrelevante Automobilsysteme nach dem in DIN EN 61508 vorgeschlagenen Risikographen in den Sicherheitsintegritätslevel SIL 3 eingestuft werden, woraus sich eine geforderte Ausfallrate von $\lambda \leq 10^{-9}/h$ für das zugehörige sicherheitsbezogene System ergibt. Dabei wird (wie bereits bei der Abschätzung auf Seite 20) von 10 solchen Systemen ausgegangen.

Eine vierte Möglichkeit ist die Orientierung am Stand der Technik von vergleichbaren, heute schon eingesetzten Systemen. Da die Sicherheitseigenschaften solcher Systeme i. A. von der Gesellschaft toleriert sind, können sie zum Vergleich herangezogen werden. In Kapitel 2.1.9 wurde beispielhaft eine solche Analyse durchgeführt, als Ergebnis ergab sich eine maximale Fehlerrate von $\lambda \leq 10^{-8}/h$.

Diese ermittelten Anforderungen weichen sehr stark voneinander ab, die geforderte Fehlerrate λ für SF 4 bewegt sich zwischen $10^{-11}/h$ und $10^{-8}/h$.

Zunächst liegt nun nahe, sich für eine Zuordnung am Stand der Technik zu orientieren. Jedoch ist zu erwarten, dass das akzeptierte Grenzrisiko für zukünftige elektronische Systeme wohl niedriger anzusetzen ist als für heute eingesetzte mechanische Systeme. Daher erscheint eine um eine Zehnerpotenz verschärfte Anforderung sinnvoll, d. h. im Folgenden wird von einer maximalen Ausfallrate von $\lambda_{max} \leq 10^{-9}/h$ ausgegangen.

Daraus folgt für die Sicherheitsstufen SF die Einstufung nach Tabelle 5.2 auf der nächsten Seite. Für die Berechnung der maximalen Fehlerwahrscheinlichkeit P_{max} wird dabei wieder eine Systemlebensdauer von $T_{System} = 10000h$ zugrundegelegt.

Der Sicherheitsstufe SF 0 wird keine Fehlerwahrscheinlichkeit zugeordnet, da Gefährdungen, die nach SF 0 eingestuft werden, als unkritisch anzusehen sind.

Nach der Ermittlung der Gefährdungen und einer Zuordnung der quantitativen Anforderungen wird als letzter Schritt der Funktionalen Gefährdungsanalyse die Verifikation der ermittelten Sicherheitsanforderungen festgelegt. Dann werden die Ergebnisse der Gefährdungsanalyse mit früheren Erfahrungen verglichen, um eine Konsistenz mit bereits durchgeführten Systementwicklungen sicherzustellen.

Vereinzelt werden zu Beginn der Systementwicklung im Automobilbereich auch heutzutage Risikoanalysen durchgeführt. Allerdings muss gewährleistet sein, dass

Tabelle 5.2.: Zuverlässigkeitsanforderungen der Sicherheitsstufen

Sicherheitsstufe	max. Fehlerrate	max. Fehlerwahrscheinlichkeit
SF 4	$\lambda_{max,SF\ 4} \leq 10^{-9}/h$	$P_{max,SF\ 4} \leq 10^{-5}$
SF 3	$\lambda_{max,SF\ 3} \leq 10^{-8}/h$	$P_{max,SF\ 3} \leq 10^{-4}$
SF 2	$\lambda_{max,SF\ 2} \leq 10^{-7}/h$	$P_{max,SF\ 2} \leq 10^{-3}$
SF 1	$\lambda_{max,SF\ 1} \leq 10^{-6}/h$	$P_{max,SF\ 1} \leq 10^{-2}$

durch eine entsprechende systematische Vorgehensweise eine Durchgängigkeit zu den nachfolgenden Prozess-Schritten sichergestellt ist.

5.3.2. FGA 1 - Identifikation und Beschreibung möglicher Gefährdungen

In diesem Schritt werden auf Basis der identifizierten Funktionen mögliche Gefährdungen identifiziert und beschrieben, dabei werden Einfach- und Mehrfachfehler sowie Eigenschaften der Umgebung des Systems berücksichtigt.

Dazu werden auf Grundlage der in der Funktionsliste festgehaltenen Funktionen der betrachteten Hierarchieebene mögliche Fehlfunktionen identifiziert. Diese Fehlfunktionen können mögliche Gefährdungen darstellen. Jedoch ist eine Gefährdung höchst unterschiedlich zu beurteilen, je nachdem in welcher Situation sich das Fahrzeug gerade befindet.

Bei Gefährdungsanalysen im Luftfahrtbereich wird dazu zwischen verschiedenen Flugphasen unterschieden. Die äquivalenten Phasen bei einem Kraftfahrzeug (z. B. Startphase, Fahrphase und Bremsphase) machen jedoch in der Regel für das betrachtete System wenig Sinn. Daher sollte nach verschiedenen Fahrsituationen unterschieden werden, in denen sich das Fahrzeug befinden kann.

Mögliche Fahrsituation sind beispielsweise für ein Lenksystem verschiedene Szenarien wie Geradeausfahrt, Kurveneinfahrt, Kurvenfahrt, Kurvenausfahrt, Überholmanöver, verschiedene Verkehrssituationen wie Stadtverkehr, Fahrt auf der Landstraße oder Autobahn, Parken bzw. Rangieren oder Rückwärtsfahren. Hierbei kann man prinzipiell Fahrsituationen danach unterscheiden, ob dabei das Lenkrad gedreht wird oder ob es nicht gedreht wird. Weiterhin können unterschiedliche Wetterbedingungen (Regen, Schnee, Eis, Aquaplaning, Kälte, Hitze, etc.) und abhängig davon der Bodenbelag (Kopfsteinpflaster, Spurrillen, Offroad, Schlaglöcher, Straßenbahnschienen, Sand, Schotter, μ-Split, etc.) berücksichtigt werden.

Auch können die Ausfallszenarien variieren (plötzlicher Ausfall des Systems bzw. langsam ausfallendes System). Das betrachtete System kann total ausfallen, degradiert ausfallen (und dies in unterschiedlichen Stufen vom Totalausfall bis hin zu Funktionsstörungen, die nur das Fachpersonal erkennt), unerwünschte Funktionsstörungen aufweisen, wie z. B. Lenkausschlag in die falsche Richtung, oder temporär ausfallen (vgl. [Bin01]).

Als Ergebnis dieses Schrittes werden mögliche Gefährdungen unter Berücksichtigung von jeweils u. U. unterschiedlichen Fahrsituationen den Funktionen der Funktionsliste zugeordnet.

5.3.3. FGA 2 - Ermittlung der Fehlerauswirkungen

Hier werden die Auswirkungen der Gefährdungen auf das Fahrzeug, seine Insassen und sein Umfeld bestimmt. Dazu müssen in der Regel Experten konsultiert werden, die Erfahrung mit Systemen mit vergleichbaren Gefährdungen haben. Alternativ kann auch unterstützendes Material wie Analysen, Studien oder Tests verwendet werden. Jeder Einsatz von externem Material muss dabei dokumentiert werden, um die Nachvollziehbarkeit von Entscheidungen sicherzustellen.

Von einer korrekten Abschätzung möglicher Auswirkungen ist die Einstufung der Gefährdungen und damit der Funktionen in entsprechende Risikoklassen oder Sicherheitsstufen abhängig. Dabei müssen auch kombinierte Auswirkungen zu Rate gezogen werden.

5.3.4. FGA 3 - Klassifikation der ermittelten Gefährdungen

Unter Verwendung einer geeigneten Methode zur Klassifikation wie beispielsweise die vorgestellte CSA (vgl. Abbildung 5.5 auf Seite 91) werden die ermittelten Gefährdungen klassifiziert. Dabei kommen die dem jeweiligen Risikographen zugrunde liegenden Parameter zur Anwendung, bei CSA also Schadensauswirkung, Beherrschbarkeit und Häufigkeit des Auftretens eines Fehlers.

5.3.5. FGA 4 - Ableitung von quantitativen bzw. qualitativen Anforderungen

Jeder Gefährdung kann auf Grundlage der Einstufung in eine Gefährdungsklasse oder Sicherheitsstufe eine quantitative Zuverlässigkeitsanforderung zugeordnet werden. Qualitative Anforderungen werden meist in Textform spezifiziert.

Auf der höchsten Hierarchieebene kann diese Zuordnung beispielsweise gemäß Tabelle 5.2 auf Seite 97 erfolgen. Auf niedrigeren Hierarchieebenen ist die Abhängigkeit der identifizierten Funktionen von übergeordneten Funktionen zu beachten. Wenn dort die Auswirkungen von Gefährdungen sich ebenfalls auf das gesamte Fahrzeug und seine Insassen auswirken (was meist der Fall ist), dann kann eine Abschätzung von quantitativen Anforderungen ebenfalls gemäß Tabelle 5.2 erfolgen. Wenn aber nur Gefährdungen identifiziert werden können, deren Auswirkungen sich lediglich auf die übergeordnete Funktion beziehen, dann müssen die quantitativen Anforderungen gemäß den Gesetzen der Booleschen Algebra abgeleitet werden (beispielsweise unter Zuhilfenahme eines Fehlerbaums).

5.3.6. FGA 5 - Spezifikation von Methoden für die Verifikation

Für jede ermittelte Gefährdung ist anzugeben, wie das System die mit der Gefährdung verbundenen Anforderungen erfüllen kann und wie diese Erfüllung verifiziert werden kann.

Bei quantitativen Anforderungen erfolgt die Verifikation in aller Regel über einen Fehlerbaum. Dabei dienen die identifizierten Gefährdungen als Top-Ereignisse für die entstehenden Fehlerbäume, d. h. für jede Gefährdung entsteht ein Fehlerbaum. Alternativ kann eine Verifikation auch mittels einer FMEA erfolgen.

5.3.7. FGA 6 - Vergleich mit bisherigen Erfahrungen

Wenn die Gefährdungen entsprechend klassifiziert sind, bietet sich noch ein Vergleich mit den Erfahrungen mit herkömmlichen Systemen mit vergleichbaren Randbedingungen an. So kann sichergestellt werden, dass die Liste der Gefährdungen vollständig ist und dass die Gefährdungen konsistent mit vergleichbaren Systemen klassifiziert wurden.

Dabei ist aber zu beachten, dass eine Gefährdung, die in einem bisherigen System unkritisch zu sehen war, unter veränderten Randbedingungen eine sehr viel größere Bedeutung bekommen kann.

5.3.8. FGA 7 - Wiederholung für niedrigere Hierarchieebenen

Am Ende einer *Funktionalen Gefährdungsanalyse* für eine bestimmte Hierarchieebene werden die Ergebnisse in einem FGA-Formblatt oder einem elektronischen Pendant festgehalten. Ein Beispiel dafür ist in Tabelle 5.3 auf der nächsten Seite zu sehen, ein weiteres Beispiel findet sich im Beispiel-Kapitel 6.1.

Tabelle 5.3.: Formblatt für die Funktionale Gefährdungsanalyse

Nr.	Funktion	Gefährdung	Randbedingung	Auswirkungen	Klass.

Auf der nächsten Hierarchieebene wird nun der *Funktionale Entwurf* und dann auch die *Funktionale Gefährdungsanalyse* für die neu identifizierten Funktionen wiederholt. Dafür werden die Einstufungen der Funktionen auf Sicherheitsstufen der neu identifizierten Funktionen vererbt. Dabei gelten folgende Regeln:

Regel 1
Auf der folgenden Hierarchieebene muss mindestens eine Subfunktion die selbe Gefährdungseinstufung erhalten, wie die übergeordnete Funktion. Andere Subfunktion können niedrigere Gefährdungseinstufungen erhalten, keine darf eine höhere Einstufung erhalten als die übergeordnete Funktion.

Regel 2
Beeinflussen sich zwei Funktionen gegenseitig, so müssen beide die gleiche Gefährdungseinstufung haben.

Diese Regeln müssen bei der Bewertung der ermittelten Funktionen auf der nächsten Ebene beachtet werden. Sollte sich ein Widerspruch ergeben, muss die Vorgehensweise auf der entsprechenden Ebene überprüft werden.

5.4. SE - System-Entwurf

Das Ergebnis der vorigen Schritte *Funktionaler Entwurf* und *Funktionale Gefährdungsanalyse* ist eine vollständige funktionale Gliederung des Systems. Dabei sind den funktionalen Einheiten entsprechende Sicherheitsanforderungen zugeordnet, nach Möglichkeit in Form von quantitativen Anforderungen an die Zuverlässigkeit der jeweiligen funktionalen Einheiten. Darüber hinaus sind aus der *System-Anforderungsanalyse* weitere Anforderungen an das System bekannt.

Ausgehend von der bis hierher rein funktionalen Beschreibung wird im *System-Entwurf* diese Beschreibung des Systems auf eine Systemarchitektur abgebildet. Ergebnis ist eine Darstellung der Architektur und eine Beschreibung der Implementierung des Systems unter Berücksichtigung auch der nicht-funktionalen Anforderungen. Dazu gehört eine Schnittstellenbeschreibung und eine Spezifikation der Systemintegration.

Die Phase *System-Entwurf* wechselt iterativ mit der *Entwurfsbegleitenden System-Sicherheitsbewertung*, in der eine Bewertung der Architekturalternativen erfolgt. Werden die Sicherheits- bzw. Zuverlässigkeitsanforderungen nicht erfüllt, so muss eine Designänderung erfolgen.

In einem ersten Schritt beim *System-Entwurf* entsteht eine grobe Beschreibung des Systems, die die grundsätzliche Systemarchitektur beinhaltet. Anhand von Realisierbarkeitsuntersuchungen und den zur Verfügung stehenden Produktinformationen wird dieser Lösungsvorschlag hinsichtlich der Machbarkeit bewertet. Der Lösungsvorschlag wird dann immer weiter verfeinert, die Definition endet mit der Identifikation und Definition der Schnittstellen.

Basierend darauf erfolgt die Bewertung der Sicherheitseigenschaften in der parallel durchgeführten *Entwurfsbegleitenden System-Sicherheitsbewertung*. Wenn das Konzept für das System die Anforderungen nicht erfüllt, muss die Systemarchitektur geändert werden.

In der Regel existieren mehrere Architekturvarianten, welche die funktionalen Anforderungen erfüllen. Die Auswahl einer Variante erfolgt durch Vergleich und Bewertung der unterschiedlichen Varianten bezüglich der Erfüllung der Zuverlässigkeitsanforderungen und der allgemeinen Randbedingungen.

Wenn eine grundsätzliche Systemarchitektur gewählt wurde wird das System weiter detailliert und konkret entworfen. Es werden einzelne Systemkomponenten ausgewählt, denen dann feste Aufgaben zugeteilt werden. Dies wird auf niedrigeren Abstraktionsebenen wiederholt. Dabei werden auch die externen und internen Schnittstellen des Systems beschrieben.

Sobald die Systemarchitektur feststeht, muss die Spezifikation der System-Integration und -Verifikation begonnen werden. In der Integrationsspezifikation ist die Integrationsstrategie festzuhalten und aufzuzeigen, welche Elemente, welche Maßnahmen und welche Hilfsmittel für die Integration des Systems erforderlich sind.

Dann erfolgt der konkrete Entwurf der Systemkomponenten, das System wird in Hardware und Software aufgeteilt. Dafür werden Hardware- und Software-Anforderungen entsprechend abgeleitet. Basierend darauf wird die Hardware- und die Software-Struktur entworfen. So werden auch beispielsweise die strukturellen Redundanzen oder u. U. redundante Kommunikationskanäle in diesem Schritt festgelegt. Die Beschreibung erstreckt sich bis auf die Ebene des Task-Scheduling und der Planung der Kommunikation zwischen verteilten Systemelementen; auf Hardware-Seite bis hin zur Auswahl von Rechnerkomponenten oder Details der Konstruktion.

Alle Aspekte, die die Sicherheit des Systems betreffen, werden wiederum in der parallel durchgeführten *Entwurfsbegleitenden System-Sicherheitsbewertung* analysiert und bewertet.

Am Ende des *System-Entwurfs* ist das System vollständig beschrieben, so dass es im nächsten Schritt *Implementierung* umgesetzt werden kann.

Sollte im Laufe der Systementwicklung zu einem späteren Zeitpunkt eine Änderung der Randbedingungen erfolgen, oder sollten sich die Randbedingungen oder die Systemanforderungen ändern (v. a. diejenigen bzgl. Sicherheit), so muss bis mindestens zum Schritt *System-Entwurf* zurückgegangen werden. Es muss erneut überprüft werden, ob das System auch vom Grundsatz her die Sicherheitsanforderungen erfüllt, u. U. müssen auch die Gefährdungen in der *Funktionalen Gefährdungsanalyse* neu bewertet werden. Daher sind Änderungen der Systemrandbedingungen spätestens ab diesem Zeitpunkt zu vermeiden.

In Abbildung 5.8 auf der nächsten Seite ist die Einordnung des *System-Entwurfs* in den Gesamtprozess zu sehen, die Tabelle darunter erläutert die mit anderen Schritten ausgetauschten Informationen. In Abbildung 5.9 auf Seite 104 sieht man die Abfolge der einzelnen Schritte beim *System-Entwurf*.

5.4.1. SE 1 - Erstellen und Auswahl eines Systemkonzepts

In diesem Schritt wird auf Basis der funktionalen Beschreibung aus dem *Funktionalen Entwurf*, der entsprechende Sicherheitsanforderungen aufgrund der Bewertung in der *Funktionalen Gefährdungsanalyse* zugeordnet sind, ein Systemkonzept erstellt. Darunter ist zu verstehen, dass die Forderungen an das System bezüglich seiner Funktionen, seiner Schnittstellen zu anderen Systemen und Komponenten,

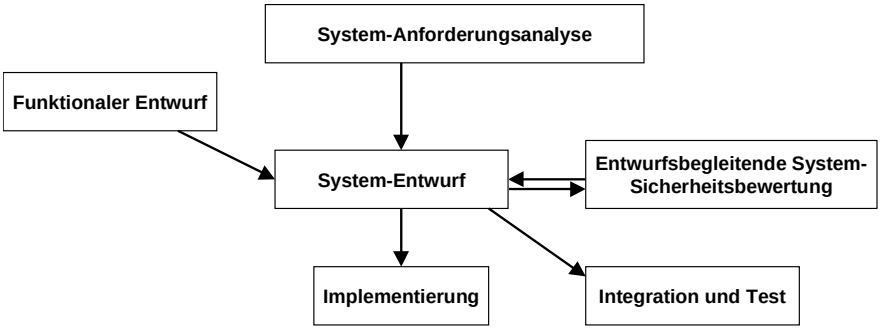

Abbildung 5.8.: Informationsfluss beim System-Entwurf

von	Information	nach
System-Anforderungsanalyse	nicht-funktionale Systemanforderungen	-
Funktionaler Entwurf	Komplette funktionale Beschreibung des Systems inkl. zugeordneten Sicherheitsanforderungen	-
-	Systembeschreibung bzw. System-Architektur	Entwurfsbegleitende System-Sicherheitsbewertung
Entwurfsbegleitende System-Sicherheitsbewertung	Rückmeldung, ob Anforderungen bzgl. Sicherheit und Zuverlässigkeit erfüllt werden inkl. Hinweisen zur Verbesserung der System-Architektur hinsichtlich Sicherheit und Zuverlässigkeit	-
-	Integrations- und Verifikations-Spezifikation	Integration und Test
-	Komplette Systembeschreibung mit System-Architektur, Hardware-/Software-Anforderungen, Schnittstellen	Implementierung

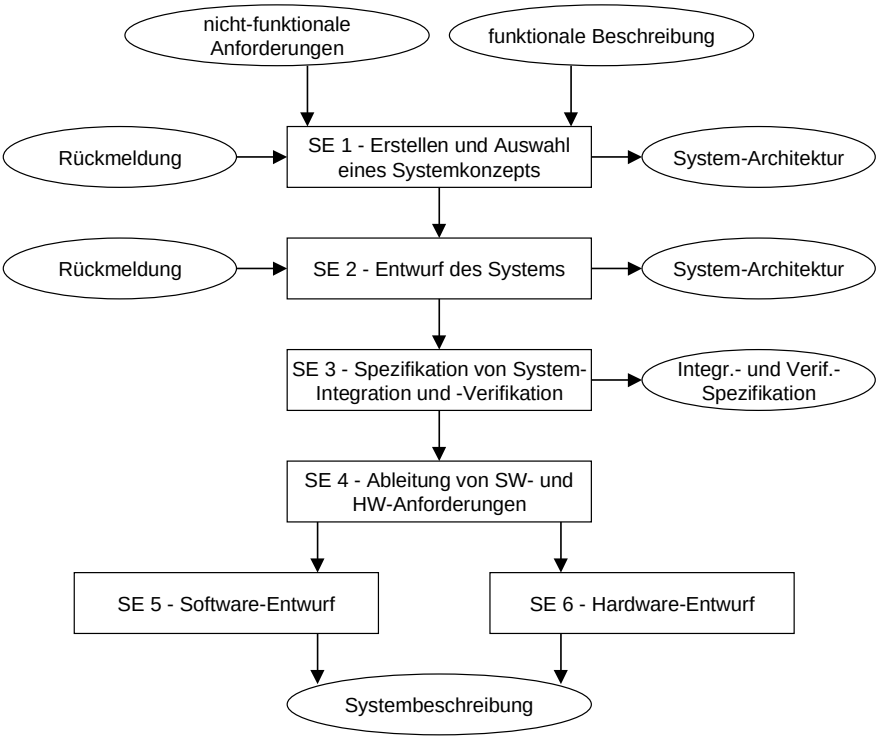

Abbildung 5.9.: Abwicklung des System-Entwurfs

seiner Betriebs-, Umwelt- und Randbedingungen sowie an seine Qualitätsmerkmale inkl. Lebensdauererwartungen umfassend festgelegt werden.

Basierend darauf wird eine Systemarchitektur skizziert, die als Grundlage für die Umsetzung des Systems dient. Das System wird in Subsysteme gegliedert, so entsteht eine Grundarchitektur. Diese Grundarchitektur beschreibt die Komponenten des Systems (beispielsweise die eingesetzten Rechnerkonzepte), oder die Kommunikationsbeziehungen zwischen den Komponenten des Systems. Auch werden hier grundsätzliche Fehlertoleranz- und Redundanzkonzepte beschrieben.

Dazu wird auf Basis der funktionalen Anforderungen und der Sicherheits- und Zuverlässigkeitsanforderungen die Gesamtfunktion auf verschiedene Subsysteme aufgeteilt, denen feste Aufgaben zugeteilt werden (z. B. Ansteuerung eines Aktuators oder Berechnung von Assistenzfunktionen). Das umfasst die Auslegung der Steuergeräte („fail-safe“ oder „fail-operational“) und die Festlegung der redundanten Kommunikationskanäle.

Dabei müssen verschiedene mögliche sicherheitstechnische Maßnahmen wie zusätzliche Hardware, zusätzliche Software, zusätzliche Rechenzeit, Überwachung, Abschaltpfade, etc. gegeneinander abgewogen werden, um die bezüglich Sicherheitsforderungen, Leistungsfähigkeit, Zuverlässigkeit und Kosten optimale Lösung zu finden.

Zur Definition einer Systemarchitektur gehört neben der Darstellung des technischen Aufbaus des Systems auch die Identifikation der Schnittstellen zwischen den Elementen der Architektur und des Gesamtsystems nach außen. Daher sind bei der weiteren Detaillierung einer ausgewählten Architektur alle in der Systemarchitektur identifizierten Schnittstellen in einer Schnittstellenbeschreibung festzuhalten.

Die Darstellung des Aufbaus des Systems wird ergänzt durch die Beschreibung des technischen Ablaufs (Interaktion zwischen den Elementen der Architektur). Durch die Architekturbeschreibung muss nachvollziehbar plausibel werden, dass die fachlichen Anforderungen und die im Rahmen der Anwenderforderungen definierten Randbedingungen durch die beschriebene technische Lösung abgedeckt werden können.

Um speziell die Anforderungen an die Sicherheit zu erfüllen, können mehrere verschiedene Lösungsmöglichkeiten vorgeschlagen werden. Denn in der Regel existieren mehrere Architekturvarianten, welche die funktionalen Anforderungen erfüllen. Die Bewertung der Varianten erfolgt in der parallel durchgeführten *Entwurfsbegleitenden System-Sicherheitsbewertung*. Basierend auf den Ergebnissen der *Entwurfsbegleitenden System-Sicherheitsbewertung* wird eine entsprechende Architektur ausgewählt und im Weiteren verfeinert. Falls die Auswahl der Lösung

hier nicht durchgeführt wird oder sie mehrere Varianten offen lässt, müssen die nachfolgenden Aktivitäten in jeder Lösungsvariante durchgeführt werden.

5.4.2. SE 2 - Entwurf des Systems

Auf Basis der festgelegten Systemarchitektur wird in diesem Schritt die Struktur des Systems weiter detailliert. Dabei werden die in der Systemarchitektur festgelegten Subsysteme beschrieben, Aspekte wie Kommunikation, Schnittstellen, Rechner, etc. kommen hinzu.

Nachdem die Architektur und die lokalen Subfunktionen bekannt sind, können die globalen Kommunikationsbeziehungen zwischen den Komponenten definiert werden. Festgelegt werden müssen die funktionale Spezifikation und die zeitlichen Bedingungen der Signale und damit auch die Semantik der Daten in der Kommunikationsschnittstelle. Um die Sicherheitsanforderungen zu erfüllen, müssen sicherheitsrelevante Nachrichten evtl. redundant übertragen werden und Sicherungsmechanismen wie Applikations-CRCs[3] oder Nachrichtenzähler verwendet werden.

Wenn mehrere Funktionen (z. B. Fahrwerks- und Antriebsfunktionen) gemeinsam in einem Kommunikationssystem integriert werden, müssen die Kommunikationsanforderungen aller Funktionen gemeinsam verwaltet und abgestimmt werden, um Ressourcenkonflikte auf dem Kommunikationssystem zu vermeiden.

Alle Schnittstellen, die im Systemkonzept identifiziert wurden, werden in einer Schnittstellenbeschreibung spezifiziert. Dabei sind unter Berücksichtigung der Art der jeweiligen Schnittstelle und der beteiligten Architekturelemente die relevanten Angaben zu Verwendung, Syntax und Semantik der Schnittstelle zu dokumentieren. Insbesondere müssen Schnittstellen zu den Sicherheitsmaßnahmen der beteiligten Architekturelementen dokumentiert werden. Dabei muss eine klare Abgrenzung des Sicherheitsanteils von den übrigen Elementen der Systemarchitektur erfolgen. Die Anzahl der Schnittstellen zwischen Sicherheitsanteil und nicht sicherheitskritischem Teil ist so klein wie möglich zu halten.

In diesem Schritt werden neue Elemente eingeführt und zusätzliche Schnittstellen geschaffen, die neue Schwachstellen bedeuten können. Sofern diese neu entstehenden Schwachstellen ausnutzbar sind, um das vorgegebene Sicherheitsziel zu verletzen, sind weitere Maßnahmen in das Sicherheitskonzept mit aufzunehmen, die dem entgegenwirken. Ebenso können beim Verfeinerungsprozess Schwachstellen in Form von unerwünschten Abhängigkeiten und Beziehungen auftreten (Fehlerfortpflanzung, „single point of failure“, etc.). Auch das Rechnerkonzept wird

[3] CRC: Cyclic Redundancy Check

von der Festlegung der Architektur beeinflusst. Abhängig von den zugeteilten Subfunktionen und den gestellten Zuverlässigkeitsanforderungen muss das Rechnerkonzept, die Prozessoren und die Peripherie der Subsysteme ausgewählt werden.

Daher werden parallel die getroffenen Entscheidungen in der *Entwurfsbegleitenden System-Sicherheitsbewertung* hinsichtlich der Erfüllung der Sicherheitsanforderungen bewertet. Dabei ist zu beachten, dass jede Anforderung auf mindestens ein Element der technischen Architektur, im Idealfall genau einem Architekturelement zugeordnet werden kann. Jede Forderung wird auf das in der Detaillierung niedrigste Element zugeordnet, das die Erfüllung der Forderung vollständig ermöglicht. Sofern eine Forderung von elementübergreifender Bedeutung ist, muss im Rahmen der Zuordnung genau abgewägt werden, welche einzelnen Architekturelemente diese Anforderung letztendlich zu erfüllen haben. Die Zuordnung muss so erfolgen, dass durch die Prüfung des entsprechenden Architekturelements die Erfüllung der Forderung nachgewiesen werden kann. Mit der Zuordnung der Sicherheitsanforderungen zu den Sicherheitsmaßnahmen ist nachzuweisen, dass jede Sicherheitsanforderung von mindestens einer Sicherheitsmaßnahme abgedeckt ist und ob das vorgegebene Sicherheitsziel mit den gewählten Sicherheitsmaßnahmen erreicht werden kann.

5.4.3. SE 3 - Spezifikation von System-Integration und -Verifikation

Neben der Umsetzung der Funktion in eine Systemarchitektur, Subsysteme und Systemkomponenten ist es notwendig, die spätere Integration der Komponenten und Subsysteme zu einem Gesamtsystem zu spezifizieren. Daher ist in einer Integrationsspezifikation festzulegen, wie die in der Systemarchitektur definierten Elemente zum Gesamtsystem zu integrieren sind. Die anzuwendende Strategie, durchzuführende Maßnahmen, eventuelle Restriktionen und Abhängigkeiten sowie notwendige Betriebs- und Hilfsmittel sind zu definieren.

Darüber hinaus ist auch die Verifikation der Subsysteme und der Systemkomponenten festzulegen. Es muss im Rahmen einer Verifikationsspezifikation festgelegt werden, wie überprüft werden kann, dass die jeweiligen Komponenten die funktionalen Anforderungen erfüllen. Dazu ist insbesondere eine genaue Zuordnung der funktionalen Beschreibung einer Einheit zu ihrer Umsetzung notwendig. Auch muss festgelegt sein, mit welchen Mitteln und Werkzeugen sowie anhand von welchen Ergebnissen die korrekte Funktion ermittelt werden kann.

5.4.4. SE 4 - Ableitung von Anforderungen an Software und Hardware

Die Präzisierung der Systemarchitektur, der Subsysteme und der Systemkomponenten im Schritt SE 2 geschieht noch unabhängig von einer möglichen konkreten Umsetzung in Hardware oder Software. Im Rahmen dieses Schrittes werden nun die technischen Anforderungen und die Betriebsinformationen bezogen auf eine Softwareeinheit oder eine Hardwareeinheit erweitert. Dabei wird festgelegt, welcher Systemteil bzw. welche Funktion in Software bzw. Hardware ausgeführt wird. Den Ausgangspunkt dieser Aktivität bilden die Anwenderforderungen, die Systemarchitektur und die bereits vorab abgeleiteten technischen Anforderungen.

Dafür werden zunächst die technischen Anforderungen an die Software- und Hardwareeinheiten spezifiziert und präzisiert, beispielsweise Anforderungen bzgl. der Realisierung der internen Schnittstellen. Des Weiteren sind Anforderungen an die externen Schnittstellen der Software- bzw. Hardwareeinheit wie Anforderungen an die technische Einsatzumgebung, Anforderungen an die Nutzerschnittstelle und Anforderungen an die Hardwareschnittstellen festzulegen.

Basierend darauf werden die Anforderungen an die Funktionalität, die Umsetzung und auch an die Qualität der Software- bzw. Hardwareeinheiten spezifiziert.

Von hier ab spaltet sich die weitere Systementwicklung in die Softwareentwicklung im Schritt SE 5 „Software-Entwurf“ und in die Hardwareentwicklung im Schritt SE 6 „Hardware-Entwurf“.

5.4.5. SE 5 - Software-Entwurf

Der Software-Entwurf und auch der Hardware-Entwurf erfolgen im Wesentlichen gemäß den Vorgaben aus dem V-Modell ’97.

Der Software-Entwurf besteht aus dem Software-Grobentwurf, bei dem die Softwarearchitektur und die Software-Schnittstellen entworfen werden, und dem Software-Feinentwurf mit dem Schwerpunkt auf dem Entwurf von Softwarekomponenten und -modulen (vgl. Abbildung 5.10 auf der nächsten Seite).

5.4.5.1. Software-Grobentwurf

Dieser Schritt beinhaltet den Entwurf der Softwarearchitektur inklusive der Vervollständigung der Schnittstellenübersicht, die Beschreibung der Software-Schnittstellen und die Fortführung des Integrationsplans auf Softwareebene.

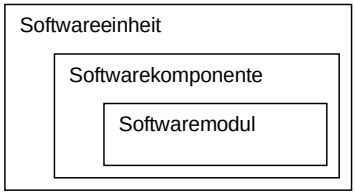

Abbildung 5.10.: Verwendete Konvention zur hierarchischen Gliederung

Der Entwurf der Softwarearchitektur hat die Aufgabe, Softwareprozesse zu bilden und gegebenenfalls die Aufteilung auf Prozessoren vorzunehmen, sowie die Kommunikation und Synchronisation der Prozesse zu entwerfen und aus Sicht der Softwareeinheit Softwaremodule und Softwarekomponenten zu definieren. Zu jedem dieser Architekturelemente ist eine kurze Leistungsbeschreibung zu verfassen und die entstehenden Schnittstellen sind zu identifizieren. In der Schnittstellenbeschreibung muss dann das Zusammenspiel der Softwaremodule und Softwarekomponenten spezifiziert werden. Diese Informationen dienen als Ausgangspunkt für den Software-Feinentwurf.

Die bereits auf Systemebene begonnenen Betriebsinformationen sind hier um Angaben für die betrachtete Softwareeinheit zu ergänzen.

Der Software-Grobentwurf lässt sich in drei einzelne Schritte unterteilen.

Entwurf der Softwarearchitektur Die Softwarearchitektur beschreibt die Zerlegung einer Softwareeinheit in Softwarekomponenten, Prozesse und Softwaremodule. Vorschläge für mögliche Softwarearchitekturen werden erarbeitet und bewertet (bzgl. Sicherheit wieder in der *Entwurfsbegleitenden System-Sicherheitsbewertung*); für die weitere Bearbeitung ist ein Lösungsvorschlag auszuwählen.

Die Auswahl einer entsprechenden Architektur kann auch unter Verwendung eines sogenannten „Systembaukastens“ oder unter Zuhilfenahme von Architekturkomponenten erfolgen (ein Beispiel für die Anwendung von „Software Design Patterns“ findet sich beispielsweise in [AHKSC02]).

Softwareeinheiten bestehen im allgemeinen aus mehreren Prozessen oder Tasks, die parallel oder quasi-parallel ablaufen. Ziel ist es, eine Softwareeinheit unter dem Gesichtspunkt notwendiger oder möglicher Parallelverarbeitung zu strukturieren. Dabei sind die Gegebenheiten des Betriebs- und Laufzeitsystems und die Erfordernisse und Einschränkungen der Hardware und der Programmiersprache zu berücksichtigen. Die Softwarearchitektur enthält eine Beschreibung des

Prozess-Ensembles samt seiner Strukturierung, Kommunikation, Synchronisation und Darstellung des dynamischen Ablaufmodells.

Zu jedem Baustein (Prozess, Softwarekomponente, Softwaremodul) des ausgewählten Vorschlags sind eine kurze Leistungsbeschreibung und die Relevanz hinsichtlich der Sicherheit anzugeben. Die aufgrund von Architekturentscheidungen festgelegten Schnittstellen sind in der Softwarearchitektur zu identifizieren und in der Schnittstellenübersicht zu dokumentieren. Die vollständige Abdeckung der Anforderungen durch die in der Softwarearchitektur definierten Prozesse, Softwarekomponenten und Softwaremodule ist nachzuweisen.

Neben den Abhängigkeiten der Sicherheitsfunktionen sind die Wechselwirkungen der Sicherheitsmechanismen, die zur Realisierung der Sicherheitsfunktionen gewählt wurden, zu untersuchen. Auch sind die Auswirkungen, die die Realisierung der Sicherheitsfunktionen auf andere Softwareeinheiten haben könnte, zu evaluieren.

Dabei ist insbesondere festzustellen, ob und gegebenenfalls welche sicherheitsrelevanten Anteile in anderen Softwarekomponenten oder Softwaremodulen bei der Realisierung entstehen. Die Schnittstellen vom sicherheitsrelevanten zum nicht-sicherheitsrelevanten Teil sind in jeder Softwareeinheit zu minimieren. Diese Schnittstellen in jeder Softwareeinheit und zwischen den einzelnen Softwareeinheiten sind exakt zu definieren.

In der Regel werden neben den für die Realisierung von operationellen Anwendungsfunktionen erforderlichen Softwarekomponenten oder Softwaremodulen auch Softwarekomponenten bzw. Softwaremodule benötigt, die lediglich technische Hilfsfunktionen zur Verfügung stellen und deshalb nicht aus den operationellen Anforderungen, sondern aus den Gegebenheiten des technischen Lösungsansatzes herzuleiten sind (z. B. Gateway-Elemente). Diese sind bei der Festlegung der Softwarearchitektur mit zu berücksichtigen.

Einfluss auf diese Modularisierungsentscheidungen haben die Prozess-Struktur der Softwareeinheit, zusätzlich notwendige (beispielsweise systeminterne) Funktionen oder auch der Zielrechner (Laufzeitsystem, Protokolle, usw.). Außerdem können verschiedene Konzepte der Fehlerbehandlung, Kommunikation, Kriterien wie „Information Hiding“, Datenabstraktion, Abgeschlossenheit, Minimalität der Schnittstellen, Überschaubarkeit, Prüfbarkeit, Integrierbarkeit, Wiederverwendbarkeit oder auch Anforderungen an das Zeitverhalten oder Zuverlässigkeitsanforderungen die Prozess-Struktur beeinflussen.

Entwurf der Software-internen und -externen Schnittstellen Die beim Entwurf der Softwarearchitektur in der Schnittstellenübersicht identifizierten Schnittstellen sind in der Schnittstellenbeschreibung im einzelnen detailliert darzustellen. Bereits beschriebene Schnittstellen sind gegebenenfalls weiter zu präzisieren.

Sicherheitsaspekte, wie sie bereits bei der Identifikation der Schnittstellen eine Rolle spielten, sind hier weiter und mit besonderer Sorgfalt zu verfolgen. Alle Schnittstellen der sicherheitsspezifischen und sicherheitsrelevanten Softwarekomponenten bzw. Softwaremodule müssen mit ihrem Zweck und ihren Parametern beschrieben werden. Die Separierung vom nicht sicherheitsrelevanten Teil muss sichtbar sein.

Spezifikation der Software-Integration und -Verifikation Für die Spezifikation der Integration und Verifikation sind Vorschriften und Vorgehensweisen für den Zusammenbau der Softwareeinheit aus Softwarekomponenten und der Softwarekomponenten selbst aus Softwaremodulen aus technischer Sicht zu regeln. Er ist im Hinblick auf Termine, Betriebs- und Hilfsmittel, Werkzeuge, Personal, Schnittstellen und ähnliche Einflüsse zu erstellen. In der Integrationsspezifikation sind die Konfigurierung der Softwareeinheit und alle für die Integrationsaktivität benötigten Informationen festzuschreiben; Informationen für den Integrationstest sind abzuleiten und festzuhalten.

Die Arbeiten an der Integrationsspezifikation liefern gleichzeitig weitere Angaben für die Betriebsinformationen. So ist auch die Vorgehensweise bei der Verifikation der Softwareeinheiten, der Softwarekomponenten und der Softwaremodule festzulegen.

5.4.5.2. Software-Feinentwurf

Die Ausgangsbasis für diesen Schritt bilden die Beschreibung der Softwarearchitektur und die Schnittstellenbeschreibung. Dort sind alle Informationen festgehalten, die erforderlich sind, um die Leistung eines Softwaremoduls in Anspruch nehmen zu können. Die Vorgaben und Details für die Realisierung jedes Softwaremoduls und jeder Softwarekomponente müssen festgelegt werden. Auf dieser Grundlage müssen sodann Betriebsmittel- und Zeitbedarf der einzelnen Architekturelemente und der gesamten Softwareeinheit ermittelt werden, welche den diesbezüglichen Anforderungen Rechnung zu tragen haben.

Der Software-Feinentwurf lässt sich in zwei einzelne Schritte unterteilen.

Beschreibung von Softwarekomponenten und -modulen Gegenstand dieses Schrittes ist die softwaretechnische Realisierung der Softwarekomponenten und Softwaremodule. Die Konstruktion jedes Softwaremoduls und jeder Softwarekomponente muss bis auf die Ebene einer Programmiervorgabe beschrieben werden.

Die Softwarekomponenten und Softwaremodule müssen hinsichtlich ihrer Umgebung, der Realisierung ihrer Funktionalität, der Datenhaltung, Ausnahme- und Fehlerbehandlung, usw. spezifiziert und bis hin zu einer Programmiervorgabe formuliert werden.

Die in der Softwareeinheit benutzten Daten sind in einem Datenkatalog aufzunehmen. Die Eingangsinformationen für den Datenkatalog liefern die datenbankbezogenen Informationen der Softwarearchitektur. Im Datenkatalog sind implementierungsabhängige Informationen, wie Bezeichner, Datentyp, Datenformat, Lebensdauer, Zugriffsart, Zugriffs- und Entstehungszeiten und -frequenzen, Zuordnung zu Datenbanken, Speicherart, Speicherplatzbedarf usw. festzuhalten.

Analyse von Betriebsmittel- und Zeitbedarf Hier erfolgt die Ermittlung und die Überprüfung des errechneten Bedarfs auf seine Realisierbarkeit hin. Auf Basis der Softwarearchitektur und der Softwarekomponenten- und Softwaremodulbeschreibung erfolgt die Ermittlung von benötigten Betriebsmitteln und des Zeitbedarfs. Die ermittelten Leistungsmerkmale sollten gegenüber den in den technischen Anforderungen festgehaltenen Eigenschaften der Softwareeinheit noch genügend Spielraum aufweisen, um spätere Pflege- und Änderungsmaßnahmen ohne Redesign ausführen zu können. Ist dieser Spielraum nicht gegeben, so hat an dieser Stelle eine genaue Überprüfung der Realisierungsentscheidungen zu erfolgen.

5.4.6. SE 6 - Hardware-Entwurf

Auch der Hardware-Entwurf erfolgt im Wesentlichen gemäß den Vorgaben aus dem V-Modell ’97.

Der Hardware-Entwurf besteht aus dem Hardware-Grobentwurf, bei dem Konzeption und der Spezifikation der eingesetzten Hardware im Mittelpunkt stehen, und dem Hardware-Feinentwurf mit dem Schwerpunkt auf dem Detailentwurf.

5.4.6.1. Hardware-Grobentwurf

Der Hardware-Grobentwurf hat die Aufgabe, die spezifizierten Funktionen in einzelne Funktionsblöcke zu zerlegen und eine Hardwarearchitektur zu definieren. Ferner wird das Zusammenspiel der Hardwarekomponenten spezifiziert.

Der Hardware-Grobentwurf lässt sich in vier einzelne Schritte unterteilen.

Erarbeitung und Bewertung von Lösungsvorschlägen Vorschläge für mögliche Grobentwürfe einer Hardwareeinheit werden erarbeitet, u. U. durch Simulation oder Versuchsmuster, und systematisch nach technischen Aspekten und Kostenaspekten bewertet.

Grobentwurf einer Hardwareeinheit Die Erstellung einer Hardwarearchitektur beinhaltet die Erstellung von Entwürfen, Blockschaltbildern, Funktionsskizzen für Mechanik-, Elektronik- oder andere Bestandteile. Auf Basis der Hardwarearchitektur erfolgt eine Untersuchung auf wiederverwendbare Teile. Diese werden daraufhin geprüft, ob die vorgegebene Qualität und Liefersicherheit den Anforderungen entsprechen; gegebenenfalls werden Einzelheiten spezifiziert. Weiter wird untersucht, ob und welche Hardwaremodule oder Hardwarekomponenten selbst oder extern entwickelt bzw. konstruiert werden. Im Zusammenhang mit dem Hardware-Grobentwurf kann eine erste Festlegung von Bauteilen und Materialien nötig sein, die die Hardwarearchitektur maßgeblich beeinflussen.

Spezifikation der Hardware-internen Schnittstellen Soweit erforderlich, werden interne Schnittstellen zwischen Hardwarekomponenten und Hardwaremodulen der Hardwareeinheit spezifiziert. Dabei sollte nach Möglichkeit auf eine Trennung zwischen Sicherheits- und nicht-Sicherheitsanteilen geachtet werden.

Spezifikation der Hardware-Integration Der Ablauf der Integration der Hardwarekomponenten und Hardwaremodulen zu einer Hardwareeinheit wird bezüglich der Abfolge und der schrittweisen Prüfung von Teilfunktionen festgelegt.

5.4.6.2. Hardware-Feinentwurf

Im Hardware-Feinentwurf erfolgt die endgültige Festlegung der Bauteile und Materialien, die Erarbeitung der Detailentwürfe für die einzelnen Hardwarekomponenten und Hardwaremodule, die Festlegung von Abmessungen, Auslegung von Schaltungen oder Logikkomponenten, Berechnung von Toleranzen, die Erstellung der Konstruktionsunterlagen und Funktionsbeschreibungen für die einzelnen Bestandteile und die Durchführung von Analysen und Nachweisberechnungen.

Der Hardware-Feinentwurf lässt sich in drei einzelne Schritte unterteilen.

Detailentwürfe für Hardwarekomponenten und Hardwaremodule Für jede Hardwarekomponente bzw. für jedes Hardwaremodul werden die endgültigen Bauteile und Materialien festgelegt. Die Entwürfe werden bis zur Bauteile-Ebene verfeinert und die für die Realisierung notwendigen Informationen dokumentiert. Eine detaillierte Funktionsbeschreibung der Hardwarekomponente bzw. des Hardwaremoduls wird erstellt.

Erstellung eines Zeichnungssatzes Der Zeichnungssatz für die gesamte Hardwareeinheit enthält alle für den Nachbau dieser Einheit notwendigen Unterlagen wie Aufbauübersichten, Zeichnungen, Stücklisten, Stromlaufpläne, Verdrahtungspläne, Layout, Liefervorschriften, usw. Die Prüfvorschriften gehören inhaltlich gesehen ebenfalls zur jeweiligen Hardwareeinheit und werden oft als Bestandteil des Zeichnungssatzes mitgeführt.

Durchführen von Analysen und Nachweisen Die besonderen Anforderungen an Sicherheit und Zuverlässigkeit der Hardwareeinheit werden gesondert in der parallel durchgeführten *Entwurfsbegleitenden System-Sicherheitsbewertung* untersucht. Dadurch wird nachgewiesen, dass die Anforderungen an das Produkt betreffend Zuverlässigkeit, Toleranzen, Festigkeit, Sicherheit, usw. von der jeweiligen Hardwareeinheit erfüllt werden. Dazu kommen vorgegebene Analyse- und Berechnungsverfahren aufgrund von Bauteiledaten oder Erfahrungen mit ähnlichen Systemen zum Einsatz.

5.5. ESSB - Entwurfsbegleitende System-Sicherheitsbewertung

In der *Entwurfsbegleitenden System-Sicherheitsbewertung* findet eine systematische Untersuchung des im Schritt *System-Entwurf* vorgeschlagenen Systemkonzepts, der daraus abgeleiteten Systemarchitektur und der dabei verwendeten Subsysteme, Systemkomponenten, Softwarekomponenten, Softwaremodule, Hardwareeinheiten, Hardwarekomponenten und Hardwaremodule statt. Ziel der Untersuchung ist die Bewertung der jeweiligen Systembestandteile auf die Erfüllung der entsprechenden Sicherheitsanforderungen.

Dazu wird analysiert, ob die Architektur bzw. die jeweiligen Systemkomponenten des geplanten Designs die qualitativen und quantitativen Anforderungen aus der *Funktionalen Gefährdungsanalyse* erfüllen, oder ob mögliche Fehler oder Ausfälle

unter Umständen zu entsprechenden Gefährdungen führen können. Die eingesetzten Methoden und Werkzeuge sind dabei quantitativer und qualitativer Art.

Die *Entwurfsbegleitende System-Sicherheitsbewertung* erfolgt parallel zur Konzeptionsphase des Systems, also zu einem Zeitpunkt, zu dem das System physikalisch noch nicht vorliegt. Dies stellt eine besondere Herausforderung dar, gibt andererseits dem Entwickler aber die Möglichkeit, zu einem relativ frühen Zeitpunkt eine Abschätzung treffen zu können, ob das System unter den gegebenen Sicherheits-Randbedingungen umgesetzt werden kann. Die Qualität dieser Abschätzung hängt sehr stark von den hierbei eingesetzten Methoden, von der Qualität der verwendeten Daten und dem dazu aufgebrachten zeitlichen Aufwand ab.

Die Bewertung des physikalisch implementierten Systems erfolgt dann parallel zu *Integration und Test* im Schritt *System-Sicherheitsbewertung*.

In einem ersten Schritt der *Entwurfsbegleitenden System-Sicherheitsbewertung* wird untersucht, ob die System-Sicherheitsanforderungen vollständig sind. Wenn notwendig, werden sie komplettiert. Durch besondere Aspekte und Eigenschaften einer gewählten Systemarchitektur oder auch durch sich ändernde Randbedingungen können zusätzliche Sicherheitsanforderungen hinzukommen.

Danach wird eine gründliche Sicherheitsuntersuchung der jeweiligen Systemarchitektur bzw. der entsprechenden Systemkomponenten durchgeführt. Zur Identifikation von Fehlern oder von Kombinationen von Fehlern, die zu einer kritischen Gefährdung im System führen könnten, werden die Architekturkonzepte und Implementierungsvorschläge des Systems durch entsprechende quantitative Bewertungsmethoden analysiert. Wenn keine quantitative Analyse möglich ist, so wird die Untersuchung qualitativ durchgeführt und quantitative Ergebnisse abgeschätzt.

Die dazu verwendeten Methoden sind Fehlerbaumanalysen (vgl. Kapitel 2.4.2) oder Markov-Analysen (vgl. Kapitel 2.4.3) für Sicherheitsanforderungen, die quantitativ formuliert sind und deren Verifikation auf einer Überprüfung von äquivalenten Zuverlässigkeitsanforderungen beruht. Sicherheitsanforderungen, die nur qualitativ formuliert sind, werden durch FMEA bzw. FMECA (vgl. Kapitel 2.4.1) überprüft. Ein weiteres Augenmerk gilt Gefährdungen, die durch die Systemarchitektur selbst herrühren. Diese werden in der Regel ebenfalls durch FMECA-Analysen untersucht. Sogenannte „Common Mode Fehler“, also Verknüpfungen von Fehlern über eine gemeinsame Fehlerursache, werden im Rahmen einer besonderen Fehleranalyse geprüft.

Auf Basis dieser Ergebnisse werden Sicherheitsanforderungen für den *System-Entwurf* auf der nächstniedrigen Ebene oder für andere beteiligte Systeme abgeleitet. So wird sichergestellt, dass dann die Implementierung des Systems die Anforderungen an Sicherheit und Zuverlässigkeit erfüllt.

In Abbildung 5.11 ist die Einordnung der *Entwurfsbegleitenden System-Sicherheitsbewertung* in den Gesamtprozess zu sehen, die Tabelle darunter erläutert die mit anderen Schritten ausgetauschten Informationen. In Abbildung 5.12 auf der nächsten Seite sieht man die Abfolge der einzelnen Schritte bei der *Entwurfsbegleitenden System-Sicherheitsbewertung*.

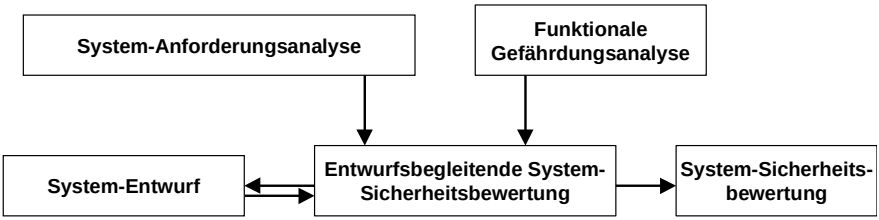

Abbildung 5.11.: Informationsfluss bei der Entwurfsbegleitenden System-Sicherheitsbewertung

von	Information	nach
System-Anforderungsanalyse	Sicherheitsanforderungen	-
Funktionale Gefährdungsanalyse	Zugeordnete Anforderungen an Sicherheit und Zuverlässigkeit sowie Vorgehensweise zur Verifikation dieser Anforderungen	-
System-Entwurf	Systembeschreibung bzw. System-Architektur	-
-	Rückmeldung, ob Anforderungen bzgl. Sicherheit und Zuverlässigkeit erfüllt werden inkl. Hinweisen zur Verbesserung der System-Architektur hinsichtlich Sicherheit und Zuverlässigkeit	System-Entwurf
-	Ergebnisse der Entwurfsbegleitenden System-Sicherheitsbewertung	System-Sicherheitsbewertung

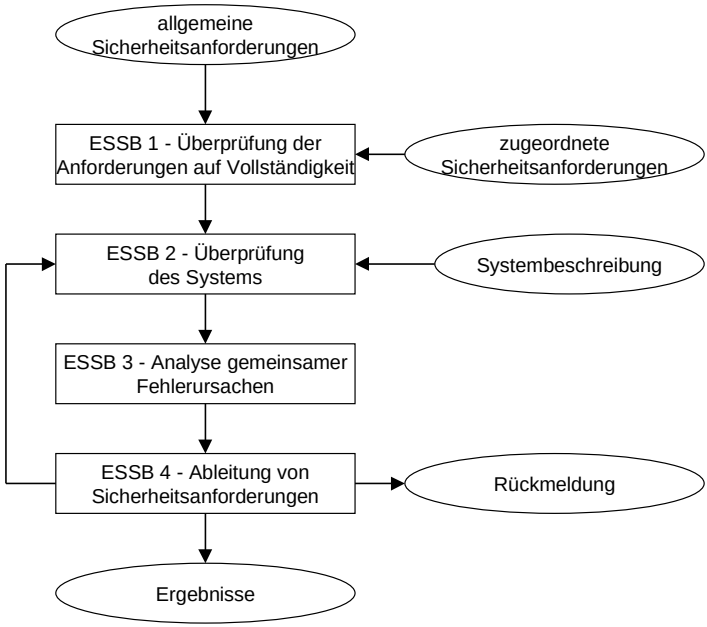

Abbildung 5.12.: Abwicklung der Entwurfsbegleitenden System-Sicherheitsbewertung

5.5.1. ESSB 1 - Überprüfung der Sicherheitsanforderungen auf Vollständigkeit

Im Rahmen der *Funktionalen Gefährdungsanalyse* wurde eine Anzahl von funktional basierten Sicherheitsanforderungen für den Entwurf des betrachteten Systems aufgestellt. Diese werden nun durch nicht-funktionale Anforderungen aus der *System-Anforderungsanalyse* ergänzt.

Bedingt durch die jeweilige konkrete Wahl der Umsetzung des Systems können durch besondere Aspekte und Eigenschaften der gewählten Systemarchitektur, der Subsysteme, Systemkomponenten, Software bzw. Hardware zusätzliche Sicherheitsanforderungen hinzukommen. Beispielsweise bedingt die Konzeption der Verteiltheit des Systems neue Sicherheitsanforderungen an eine gewünschte Fehlereindämmung, oder bestimmte Werkstoffe stellen neue Anforderungen an die Brennbarkeit.

Zusätzlich können durch sich ändernde Randbedingungen aufgrund der Wahl eines bestimmten Systemkonzeptes zusätzliche Sicherheitsanforderungen hinzukommen (beispielsweise durch bestimmte Aspekte der Bedienung oder Wartung, um nur zwei Bereiche zu nennen).

Daher wird in einem ersten Schritt der *Entwurfsbegleitenden System-Sicherheitsbewertung* untersucht, ob die bis hierher formulierten System-Sicherheitsanforderungen vollständig sind. Zusätzliche Anforderungen müssen ergänzt werden.

5.5.2. ESSB 2 - Überprüfung des Systems gegen die Sicherheitsanforderungen

Die beim *System-Entwurf* vorgeschlagene Systemarchitektur bzw. die entsprechenden Systemkomponenten in Form von Subsystemen, Hardware oder Software werden nun in diesem Schritt bezüglich der Erfüllung der Sicherheitsanforderungen untersucht. Dies geschieht mit Hilfe von quantitativen und qualitativen Methoden.

Für Systembestandteile, deren Sicherheitsanforderungen im Rahmen der *Funktionalen Gefährdungsanalyse* in äquivalente quantitative Zuverlässigkeitsanforderungen überführt werden konnten, erfolgt diese Bewertung in Form von Fehlerbäumen (vgl. Kapitel 2.4.2). Dabei existiert für jede in der *Funktionalen Gefährdungsanalyse* identifizierte Gefährdung ein Fehlerbaum, die Gefährdung stellt jeweils das Top-Ereignis dar. Die vorhandene Grundstruktur dieser Fehlerbäume wird nun in diesem Schritt ausgefüllt, die einzelnen Bestandteile des Fehlerbaums ergeben sich durch die Umsetzung der funktionalen Systembeschreibung durch eine Systemarchitektur bzw. durch Systemkomponenten. Es werden die dabei verwendeten

Anforderungen an die Unabhängigkeit verschiedener Ereignisse festgehalten, die dann in ESSB 3 „Analyse gemeinsamer Fehlerursachen“ verifiziert werden müssen.

Das Ergebnis ist eine Auftretenswahrscheinlichkeit für das Top-Ereignis (also die Gefährdung), deren Wert mit der Anforderung aus der *Funktionalen Gefährdungsanalyse* verglichen werden kann. Allerdings ist zu beachten, dass dieser ermittelte Wert in der Regel großen Unwägbarkeiten ausgesetzt ist. Neben der Qualität der Datenquellen für die Elementarereignisse beeinflusst auch die Struktur des Fehlerbaumes das Ergebnis.

Können den Elementarereignissen keine Auftretenswahrscheinlichkeiten hinterlegt werden, so kann man trotzdem beispielsweise durch Analyse der Minimalschnitte des Fehlerbaumes Aussagen über die Signifikanz bzw. Importanz einzelner Elementarereignisse für das Auftreten des Top-Ereignisses geben.

Die Daten für die Auftretenswahrscheinlichkeiten der jeweiligen Elementarereignisse entstammen entsprechend gepflegten firmeninternen Datenbanken oder aus den jeweiligen öffentlich verfügbaren Datenbanken (vgl. Tabelle 2.4 auf Seite 26 in Kapitel 2.2.3).

Das Vorgehen kann dabei sehr umfangreich und komplex werden und sollte daher nach Möglichkeit automatisiert werden. Dazu sind bereits Hinweise in der Literatur auf entsprechende Forschungsaktivitäten vorhanden (z. B. bei Hedenetz [Hed01] oder Längst [Län03]).

Mit Hilfe von Fehlerbäumen sind nur statische Systeme modellierbar. Für dynamische Systeme kann die Markov-Analyse zum Einsatz kommen (vgl. 2.4.3). In [Mah00] wird beschrieben, wie die Markov-Analyse und Fehlerbäume miteinander verknüpft werden können.

Sicherheitsanforderungen, die qualitativ formuliert sind, bzw. Systembestandteile, die die Voraussetzungen für eine quantitative Zuverlässigkeitsanalyse nicht erfüllen, werden mit Hilfe von FMEA und FMECA analysiert (vgl. Kapitel 2.4.1). Hier kommt v. a. die System-FMEA zum Einsatz, deren Grundstruktur ebenfalls für jede Gefährdung bei der *Funktionalen Gefährdungsanalyse* angelegt wurde. Wenn die entsprechende Risikoprioritätszahl zu hoch ist, müssen entsprechende Gegenmaßnahmen ergriffen werden.

Grundsätzlich werden die eingesetzten Fehlerbäume bzw. FMECAs in der *Funktionalen Gefährdungsanalyse* begonnen und hier in der *Entwurfsbegleitenden System-Sicherheitsbewertung* strukturiert und mit Daten hinterlegt. Die Verifikation der Fehlerbäume bzw. FMECAs erfolgt in der *System-Sicherheitsbewertung*, wo auch evtl. quantitative Werte korrigiert werden.

Für jeden Detaillierungsschritt beim *System-Entwurf* wird eine entsprechende Analyse der Sicherheitsanforderungen durchgeführt. Diese Bewertung wird zu einem Zeitpunkt im Entwurfsprozess durchgeführt, wenn teilweise noch keine genauen Erfahrungen auf Komponentenebene vorliegen. Die durchgeführten Bewertungen der *Entwurfsbegleitenden System-Sicherheitsbewertung* müssen daher teilweise auf Ingenieurswissen und Erfahrungen mit ähnlichen Systemen zurückgreifen. Dieser projektübergreifende Prozess ist iterativer Natur und wird daher mit länger andauernder Anwendung zunehmend besser werden.

Ein besonderer Fall ist für Software gegeben, wo die Überprüfung gerade von quantitativen Sicherheitsanforderungen nicht trivial ist. Hier sei für entsprechende Verfahren auf die Literatur verwiesen, beispielsweise auf [EM03].

5.5.3. ESSB 3 - Analyse gemeinsamer Fehlerursachen

Im Rahmen dieses Schrittes (auch Common Cause Analyse, kurz CCA genannt) werden sogenannte „Common Mode Fehler“ analysiert. Unter Common Mode Fehlern werden Fehler im Sinne von Fehlerauswirkungen verstanden, die auf die gleiche Fehlerursache zurückzuführen sind (nach [Lap92]).

Die Analyse gemeinsamer Fehlerursachen gliedert sich in die „Zonale Sicherheitsanalyse“, in die „Analyse Spezieller Risiken“ und die eigentliche „Common Mode Analyse“.

5.5.3.1. Zonale Sicherheitsanalyse

In der Zonalen Sicherheitsanalyse (kurz ZSA) wird das jeweils betrachtete System in abgeschlossene physikalische Bereiche („Zonen“) aufgeteilt, dessen Schnittstellen zu den jeweiligen Nachbarbereichen minimiert sind. Für jede dieser Zonen wird eine gesonderte Sicherheitsanalyse durchgeführt, bei der ein besonderes Augenmerk auf die Schnittstellen zu den Nachbarbereichen gelegt wird. Dabei wird untersucht, inwiefern sich Fehler innerhalb einer Zone über die Schnittstellen auf die Nachbarzonen auswirken können. Dies wird als Grundlage für die Optimierung der Systemarchitektur verwendet.

Ziel ist dabei, sogenannte „fault containment regions“ zu schaffen. Darunter wird die Systemeigenschaft verstanden, dass sich Fehler in solchen Systembereichen sich nicht auf Nachbarbereiche auswirken können.

5.5.3.2. Analyse Spezieller Risiken

Bei der Analyse Spezieller Risiken (engl. Particular Risk Analysis, kurz PRA) wird das betrachtete System dahingehend untersucht, ob spezielle externe[4] Ursachen einen signifikanten Einfluss auf das System haben können. Sollten mögliche Auswirkungen zu größeren Gefährdungen führen, so sind entsprechende Gegenmaßnahmen zu treffen, um den Einfluss dieser externen Ursachen abzuschwächen.

Mögliche zu betrachtende Ursachen sind Feuer, hohe Energie, hoher Druck, Blitzschlag, Hochspannung, EMV[5], Steinschlag, besondere Flüssigkeiten (wie z. B. Öl, Benzin, Hydraulikflüssigkeit, Batteriesäure, Wasser usw.), Umweltbedingungen wie Hagel, Eis oder Schnee, Einwirkungen durch Tiere (z. B. Marderbiss), Reifenplatzer oder Sabotage, um nur einige Beispiele zu nennen.

5.5.3.3. Common Mode Analyse

Im Rahmen der Common Mode Analyse (kurz CMA) wird untersucht, inwiefern mögliche betrachtete Fehler in ihren Ursachen unabhängig sind. Insbesondere werden solche Ereignisse, die bei einer Fehlerbaumanalyse durch UND-Verknüpfungen verbunden sind, auf ihre Unabhängigkeit hin untersucht.

Besonderes Augenmerk bei der Common Mode Analyse wird auf Software-Entwicklungsfehler, Hardware-Entwicklungsfehler, Hardware-Ausfälle, außergewöhnliche Belastungen und Situationen, Anforderungsfehler, Umweltfaktoren, kaskadierende Fehler bzw. Fehler aufgrund der selben Quelle gelegt.

Für jede identifizierte mögliche Schwachstelle wird untersucht, inwiefern sie diese Anforderungen an die Fehlerunabhängigkeit verletzt. Die Ergebnisse werden entsprechend dokumentiert.

5.5.4. ESSB 4 - Ableitung von Sicherheitsanforderungen

Hier werden die Sicherheitsanforderungen für Komponenten auf der nächstniedrigen Hierarchieebene abgeleitet. Jede Sicherheitsanforderung auf Systemebene muss man entsprechenden Subsystemen bzw. Komponenten zuordnen können, die Bestandteil des betrachteten Systems sind.

Diese Zuordnungen beinhalten u. a. eine aktualisierte Liste von Gefährdungen, die eine Aufstellung beinhaltet, wie jeweils die qualitativen und quantitativen

[4] „extern“ bedeutet, dass die Ursachen sich außerhalb des betrachteten Systems befinden. Sie können aber auch innerhalb des Fahrzeugs zu finden sein.

[5] EMV: Elektromagnetische Verträglichkeit

Sicherheitsanforderungen mit der gewählten Architektur erfüllt werden können. Zudem enthält die Liste die den Komponenten (Hardware und Software) zugeordneten Sicherheitsanforderungen (quantitativ und qualitativ).

5.6. Impl - Implementierung

Das Ergebnis des *System-Entwurfs* und der *Entwurfsbegleitenden System-Sicherheitsbewertung* sind konkrete, detaillierte Vorgaben für die *Implementierung*. Das System wird diesen Vorgaben folgend umgesetzt, d. h. die Hardware des Systems wird konstruiert bzw. der Software-Code wird erzeugt.

Die *Implementierung* ist in der Praxis ein sehr kritischer Schritt, daher muss mit entsprechender Sorgfalt vorgegangen werden. In der Regel wird sie durch zahlreiche spezialisierte Werkzeugketten unterstützt. Eine Besonderheit ist dabei die Möglichkeit der Integration eines kompletten Hardware-Software-Systems auf einem einzigen Chip (ein sogenanntes „System-on-Chip“, kurz SoC). Prinzipiell ist dabei nicht relevant, ob eine entsprechende Funktion in Hardware oder Software umgesetzt wird.

In Abbildung 5.13 auf der nächsten Seite ist die Einordnung der *Implementierung* in den Gesamtprozess zu sehen, die Tabelle darunter erläutert die mit anderen Schritten ausgetauschten Informationen. In Abbildung 5.14 auf Seite 124 sieht man die Abfolge der einzelnen Schritte bei der *Implementierung*.

5.6.1. Impl 1 - Software-Implementierung

Im Rahmen dieses Schrittes sind die Softwaremodule der jeweiligen Softwareeinheit zu realisieren und die Prozeduren zur Erzeugung der ausführbaren Dateien zu erstellen (in der Regel entsprechende Compile-, Binde- und Generierprozeduren). Der Code wird dabei Selbstprüfungen des Entwicklers unterzogen.

Bei der Codierung der Software-Module muss die Programmiervorgabe (Pseudocode, Spezifikationssprache, ASCET-SD o. ä.) in Anweisungen der Implementierungssprache (Programmiersprache, Abfragesprache, Scriptsprache usw.) umgesetzt werden. Das Ergebnis ist der übersetzte und gebundene Code.

Zur Codierung zählen die Arbeitsschritte Programmierung (unter Einhaltung entsprechender Standards und Richtlinien wie z. B. MISRA C, vgl. [MIS04]), Erstellung von Compileprozeduren, Bindeprozeduren, Ladeprozeduren, Installationsprozeduren und Generierprozeduren, Durchführung von Compile- und Bindeläufen sowie Korrekturen bis zur Fehlerfreiheit des Compilierens und Bindens.

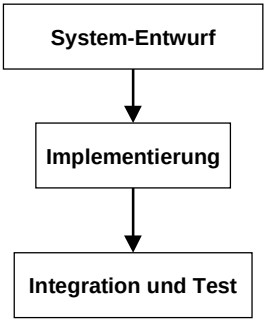

Abbildung 5.13.: Informationsfluss bei der Implementierung

von	Information	nach
System-Entwurf	Komplette Systembeschreibung mit System-Architektur, Hardware-/Software-Anforderungen, Schnittstellen	-
-	Implementiertes System	Integration und Test

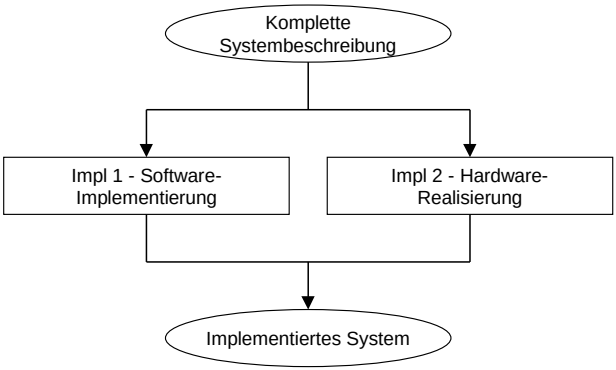

Abbildung 5.14.: Abwicklung der Implementierung

Neben der eigentlichen Entwicklung sind vom Entwickler auch Selbstprüfungen der von ihm realisierten Softwaremodule durchzuführen. Nach der Durchführung der Prüfung erfolgt die Auswertung der Prüfergebnisse, die den weiteren Entwicklungsverlauf bestimmt. Sofern die Prüfung erfolgreich war, erfolgt die Weitergabe des Prüfgegenstandes an den nächsten Schritt *Integration und Test* zum Zweck der formalen Produktprüfung.

5.6.2. Impl 2 - Hardware-Realisierung

Bei der Hardware-Realisierung werden gemäß den Unterlagen aus den Entwurfsaktivitäten die entsprechenden Hardwarekomponenten und Hardwaremodule angefertigt bzw. komplettiert.

Die Hardwarekomponenten bzw. die Hardwaremodule werden gemäß den Bauplänen und Entwicklungsunterlagen angefertigt oder extern in Auftrag gegeben. Kaufteile werden gegebenenfalls entsprechend angepasst und eingebaut. Hierbei getroffene Festlegungen sowie Erkenntnisse, die während der Realisierung der Hardwareeinheit entstehen, sind zum Zweck der späteren Verwendung z. B. bei Installations- oder Integrationsarbeiten zu dokumentieren. Die Beschaffung von im eigenen Betrieb gefertigten oder zugekauften Teilen erfolgt in enger Zusammenarbeit von Entwicklung, Fertigung und weiteren betroffenen Stellen wie z. B. Einkauf, Controlling und Qualitätsmanagement.

Auch die Hardwarekomponenten bzw. Hardwaremodule werden Selbstprüfungen durch den Entwickler unterzogen.

Dieser Schritt regelt die Realisierung beispielsweise von Funktions- und Erprobungsmustern. Sie betrifft aber nicht die Realisierung im Sinne einer Produktion (d. h. Serienfertigung), die nicht im Betrachtungsbereich der Darstellung hier liegen kann.

5.7. IT - Integration und Test

Nach der *Implementierung* des Systems erfolgt die *Integration* der einzelnen Systemkomponenten und der funktionale *Test*. Die Software und Hardware des Systems werden entsprechend den jeweiligen Integrationsspezifikationen integriert und dann gegen ihre funktionale Spezifikation getestet. Dann werden die Systemteile zum System integriert und iterativ getestet. Durch Wiederholen der Integrationsschritte entsteht so nach und nach das komplette System.

Die beim Test verwendeten Methoden sind Verifikation (d. h. die Überprüfung der Fragestellung, ob das System korrekt entwickelt wurde, die Überprüfung erfolgt gegen die Vorgaben aus dem *System-Entwurf*) und Validierung (d. h. die Überprüfung der Fragestellung, ob ein korrektes System entwickelt wurde, die Überprüfung erfolgt gegen die Vorgaben aus der *System-Anforderungsanalyse*).

Die Vorgehensweise in diesem Schritt wird von der *System-Sicherheitsbewertung* begleitet, bei der die Erfüllung der Sicherheitsanforderungen verifiziert und validiert wird.

Für die Verifikation bzw. Validierung von Elektroniksystemen im Automobil können unterschiedliche Methoden zur Anwendung kommen (nach [Hed01]). Grob wird dabei in Validierung durch Testen und in Formale Methoden unterschieden.

Eine der eingesetzten Testmethoden ist der White-Box-Test, auch Glass-Box-Test oder Strukturtest genannt. Dabei wird das betrachtete System primär anhand seiner Struktur betrachtet. Das System wird „gläsern“, man möchte alle relevanten Teile wie Verzweigungen oder Pfade prüfen. Bei dieser Testart bildet die vorliegende Implementierung des Prüflings die Grundlage des Testvorgangs.

Beim Black-Box-Test, auch spezifikationsorientierter Test genannt, bezieht sich die dynamische Prüfung auf die Funktionsvielfalt und Vollständigkeit, man blickt nicht in die Struktur der „Black Box“ hinein. Man beobachtet die Eingabe- und Ausgabegrößen der Testläufe und entscheidet dann entsprechend der Systemspezifikation, ob aufgrund der Ergebnisse eine Abnahme erfolgen kann.

Eine Mischung dieser beiden Tests wird Grey-Box-Test genannt.

Bei der Vorgehensweise in der Automobilindustrie werden meist funktionale Tests angewandt. Oft wird die Implementierung eines Systems durch einen Zu-

lieferer nicht offengelegt, so dass ein White-Box-Test nicht ganz unproblematisch ist. Vor allem für sicherheitsrelevante Systeme ist dieses Vorgehen nicht mehr ausreichend. Allein durch das funktionale Testen der Spezifikation können die Sicherheitsziele nicht gewährleistet werden. Aus diesem Grund werden heute schon die Automobilhersteller in den Hardware- und Software-Entwicklungsprozess bei den Zulieferern durch Reviews und gemeinsame FMEA- und FTA-Analysen eingebunden.

Formale Methoden kommen im Automobilbereich seltener zum Einsatz. Sie gliedern sich in Theorem-Beweise und Model-Checking.

Theorem-Beweise, auch deduktive Verifikation genannt, basieren darauf, dass Systemeigenschaften durch formale Deduktion unter Anwendung eines Satzes von Inferenzregeln bewiesen werden. Computerunterstützte Werkzeuge, die diese Technik anwenden, die sogenannten Theorem-Beweiser, erfordern jedoch i. A. die Interaktion mit einem Experten. Die Beweisführung kann oft schwierig und langwierig sein. Es gibt dabei keine Einschränkung hinsichtlich der Systemgröße und der Systemeigenschaften. Die Beweise verlaufen meist interaktiv, eine vollständige Automatisierung ist nicht möglich. Allerdings ist der Zeitaufwand für industrielle Anwendungen sehr hoch und praktisch sind Theorem-Beweise nur für kleine Software-Algorithmen und mittelgroße Hardwarekomponenten durchführbar. Beispiele für Spezifikationssprachen für Theorem-Beweiser sind B, Coq, FSE, VDM oder Z.

Beim Model-Checking (auch algorithmische Verifikation genannt) wird anhand eines Algorithmensatzes geprüft, ob ein gegebenes Systemmodell ein bestimmtes Verhalten bzw. eine bestimmte zeitliche Bedingung erfüllt. Model-Checking ist für reaktive Systeme einsetzbar, die sich auf endliche Automaten abbilden lassen. Der Prozess ist vollständig automatisierbar. Allerdings muss der Zustandsraum endlich sein und kontinuierliche Zustandsvariablen können nicht (bzw. eingeschränkt bei Diskretisierung) betrachtet werden. Beispiele für Beschreibungssprachen für Model-Checking-Werkzeuge sind ASA+, Esterel, Estelle, LDS, Lustre, MEC, Petri-Netze, SIS, SMV, Statecharts oder VSE.

Gemäß [Hed01] sind Theorem-Beweise zu speziell, als dass sie in naher Zukunft breite industrielle Anwendung finden werden. Für Model-Checking sind bereits kommerzielle Werkzeuge verfügbar und im Einsatz. Auch wenn Model-Checking nur bedingt auf kontinuierliche Softwarekomponenten anwendbar ist, so werden Model-Checker eine wachsende Bedeutung in industriellen Anwendungen finden.

Für eine umfassende Übersicht zu Formalen Methoden und den dabei eingesetzten Werkzeugen sei auf [Inf04] verwiesen.

In Abbildung 5.15 auf der nächsten Seite ist die Einordnung von *Integration und Test* in den Gesamtprozess zu sehen, die Tabelle darunter erläutert die mit anderen

Schritten ausgetauschten Informationen. In Abbildung 5.16 sieht man die Abfolge der einzelnen Schritte bei *Integration und Test*.

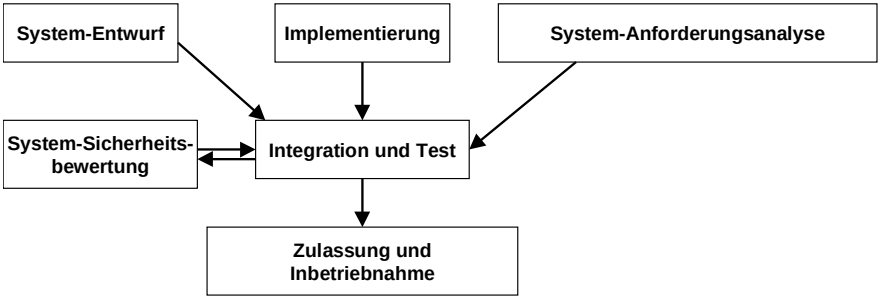

Abbildung 5.15.: Informationsfluss bei Integration und Test

von	Information	nach
Implementierung	Implementiertes System	-
System-Anforderungsanalyse	Spezifikation der Validierung	-
System-Entwurf	Integration- und Verifikations-Spezifikation	-
-	Schrittweise integriertes und funktional getestetes System	System-Sicherheitsbewertung
System-Sicherheitsbewertung	Rückmeldung, ob Anforderungen bzgl. Sicherheit und Zuverlässigkeit erfüllt werden inkl. Hinweisen zur Systemverbesserung hinsichtlich Sicherheit und Zuverlässigkeit	-
-	Integriertes und getestetes System	Zulassung und Inbetriebnahme

5.7.1. IT 1 - Software-Integration und -Test

In diesem Schritt erfolgt die Integration der Softwaremodule zur Softwareeinheit. Als Zwischenstufen sind Teilstrukturen der Softwareeinheit möglich, die sich durch Integration von Softwaremodulen, Softwarekomponenten und bereits integrierter Teilstrukturen ergeben.

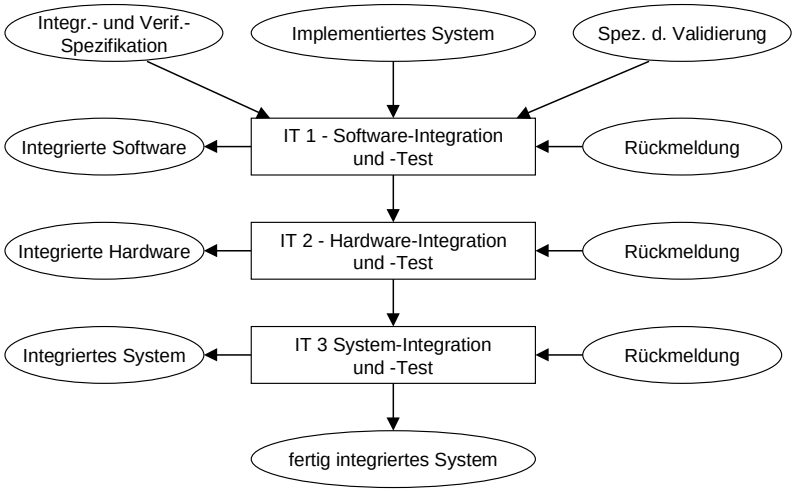

Abbildung 5.16.: Abwicklung von Integration und Test

Software-Integration und -Test gliedert sich in vier Teilschritte.

Integration zur Softwarekomponente Eine Software-Teilstruktur kann ein oder mehrere Softwaremodule oder Softwarekomponenten umfassen. Darüber hinaus kann eine Software-Teilstruktur Platzhalter enthalten. Diese Platzhalter werden bei wiederholtem Durchlauf der Integration zur Softwarekomponente, spätestens jedoch bei der Integration zur Softwareeinheit, durch die entsprechenden operationellen Softwaremodule ersetzt. Bei jeder Durchführung dieser Aktivität wird aus einer oder mehreren vorgegebenen Software-Teilstrukturen eine neue Software-Teilstruktur höheren Integrationsgrades hergestellt.

Zu Integrationszwecken kann es erforderlich sein, dass zusätzlich zu den vorhandenen Softwaremodulen Code erstellt oder Prozeduren geschrieben werden müssen.

Die Softwarekomponenten sind anschließend einzeln der Selbstprüfung zu unterziehen. Hierbei darf die Softwarekomponente noch Platzhalter enthalten.

Die Anordnung der Softwaremodule und Softwarekomponenten in der Erzeugnisstruktur legt eine Bottom-up-Integrationsstrategie nahe. Dies ist jedoch nur eine unter mehreren möglichen Vorgehensweisen. Weitere Beispiele für

Integrationsstrategien sind Top-down-Integration, „Sandwich“-Integration, vorgezogene Integration der kritischen und wichtigen Funktionen, Integrationsfolge in Anlehnung an organisatorische Gegebenheiten und Zuständigkeiten, Integration orientiert an den Funktionen oder Prozessen der Softwareeinheit, vorgezogene Integration des Basissystems (alle zentralen, mehrfach genutzten Funktionen) sowie Integration entsprechend der Abhängigkeit der Integrationsobjekte.

Die Software-Teilstruktur tritt nur vorübergehend, nämlich während der Software-Integration, in Erscheinung. Sie ist eine Hilfskonstruktion, die die Software-Integration nach beliebigen Strategien ermöglicht. Am Ende der Software-Integration steht die Softwareeinheit mit ihrer Untergliederung in Softwarekomponenten und Softwaremodule.

Die einzelnen Schritte der Software-Integration sind in der Integrationsspezifikation festgelegt.

Test und Verifikation der Softwarekomponente Vom Entwickler werden Prüfungen der Softwarekomponenten durchgeführt und ihre Funktion gegen ihre Spezifikation verifiziert.

Integration zur Softwareeinheit Hier werden Softwaremodule und Softwarekomponenten zur Softwareeinheit integriert. Dies erfolgt analog zu „Integration zur Softwarekomponente“ und kann nach unterschiedlichen Vorgehensweisen (festgelegt in der Integrationsspezifikation) geschehen. Die resultierende Softwareeinheit muss frei von Platzhaltern sein.

Test und Verifikation der Softwareeinheit Vom Entwickler werden Prüfungen der Softwareeinheiten durchgeführt und ihre Funktion gegen ihre Spezifikation verifiziert.

Die Verifikation einer Softwareeinheit sollte insbesondere die Prüfung der Funktionalität der Softwareeinheit, die Prüfung der Zuverlässigkeit, der Effizienz und des Realzeitverhaltens der Softwareeinheit, die Prüfung der Benutzbarkeit, Änderbarkeit und Übertragbarkeit, die Prüfung der Software-internen Schnittstellen zwischen Softwaremodulen, Softwarekomponenten, Prozessen sowie die Prüfung der Software-externen Schnittstellen umfassen.

5.7.2. IT 2 - Hardware-Integration und -Test

Im Rahmen der Integration wird die Hardwareeinheit aus einzelnen Hardwarekomponenten, Hardwaremodulen und sonstigen Bauteilen zusammengeführt. Außerdem wird die spezifizierte Funktion erprobt und die Hardwareeinheit einer Verifikation unterzogen.

Integration zur Hardware-Teilstruktur Entsprechend den Vorgaben der Integrationsspezifikation werden Hardwarekomponenten und Teilstrukturen zusammengefügt.

Test und Verifikation der Hardware-Teilstruktur Bei komplexen funktionalen Strukturen werden Teilstrukturen geprüft. So werden beispielsweise mechanische Teile auf Toleranzen an Funktionsmaßen oder elektronische Logikbausteine wie PALs[6] oder ASICs[7] funktionell gegen die Vorgaben geprüft.

Integration zur Hardwareeinheit Entsprechend den Vorgaben der Integrationsspezifikation werden einzelne Bauteile, Hardwaremodule und die verschiedenen Hardwarekomponenten zu einer Hardwareeinheit zusammengefügt.

Test und Verifikation der Hardwareeinheit Bei der schrittweisen Inbetriebnahme von komplexen Hardwareeinheiten sind Integration und Verifikation eng verknüpft. Teilfunktionen werden schrittweise erweitert und jeweils einer Verifikation durch den Entwickler unterzogen. Die Erfüllung der Anforderungen oder Abweichungen werden protokolliert.

5.7.3. IT 3 - System-Integration und -Test

Nach der Integration der Software und der Hardware erfolgt die Integration der einzelnen Systemkomponenten und der funktionale Test des gesamten Systems. Alle Systemkomponenten werden dann gegen ihre Spezifikation getestet und ihre korrekte Funktion verifiziert und validiert. Durch iteratives Wiederholen der Integrationsschritte entsteht so nach und nach das komplette System.

Begleitet wird diese Vorgehensweise von der *System-Sicherheitsbewertung*, mit deren Hilfe die Erfüllung der Sicherheitsanforderungen verifiziert wird.

[6] PAL: Programmable Array Logic
[7] ASIC: Application Specific Integrated Circuit

Die Integration des Systems aus Software- und Hardwareeinheiten erfolgt gemäß der festgelegten Systemarchitektur und unter Einhaltung der in der Integrationsspezifikation getroffenen Regelungen. Ein System kann vorübergehend Platzhalter für noch nicht vorhandene Elemente enthalten. Ein solches System wird teilintegriert genannt. Dies bedeutet, dass die Aktivitäten der System-Integration gegebenenfalls mehrfach durchlaufen werden, solange bis alle Platzhalter im System ersetzt sind.

Die System-Integration gliedert sich in zwei Teilschritte.

Integration zum System In diesem Schritt werden die Softwareeinheiten, Hardwareeinheiten und gegebenenfalls weitere Systembestandteile zum System integriert. Es ist dabei gemäß der Integrationsspezifikation vorzugehen, die die durchzuführenden Integrationsmaßnahmen beschreibt.

Unter Umständen werden die spezifizierten Architekturelemente zunächst zu Segmenten integriert. Falls Segmente vorgesehen sind, sind Verifikationsaktivitäten auch für Segmente durchzuführen.

Falls eine Verifikation erforderlich ist, obwohl noch nicht alle Elemente für die Integration bereitstehen, so sind für die fehlenden Elemente Platzhalter vorzusehen.

Verifikation und Validierung des Systems Es ist die Verifikation und die Validierung des Systems und gegebenenfalls der Segmente durchzuführen. In dieser Aktivität sind die Softwareeinheiten auf der definierten Zielhardware zu überprüfen.

Bei der Verifikation sind zunächst Prüfvoraussetzungen wie die Installation des Systems in der Prüfumgebung und Festlegung der benötigten Prüffälle herzustellen. Nach der Durchführung der Prüfung erfolgt die Auswertung der Prüfergebnisse, die die weitere Vorgehensweise bestimmt.

Dabei sollte die Verifikation folgendes umfassen:

- Prüfung gegen die Anwenderforderungen und die Systemarchitektur
- Prüfung zunächst in einer speziellen Prüfumgebung (z. B. Prototyp); die abschließende Systemprüfung findet in einer realitätsnahen Einsatzumgebung statt, die aber auch simuliert werden kann (beispielsweise in Form einer Hardware-in-the-Loop-Simulation).
- Prüfung des technischen und funktionalen Zusammenwirkens der Systembestandteile mit dem Hauptaugenmerk auf Systembelastung hinsichtlich Effizienz und Robustheit, Verhalten in „Worst-case“-Bedingungen, Demonstration

der Systemsicherheit, typische Situationen und Bedingungen sowie kritische Situationen und Funktionen.

- Prüfung des Zusammenwirkens von System und Nutzer hinsichtlich typischer Situationen und ihrer Bedienung sowie kritischer Situationen und der Auswirkung von Fehlbedienung.

Abschließend wird die korrekte Funktion des betrachteten Systems gegen die Anforderungen der *System-Anforderungsanalyse* validiert.

5.8. SSB - System-Sicherheitsbewertung

Die *System-Sicherheitsbewertung* orientiert sich in ihrer Vorgehensweise stark an die der *Entwurfsbegleitenden System-Sicherheitsbewertung*.

Parallel zum Schritt *Integration und Test* wird bei der *System-Sicherheitsbewertung* überprüft, ob das System die an es gestellten Sicherheitsanforderungen erfüllt. Dies geschieht durch Verifikation der System-Sicherheitsanforderungen aus der *Funktionalen Gefährdungsanalyse*, durch Betrachtung der Gefährdungseinstufungen aus der *Funktionalen Gefährdungsanalyse* und durch Verifikation und Validierung weiterer Sicherheitsanforderungen.

Der funktionale Anteil der Systemanforderungen wie auch der nicht-funktionale aber nicht-sicherheitsrelevante Aspekt des Systems wird im Schritt *Integration und Test* überprüft. Sicherheitsrelevante Systemaspekte werden hier in der *System-Sicherheitsbewertung* untersucht.

Vom Grundprinzip ist die Vorgehensweise bei der *System-Sicherheitsbewertung* sehr ähnlich derjenigen bei der *Entwurfsbegleitenden System-Sicherheitsbewertung*. Es werden die selben Werkzeuge verwendet, der Hauptunterschied ist, dass jetzt nicht mehr der Entwurf des Systems, sondern das System selbst in physikalisch implementierter, realer Form vorliegt. Es wird weiterhin der selbe Fehlerbaum wie in der *Entwurfsbegleitenden System-Sicherheitsbewertung* verwendet, nur kommen die Fehlerraten nun nicht mehr aus Datenbanken, sondern werden aus Erfahrungen mit dem realen System gewonnen.

Wie schon beim Vorgehen in den Schritten *System-Entwurf* und der parallelen *Entwurfsbegleitenden System-Sicherheitsbewertung* wird hier iterativ zwischen *Integration und Test* und der *System-Sicherheitsbewertung* gewechselt. Das System wird integriert und funktional getestet, parallel erfolgt der Nachweis der Sicherheitseigenschaften. Dafür müssen die Systemteile, mit denen eine bestimmte

Funktion erbracht wird, die entsprechenden quantitativen Anforderungen aus der Funktionalen Gefährdungsanalyse erfüllen.

Die Überprüfung geschieht durch systematische Untersuchung des Systems, seiner Architektur und seiner Realisierung durch Methoden der Verifikation und Validierung wie z. B. Testen, Fehlerinjektion oder Formale Methoden. So kann nachgewiesen werden, dass das System die gestellten Sicherheitsanforderungen erfüllt.

In Abbildung 5.17 auf der nächsten Seite ist die Einordnung der *System-Sicherheitsbewertung* in den Gesamtprozess zu sehen, die Tabelle darunter erläutert die mit anderen Schritten ausgetauschten Informationen. Abbildung 5.18 auf Seite 135 stellt die Abfolge der einzelnen Schritte bei der *System-Sicherheitsbewertung* dar.

5.8.1. SSB 1 - Verifikation der Erfüllung der System-Sicherheitsanforderungen

In einem ersten Schritt wird verifiziert, dass die Sicherheitsanforderungen an das System, wie sie in der *System-Anforderungsanalyse* und der *Funktionalen Gefährdungsanalyse* formuliert wurden, erfüllt werden. Dabei wird der jeweilige im parallelen Schritt *Integration und Test* integrierte Systembestandteil untersucht. Die eingesetzten Methoden sind dabei grundsätzlich mit denen in Schritt ESSB 2 (vgl. Kapitel 5.5.2 auf Seite 118) identisch. Als zusätzliche Methode kommt noch Fehlerinjektion zum Einsatz.

Die Fehlerinjektion ist eine Methode, die Fehlertoleranz eines Systems unter Berücksichtigung des spezifizierten Verhalten zu untersuchen (nach [Hed01]). Fehlerinjektion wird benötigt, um den korrekten Ablauf der fehlertoleranten Algorithmen bzw. Mechanismen und um das Verhalten des Systems in einer realistischen Umgebung, in der das Auftreten von Fehlern erwartet wird und die Behandlung korrekt geschehen muss, vorherzusagen. Dabei werden unterschiedliche Arten der Fehlerinjektion unterschieden.

Bei der „Physikalische Fehlerinjektion“ erfolgt die Fehlerinjektion beispielsweise an den Pins eines Microchips („Pin-Level-Injection“), es erfolgt eine Bestrahlung des betrachteten Systems mit Schwerionen oder eine Injektion von elektromagnetischer Strahlung. Die physikalische Fehlerinjektion ist im allgemeinen mit einem hohen Aufwand verbunden.

Bei der „Software implementierten Fehlerinjektion“ (kurz SWIFI) werden Fehler systematisch durch Veränderung oder Ergänzung von Programmen erzeugt. Alle SWIFI-Methoden besitzen das unvermeidliche Problem, dass die Fehlerinjektion

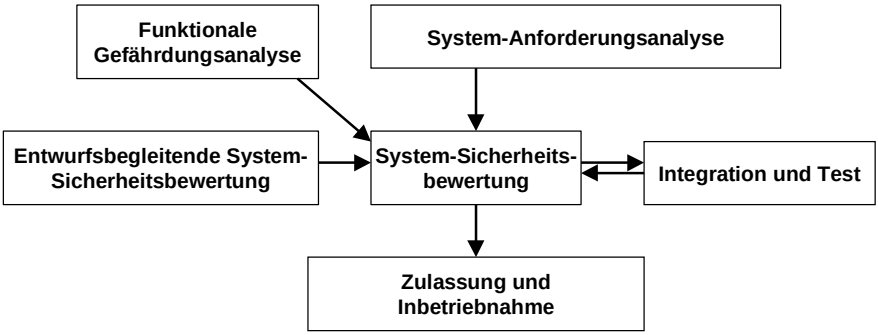

Abbildung 5.17.: Informationsfluss bei der System-Sicherheitsbewertung

von	Information	nach
Entwurfsbegleitende System-Sicherheitsbewertung	Ergebnisse der Entwurfsbegleitenden System-Sicherheitsbewertung	-
Funktionale Gefährdungsanalyse	Zugeordnete Anforderungen an Sicherheit und Zuverlässigkeit sowie Vorgehensweise zur Verifikation dieser Anforderungen	-
System-Anforderungsanalyse	Anforderungen bzgl. Sicherheit	-
Integration und Test	Schrittweise integriertes und funktional getestetes System	-
-	Rückmeldung, ob Anforderungen bzgl. Sicherheit und Zuverlässigkeit erfüllt werden inkl. Hinweisen zur Systemverbesserung hinsichtlich Sicherheit und Zuverlässigkeit	Integration und Test
-	Ergebnisse bzgl. Sicherheit und Zuverlässigkeit	Zulassung und Inbetriebnahme

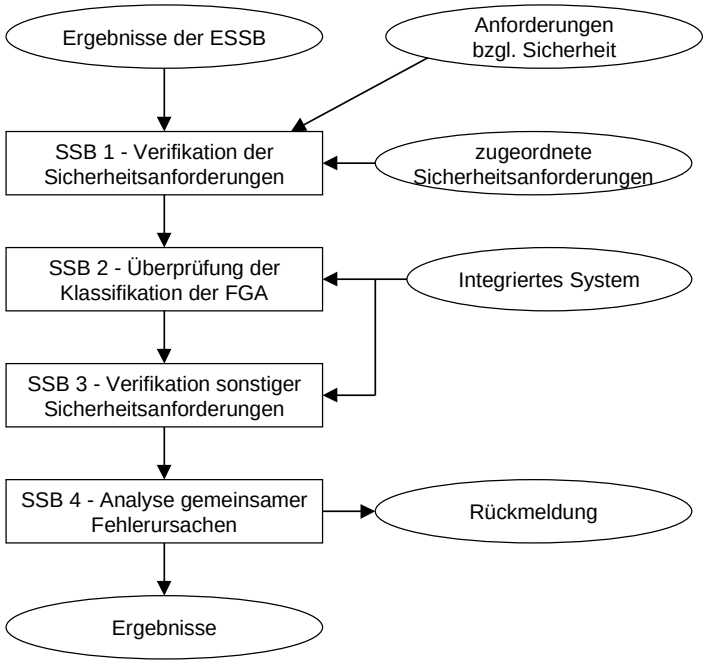

Abbildung 5.18.: Abwicklung der System-Sicherheitsbewertung

Rechenzeit benötigt, und somit das Verhalten des zu untersuchenden Rechnersystems verändert wird.

Bei der „Fehlerinjektion in Simulationsmodellen“ kann der Fehlertyp, -ort und -zeitpunkt frei bestimmt werden. Der Detaillierungsgrad des Modells setzt der Realitätstreue Grenzen.

In der Automobilindustrie wird Fehlerinjektion hauptsächlich im Rahmen des Tests zur elektromagnetischen Verträglichkeit eingesetzt (nach [Hed01]). Im Zuge der zunehmenden Vernetzung in Fahrzeugen werden auch verstärkt Fehler auf den Datenbussen eingestreut, um die Fehlerbehandlung und -rekonfiguration in den verteilten Systemen zu untersuchen.

5.8.2. SSB 2 - Überprüfung der Klassifikation der Gefährdungen in der FGA

Auf Basis der Ergebnisse im vorigen Schritt wird untersucht, ob die Klassifizierung der Gefährdungen für das System korrekt war. Gibt es mehrere signifikante Abweichungen der Ergebnisse von den Vorgaben aus der *Funktionalen Gefährdungsanalyse*, so müssen diese Vorgaben unter Umständen in Frage gestellt werden. Eine genaue Untersuchung und evtl. eine Anpassung der Anforderungen ist notwendig.

5.8.3. SSB 3 - Verifikation und Validierung sonstiger Sicherheitsanforderungen

Neben der Verifikation der funktionalen Sicherheitsanforderungen im Schritt SSB 1 werden bei der *System-Sicherheitsbewertung* auch weitere nicht-funktionale Sicherheitsanforderungen überprüft. Dazu gehören beispielsweise Sicherheitsanforderungen, die im Rahmen von Schritt ESSB 1 zusätzlich formuliert wurden (vgl. Kapitel 5.5.1 auf Seite 118).

5.8.4. SSB 4 - Analyse gemeinsamer Fehlerursachen

Ähnlich wie in Schritt ESSB 3 der *Entwurfsbegleitenden System-Sicherheitsbewertung* in Kapitel 5.5.3 auf Seite 120 werden im Rahmen der *System-Sicherheitsbewertung* mögliche gemeinsame Fehlerursachen analysiert und bewertet. Die Vorgehensweise ist dabei auch bei diesem Schritt in der *System-Sicherheitsbewertung* analog dem bei der *Entwurfsbegleitenden System-Sicherheitsbewertung*, daher sei für detaillierte Ausführungen auf Kapitel 5.5.3 verwiesen.

5.9. ZI - Zulassung und Inbetriebnahme

Am Ende des Entwicklungsprozesses im Automobil steht die *Typzulassung* und die *Inbetriebnahme* des Gesamtfahrzeugs. Beide Schritte, die Inbetriebnahme des entwickelten Elektroniksystems und die abschließende Typzulassung des Gesamtsystems im Fahrzeug, erfolgen in aller Regel beim Fahrzeughersteller.

Die Vorgehensweise bei der Typzulassung wird durch entsprechende Richtlinien festgelegt. Die relevanten Standards für die Zulassung von Automobilsystemen für den Straßenverkehr in Europa sind die entsprechenden europäischen Zulassungsvorschriften, so beispielsweise für Bremssysteme die Richtlinie UN ECE-R 13 (vgl. [UN 01a]) oder für Lenksysteme die Richtlinie UN ECE-R 79 (vgl. [UN 01b]).

Diese Richtlinien enthalten einen Elektronikanhang, der für die Zulassung von Elektroniksystemen in Kontext von Bremse bzw. Lenkung anzuwenden ist. In ihm werden Anforderungen an die im Fahrzeug für das Bremssystem bzw. die Lenkung eingesetzten Elektroniksysteme definiert, deren Erfüllung über ebenfalls in den Richtlinien definierte Verfahren abgeprüft wird.

Wesentliche Inhalte der Richtlinien sind u. a., dass bei der Typzulassung die Dokumentation des Sicherheitskonzepts vorgelegt werden muss; zudem darf die Zulassungsbehörde das korrekte Systemverhalten hinsichtlich Sicherheit und Zuverlässigkeit durch Verfahren wie z. B. Fehlerinjektion überprüfen.

In Abbildung 5.19 auf der nächsten Seite ist die Einordnung von *Zulassung und Inbetriebnahme* in den Gesamtprozess zu sehen, die Tabelle darunter erläutert die mit anderen Schritten ausgetauschten Informationen. In Abbildung 5.20 auf Seite 139 sieht man die Abfolge der einzelnen Schritte bei der *Zulassung und Inbetriebnahme*.

5.9.1. ZI 1 - Inbetriebnahme

Als letzter Schritt bei der Systementwicklung werden noch evtl. fehlende Integrationsschritte durchgeführt und das Gesamtsystem in Betrieb genommen. Dieses System wird erprobt und die korrekte Funktion bei den vorgesehenen Randbedingungen validiert. Zudem erfolgt hier die Integration ins Gesamtfahrzeug.

Zusätzlich werden Maßnahmen zur korrekten Wartung und Instandhaltung des Systems ermittelt und festgelegt.

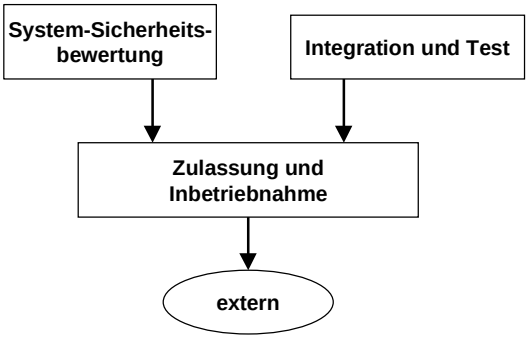

Abbildung 5.19.: Informationsfluss bei der Zulassung und Inbetriebnahme

von	Information	nach
Integration und Test	Integriertes und getestetes System	-
System-Sicherheitsbewertung	Ergebnisse bzgl. Sicherheit und Zuverlässigkeit	-
-	Komplett erstelltes, funktionierendes, sicheres, zuverlässiges und zugelassenes System	extern

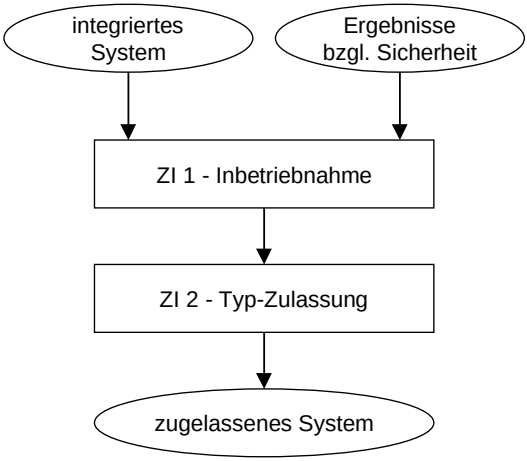

Abbildung 5.20.: Abwicklung der Zulassung und Inbetriebnahme

5.9.2. ZI 2: Typzulassung

Für die Betrachtung von sicherheitsrelevanten Elektroniksystemen sind insbesondere die Systeme Lenkung und Bremse interessant. Die Typzulassung solcher Systeme ist durch entsprechende Vorschriften geregelt, für Deutschland ist dies die Straßenverkehrs-Zulassungsordnung, bei der §41 die Zulassung von Bremssystemen und §38 die Zulassung von Lenksystemen regelt. Weltweit (die Gültigkeit ist dabei aber auf Europa, Japan und Australien beschränkt) werden die Zulassungsvorschriften von der „Economic Commission for Europe“ der Vereinten Nationen herausgegeben (nach [Dil02]).

Um dem zunehmenden Einsatz von Elektronik für diese Systeme Rechnung zu tragen wurde die Zulassungsvorschrift UN ECE Regulation 13 „Uniform provisions concerning the approval of vehicles of categories M, N and O with regard to braking “ für Bremssysteme (vgl. [UN 01a]) um einen entsprechenden Anhang „Special requirements to be applied to the safety aspects of complex electronic vehicle control systems“ zum Sicherheitsnachweis der elektronischen Bestandteile von Bremssystemen ergänzt. Für die UN ECE Regulation 79 „Uniform provisions concerning the approval of vehicles with regard to steering equipment“ für Lenksysteme (vgl. [UN 01b]) ist ein Anhang mit vergleichbarem Inhalt in Arbeit.

Dieser Anhang kann grundsätzlich auf alle sicherheitsrelevanten elektronischen Fahrzeugsysteme Anwendung finden. Damit erübrigt es sich, auch für elektronisch gesteuerte und auf die Bremsanlage wirkende Zusatzfunktionen besondere Vorschriften oder gar spezielle ECE-Regelungen zu erstellen

Diese Anhänge stellen besondere Anforderungen an das Sicherheitskonzept, die Dokumentation des Sicherheitskonzepts und die Verifikation des Sicherheitskonzepts, die dem prüfenden Technischen Dienst - in Deutschland ist dies in der Regel der TÜV[8] - zwecks Typzulassung solcher Systeme nachzuweisen sind.

Als Zitat findet sich in [UN 01a]:

> *„... the manufacturer shall provide a documentation package ... [in which] the safety concept ... shall be explained.“.*

Und weiter:

> *„... the reaction of „The System“ shall, at the discretion of the Type Approval Authority, be checked under the influence of a failure in any individual Unit by applying corresponding output signals to electrical Units or mechanical elements in order to simulate the effects of internal faults within the unit.“*

Dabei soll die Dokumentation in zwei Teilen zur Verfügung gestellt werden. Zum einen ein formaler Dokumentationsteil mit eingeschränktem Umfang, der dem prüfenden technischen Dienst übergeben wird, und zusätzliches vertrauliches Material, das der Prüfer bei der Typzulassung nur einsehen kann (nach [Dil02]).

Die Dokumentation beinhaltet eine Beschreibung der Entwicklungsmethodik und der verwendeten Werkzeuge, eine Systembeschreibung mit einer Liste der Eingangsgrößen, Liste der Ausgangsgrößen und deren Arbeitsbereiche, eine Beschreibung aller Komponenten mit deren Funktion, Verbindungen, Signalfluss und der Identifikation von Hard- und Software sowie eine Beschreibung des Sicherheitskonzepts des Herstellers.

Darüber hinaus muss der Nachweis erbracht werden, dass das Sicherheitskonzept eventuell auftretende Fehler tatsächlich erkennt und das System in den sogenannten sicheren Zustand überführt. Dieser Nachweis kann etwa durch Einsichtnahme des Technischen Dienstes in eine FMEA während der Typzulassung erbracht werden.

[8] TÜV: Technischer Überwachungsverein

Für die Dokumentation des Sicherheitskonzeptes kann die Dokumentation des *System-Entwurfs* und die Darstellung der Ergebnisse der *System-Sicherheitsbewertung* verwendet werden. Auch der Sicherheitsnachweis kann über die Ergebnisse der *System-Sicherheitsbewertung* erfolgen.

6. Beispiele

In diesem Kapitel wird die Vorgehensweise beim *Funktionalen Entwurf* und der *Funktionalen Gefährdungsanalyse* sowie beim *System-Entwurf* und der *Entwurfsbegleitenden System-Sicherheitsbewertung* anhand von zwei Beispielen dargestellt.

Grundlage ist eine Steer-by-Wire Anwendung, d. h. ein Lenksystem, bei dem die Fahrerwunscherfassung und die Umsetzung in eine Drehung der Vorderräder mechanisch getrennt sind, die Informations- und Energieübertragung erfolgt nur elektrisch.

In Abschnitt 6.1 wird im Rahmen des *Funktionalen Entwurfs* und der parallelen *Funktionalen Gefährdungsanalyse* das System beispielhaft funktional gegliedert, Gefährdungen ermittelt und basierend darauf die Funktionen in Sicherheitsstufen klassifiziert.

In Abschnitt 6.2 wird dann eine mögliche Systemarchitektur für ein Steer-by-Wire System untersucht und Teile davon durch eine beispielhafte *Entwurfsbegleitende System-Sicherheitsbewertung* bewertet.

Die beiden in diesem Kapitel dargestellten Beispiele sind auch nur als solche zu sehen. Es ist zu beachten, dass sie keine realen oder in Entwicklung befindlichen Systeme beschreiben. Rückschlüsse auf Anforderungen oder Architekturen realer Systeme aufgrund der Darstellung hier sind daher nicht möglich. Dies gilt insbesondere für quantitative Anforderungen.

6.1. Beispiel 1: Funktionaler Entwurf und Funktionale Gefährdungsanalyse

In diesem Abschnitt wird beispielhaft die Vorgehensweise beim *Funktionalen Entwurf* und der *Funktionalen Gefährdungsanalyse* an einem fiktiven „Steer-by-Wire“ System aufgezeigt.

Dazu wird der *Funktionale Entwurf* für ein beispielhaftes Lenksystem mit Hilfe der SA/RT-Methodik dargestellt (SA/RT vgl. Kapitel 3.7.1). Prinzipiell ist aber jede Methode, die zur Ermittlung der Systemfunktionen dient, für diesen Zweck verwendbar.

Für einige definierte Funktionen erfolgt basierend darauf eine *Funktionale Gefährdungsanalyse*. Dabei ist zu beachten, dass hier im Beispiel kein reales oder ein sich in der Entwicklung befindliches System dargestellt wird, sondern ein rein fiktives System. Weder die Systemarchitektur noch Details des Systems entsprechen einer tatsächlichen Realisierung, das System hier dient nur der Veranschaulichung der Vorgehensweise.

6.1.1. Beschreibung des betrachteten Systems, Randbedingungen, Anforderungen

Aus Gründen der Komplexität kann an dieser Stelle selbstverständlich keine umfassende *System-Anforderungsanalyse* durchgeführt werden. Im Nachfolgenden soll nur eine kurze Beschreibung des Beispielsystems erfolgen.

Zu entwickeln ist ein Lenksystem für einen Pkw. Beide Räder der Vorderachse werden gelenkt. Die Erfassung des Richtungswunsches des Fahrers erfolgt über ein herkömmliches Lenkrad, das Fahrzeug soll sich daraufhin in die gewünschte Richtung bewegen. Zusätzliche Signale beispielsweise von Fahrerassistenzsystemen sollen nicht berücksichtigt werden. Der Fahrer soll eine Rückmeldung in der Form erhalten, dass er ein „natürliches Lenkgefühl“ hat. Er soll Fahrbahnunebenheiten ebenso spüren wie Kurvenkräfte. Die Erfassung des Fahrerwunsches soll ebenso wie die Rückmeldung des Lenkmoments über das Lenkrad erfolgen. Zusätzlich soll es im Kombiinstrument des Fahrzeugs eine Leuchte geben, die bei Problemen im System den Fahrer warnt.

Das Lenksystem soll rein elektrisch bzw. elektronisch ausgeführt werden, d. h. die Informationsübertragung soll weder mechanisch noch hydraulisch oder pneumatisch erfolgen (sog. Steer-by-Wire Prinzip). Die Anordnung der Systembestandteile soll sich aus Sicht des Fahrers von der in einem heutigen Fahrzeug nicht unterscheiden.

Die technischen Randbedingungen des Systems orientieren sich an den Zulassungsvorschriften nach UN ECE R 79 (vgl. Kapitel 5.9).

6.1.2. Funktionaler Entwurf - Hierarchieebene 1

In einem ersten Schritt wird das System auf einer abstrakten Ebene funktional beschrieben und gegliedert.

6.1.2.1. FE 1 - Identifikation der Systemfunktionen

Erster Schritt beim *Funktionalen Entwurf* ist die Identifikation aller Systemfunktionen. Beim Vorgehen nach SA/RT entsteht dabei zunächst das Kontextdiagramm des Systems.

Das Kontextdiagramm der gesamten Lenkung zeigt Abbildung 6.1. Zentrales Element ist das Lenksystem, das mit dem Fahrer und dem Kombiinstrument interagiert. Der Fahrer teilt dem Lenksystem einen Richtungswunsch durch Einstellen eines Lenkwinkels (Drehen am Lenkrad) mit, der vom Lenksystem verarbeitet wird. Außerdem erhält er eine Rückmeldung über das an der Rädern anliegende Lenkmoment, das er als Lenkradmoment rückgemeldet bekommt. Zusätzliches Element ist das Kombiinstrument, auf dem evtl. auftretende Fehler des Systems dem Fahrer signalisiert werden.

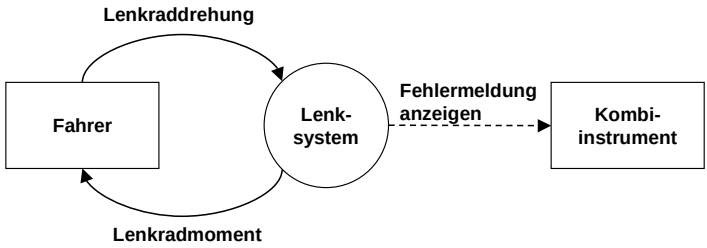

Abbildung 6.1.: Kontextdiagramm des Lenksystems

Basierend auf diesem Kontextdiagramm kann die betrachtete Systemfunktion *Lenksystem* weiter verfeinert werden. Das Flussdiagramm von *Lenksystem* findet sich in Abbildung 6.2 auf der nächsten Seite. Es ergeben sich zwei Funktionen, *Lenken* und *Lenkradmoment rückmelden*.

Lenken ist die primäre Funktion des Lenksystems. Dabei wird als Eingangsgröße der Informationsfluss *Lenkraddrehung* in Form des anliegenden Lenkwinkels verarbeitet. Entsprechend der *Lenkraddrehung* führt das System eine Lenkbewegung durch. Die Ausgangsgröße ist der Kontrollfluss *Fehlermeldung anzeigen*.

Die Funktion *Lenkradmoment rückmelden* dient dazu, dem Fahrer über den Informationsfluss *Lenkradmoment* eine haptische Rückmeldung der Fahrsituation zu geben. Auch hier gibt es wieder den Kontrollfluss *Fehlermeldung anzeigen*.

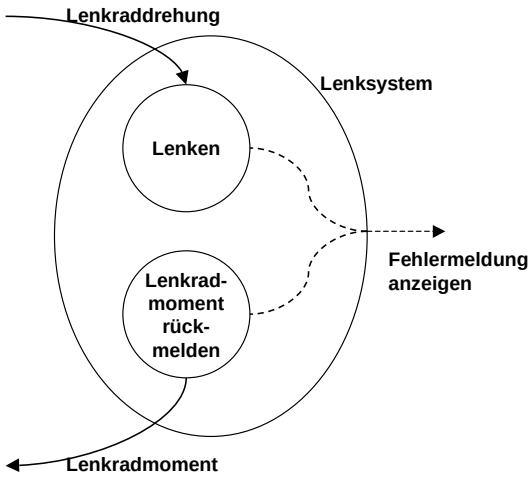

Abbildung 6.2.: Flussdiagramm von *Lenksystem* aus Abbildung 6.1 auf Seite 145

6.1.2.2. FE 2 - Erstellen einer Funktionsliste

Basierend auf dieser Darstellung erfolgt das Erstellen der Funktionsliste. Es ergibt sich die Funktionsliste nach Tabelle 6.1.

Tabelle 6.1.: Funktionsliste in der Hierarchieebene 1

Nr.	Funktion
1	Lenken
2	Lenkradmoment rückmelden

6.1.2.3. FE 3 - Wiederholung für niedrigere Hierarchieebenen

Anhand dieser Funktionsliste wird nun die *Funktionale Gefährdungsanalyse* durchgeführt. Danach wird der *Funktionale Entwurf* auf der nächsten Hierarchieebene wiederholt.

6.1.3. Funktionale Gefährdungsanalyse - Hierarchieebene 1

In diesem Schritt wird die *Funktionale Gefährdungsanalyse* für die identifizierten Funktionen *Lenken* und *Lenkradmoment rückmelden* durchgeführt.

Tabelle 6.3 auf Seite 149 und Tabelle 6.4 auf Seite 150 zeigen die komplette *Funktionale Gefährdungsanalyse* für die Funktionen *Lenken* und *Lenkradmoment rückmelden* und die in Tabelle 6.2 dargestellten Gefährdungen.

6.1.3.1. FGA 1 - Identifikation und Beschreibung möglicher Gefährdungen

Basierend auf diesen beiden Funktionen werden mögliche Fehlfunktionen und die damit verbundenen Gefährdungen ermittelt. Beispielsweise kann bei der Funktion *Lenken* der Fall auftreten, dass gar keine Lenkbewegung erfolgt. Zusätzlich kann das Fahrzeug selbsttätig um einen falschen Betrag oder in die falsche Richtung lenken.

Die aus den beiden Funktionen *Lenken* und *Lenkradmoment rückmelden* ermittelten Gefährdungen sind in Tabelle 6.2 auf der vorherigen Seite wiedergegeben.

Tabelle 6.2.: Ermittelte Gefährdungen

Nr.	Funktion	Gefährdung
1.A	Lenken	Fahrzeug lenkt nicht
1.B		Lenkung blockiert
1.C		Selbstlenker (d. h. Fahrzeug lenkt in eine Richtung bzw. mit einem Betrag, die nicht der Drehung am Lenkrad entsprechen)
2.A	Lenkradmoment rückmelden	es wird kein Lenkradmoment rückgemeldet
2.B		Lenkrad dreht sich selbst
2.C		Lenkrad blockiert

Es ist sinnvoll, für die Betrachtung der Gefährdungen die Situation zu berücksichtigen, in der sich das Fahrzeug im konkreten Fall befindet. In diesem Beispiel werden nur zwei solche Randbedingungen betrachtet: der Fall, dass das Lenkrad momentan gedreht wird (d. h. die Lenkfunktion wird verwendet), und der Fall, dass das Lenkrad nicht gedreht wird (d. h. die Lenkfunktion wird nicht verwendet).

Unter Berücksichtigung dieser zusätzlichen Randbedingungen ergibt sich die vierte Spalte in Tabelle 6.3 auf der nächsten Seite bzw. Tabelle 6.4 auf Seite 150.

6.1.3.2. FGA 2 - Ermittlung der Fehlerauswirkungen

Basierend auf der Darstellung in Tabelle 6.2 können den Gefährdungen nun mögliche Fehlerauswirkungen zugeordnet werden. Das Ergebnis fließt wieder in die Tabelle ein, es kommt die fünfte Spalte in Tabelle 6.3 bzw. Tabelle 6.4 hinzu.

6.1.3.3. FGA 3 - Klassifikation der ermittelten Gefährdungen

Unter Verwendung des Risikographen aus Abbildung 5.5 auf Seite 91 können die Gefährdungen anhand der möglichen Auswirkungen in Sicherheitsstufen klassifiziert werden. Für die jeweils betrachtete Funktion ergibt sich in Summe die höchste Sicherheitsstufe, die einer der Gefährdungen zugeordnet wurde, die durch die jeweilige Funktion bedingt werden kann.

Tabelle 6.3 und Tabelle 6.4 zeigen die jeweils zugeordneten Klassifizierungen, in Summe ergibt sich für *Lenken* eine Einstufung nach SF 4 und für *Lenkradmoment rückmelden* eine Einstufung nach SF 3.

6.1.3.4. FGA 4 - Ableitung von quantitativen bzw. qualitativen Anforderungen

Nach Tabelle 5.2 auf Seite 97 können den in Sicherheitsstufen klassifizierten Gefährdungen und damit auch den jeweiligen Funktionen quantitative Zuverlässigkeitsanforderungen zugeordnet werden. Die Voraussetzung für die Zuordnung ist hier beim Beispielsystem gegeben.

Daraus ergibt sich für die Funktion *Lenken* (SF 4) eine maximale Fehlerrate von $\lambda_{Lenken} \leq 10^{-9}/h$ und für die Funktion *Lenkradmoment rückmelden* (SF 3) eine maximale Fehlerrate von $\lambda_{Lenkmoment} \leq 10^{-8}/h$.

6.1.3.5. FGA 5 - Spezifikation von Methoden für die Verifikation

Für die Verifikation der Sicherheitsanforderungen an die Funktionen *Lenken* und *Lenkradmoment rückmelden* ist die Erfüllung der in FGA 4 festgelegten Sicherheitsanforderungen nachzuweisen. Dies kann beispielsweise in Form eines Fehlerbaums geschehen.

6.1.3.6. FGA 6 - Vergleich mit bisherigen Erfahrungen

Entfällt im Beispiel.

6.1.3.7. FGA 7 - Wiederholung für niedrigere Hierarchieebenen

In einem nächsten Schritt wird das System im Rahmen des *Funktionalen Entwurfs* weiter verfeinert, dann erfolgt eine erneute *Funktionale Gefährdungsanalyse* für die nächste Hierarchieebene.

Dabei ist zu berücksichtigen, dass mindestens eine Subfunktion der Funktion *Lenken* nach Sicherheitsstufe SF 4 zu klassifizieren ist, für die Subfunktionen von *Lenkradmoment rückmelden* gilt selbiges für Sicherheitsstufe SF 3.

Tabelle 6.3.: Funktionale Gefährdungsanalyse der Funktion *Lenken*

Nr.	Funktion	Gefährdung	Randbedingung	Auswirkungen	Klass.
1.A.1	Lenken	Fahrzeug lenkt nicht	Lenkfunktion wird momentan verwendet	Tod oder schwere Verletzung einer oder mehrerer Personen möglich, Situation ist dabei nicht beherrschbar, Auftreten im Regelfall	SF 4
1.A.2			Lenkfunktion wird momentan nicht verwendet	Tod oder schwere Verletzung einer oder mehrerer Personen möglich, Situation ist dabei nicht beherrschbar, Auftreten im Einzelfall	SF 3
1.B.1		Lenkung blockiert	Lenkfunktion wird momentan verwendet	Tod oder schwere Verletzung einer oder mehrerer Personen möglich, Situation ist dabei nicht beherrschbar, Auftreten im Regelfall	SF 4
1.B.2			Lenkfunktion wird momentan nicht verwendet	Tod oder schwere Verletzung einer oder mehrerer Personen möglich, Situation ist dabei nicht beherrschbar, Auftreten im Einzelfall	SF 3
1.C.1		Selbstlenker	Lenkfunktion wird momentan verwendet	Tod oder schwere Verletzung einer oder mehrerer Personen möglich, Situation ist dabei nicht beherrschbar, Auftreten im Regelfall	SF 4
1.C.2			Lenkfunktion wird momentan nicht verwendet	Tod oder schwere Verletzung einer oder mehrerer Personen möglich, Situation ist dabei nur schwer beherrschbar, Auftreten im Regelfall	SF 4

Tabelle 6.4.: Funktionale Gefährdungsanalyse der Funktion *Lenkradmoment rückmelden*

Nr.	Funktion	Gefährdung	Randbedingung	Auswirkungen	Klass.
2.A.1	Lenkrad-moment rückmel-den	es wird kein Lenkradmo-ment rückgemeldet	Lenkfunktion wird momentan verwendet	Irritation des Fahrers, dabei im Einzelfall Unfall mit Tod oder schwerer Verletzung einer oder mehrerer Personen möglich, Situation in der Regel nur schwer beherrschbar	SF 2
2.A.2			Lenkfunktion wird momentan nicht verwendet	Irritation des Fahrers, dabei im Einzelfall Unfall mit Tod oder schwerer Verletzung einer oder mehrerer Personen möglich, Situation in der Regel nur schwer beherrschbar	SF 2
2.B.1		Lenkrad dreht sich selbst	Lenkfunktion wird verwendet	Beeinflussung des Fahrers, dabei Unfall mit Tod oder schwerer Verletzung einer oder mehrerer Personen möglich, Situation in der Regel nur schwer beherrschbar	SF 3
2.B.2			Lenkfunktion wird momentan nicht verwendet	Beeinflussung des Fahrers, dabei Unfall im Einzelfall mit Tod oder schwerer Verletzung einer oder mehrerer Personen möglich, Situation in der Regel nur schwer beherrschbar	SF 2
2.C.1		Lenkrad blockiert	Lenkfunktion wird verwendet	Beeinflussung des Fahrers, dabei Unfall mit Tod oder schwerer Verletzung einer oder mehrerer Personen möglich, Situation in der Regel nur schwer beherrschbar	SF 3
2.C.2			Lenkfunktion wird momentan nicht verwendet	Beeinflussung des Fahrers, dabei Unfall im Einzelfall mit Tod oder schwerer Verletzung einer oder mehrerer Personen möglich, Situation in der Regel nur schwer beherrschbar	SF 2

6.1.4. Funktionaler Entwurf - Hierarchieebene 2

Die funktionale Gliederung des Systems aus Abbildung 6.2 auf Seite 146 wird in diesem Schritt eine Hierarchieebene tiefer weiter detailliert.

6.1.4.1. FE 1 - Identifikation der Systemfunktionen

Im vorigen Schritt wurden für das *Lenksystem* die beiden Grundfunktionen *Lenken* und *Lenkradmoment rückmelden* ermittelt. Diese beiden Funktionen werden nun in diesem Schritt weiter verfeinert.

Lenken Die Funktion *Lenken* gliedert sich in mehrere Unterfunktionen. Als Eingangssignal dient der Informationsfluss *Lenkraddrehung*. Diese Information wird in einer Funktion *Lenkradwinkel erfassen* aufgenommen und entsprechend aufbereitet. Als *Lenkwinkel* wird die Information an *Raddrehung ermitteln* weitergegeben. Hier wird ermittelt, wie die Räder gedreht werden müssen, um dem Wunsch des Fahrers zu entsprechen. Ein entsprechender *Radwinkel* ist der Informationsfluss an die Funktion *Räder drehen*. Hierbei werden die Räder des Fahrzeugs entsprechend den Vorgaben in eine bestimmte Richtung gedreht.

Eine vierte Funktion ist *Fehlermanagement „Lenken“*. Hier werden *Fehlermeldungen* in Form von Kontrollflüssen gesammelt, aufbereitet, und bei Bedarf als Kontrollfluss *Fehlermeldung anzeigen* an die übergeordnete Funktion ausgegeben.

Es ergeben sich somit für *Lenken* die Subfunktionen *Lenkradwinkel erfassen*, *Raddrehung ermitteln*, *Räder drehen* und *Fehlermanagement „Lenken“*.

Das Flussdiagramm von *Lenken* findet sich in Abbildung 6.3.

Lenkradmoment rückmelden Die Funktion *Lenkradmoment rückmelden* gliedert sich ebenfalls in vier Unterfunktionen. Ebenso wie bei *Lenken* gibt es eine zentrale Funktion, hier *Fehlermanagement „Moment“* genannt. Dort werden *Fehlermeldungen* aus den anderen drei Funktionen gesammelt, aufbereitet, und in Form des Kontrollflusses *Fehlermeldung anzeigen* weitergegeben.

Ausgangssignal der Funktion *Lenkradmoment rückmelden* ist der Informationsfluss *Lenkradmoment*. Dieser wird von einer Unterfunktion *Lenkradmoment aufbringen* erzeugt, hier wird also das am Lenkrad anliegende Moment generiert. Die entsprechenden Momentenanforderung erhält diese funktionale Einheit in Form einer *Momenentenanforderung* von der Funktion *Lenkradmoment ermitteln*. Diese zentrale Funktion erhält als Eingangssignal das *Radmoment*, das von der Funktion *Radmoment erfassen* an den Rädern ermittelt wurde.

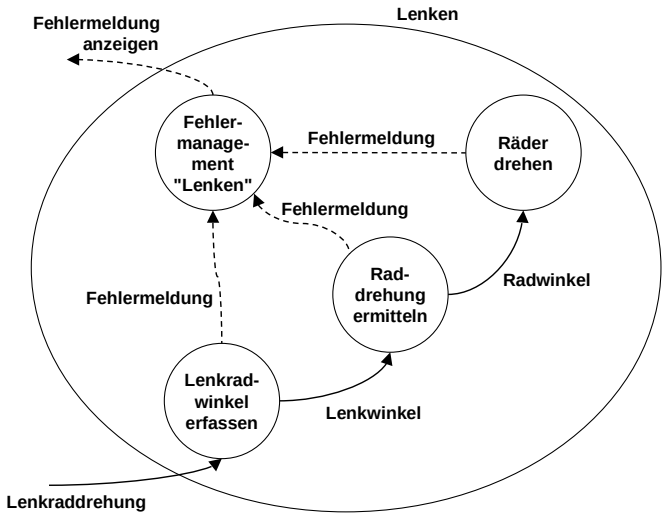

Abbildung 6.3.: Flussdiagramm der Funktion *Lenken* aus Abbildung 6.2 auf Seite 146

Es ergeben sich somit für *Lenkradmoment rückmelden* die Subfunktionen *Radmoment erfassen*, *Lenkradmoment ermitteln*, *Lenkradmoment aufbringen* und *Fehlermanagement „Moment“*.

Das Flussdiagramm von *Lenkradmoment rückmelden* findet sich in Abbildung 6.4 auf Seite 153.

6.1.4.2. FE 2 - Erstellen einer Funktionsliste

Basierend auf den Darstellungen in den Abbildungen 6.3 und 6.4 ergibt sich die Funktionsliste nach Tabelle 6.5.

6.1.4.3. FE 3 - Wiederholung für niedrigere Hierarchieebenen

Anhand dieser Funktionsliste wird nun eine *Funktionale Gefährdungsanalyse* durchgeführt. Danach wird der *Funktionale Entwurf* auf der nächsten Hierarchieebene wiederholt.

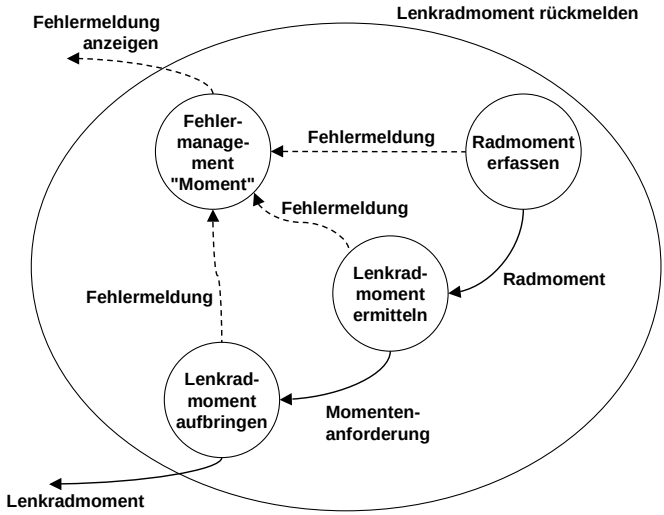

Abbildung 6.4.: Flussdiagramm der Funktion *Lenkradmoment rückmelden* aus Abbildung 6.2 auf Seite 146

Tabelle 6.5.: Funktionsliste in der Hierarchieebene 2

Nr.	Funktion
1.1	Lenkradwinkel erfassen
1.2	Raddrehung ermitteln
1.3	Räder drehen
1.4	Fehlermanagement „Lenken“
2.1	Radmoment erfassen
2.2	Lenkradmoment ermitteln
2.3	Lenkradmoment aufbringen
2.4	Fehlermanagement „Moment“

6.1.5. Funktionale Gefährdungsanalyse - Hierarchieebene 2

Nun wird die *Funktionale Gefährdungsanalyse* für die neu identifizierten Funktionen (*Lenkradwinkel erfassen*, *Raddrehung ermitteln*, *Räder drehen*, *Fehlermanagement „Lenken“*, *Radmoment erfassen*, *Lenkradmoment ermitteln*, *Lenkradmoment aufbringen* und *Fehlermanagement „Moment“*) durchgeführt.

In der Darstellung hier werden aus Gründen der Komplexität und des Umfangs nicht alle diese Funktionen betrachtet, sondern nur exemplarisch die Funktion 1.3 *Räder drehen* und die Funktion 1.4 *Fehlermanagement „Lenken“*.

Das Ergebnis ist in den beiden Tabellen 6.6 auf Seite 156 und 6.7 auf Seite 156 dargestellt.

6.1.5.1. FGA 1 - Identifikation und Beschreibung möglicher Gefährdungen

Für diese beiden Funktionen *Räder drehen* und *Fehlermanagement „Lenken“* werden mögliche Fehler und dazugehörige Gefährdungen ermittelt. Beispiele dafür finden sich in der dritten Spalte in Tabelle 6.6 bzw. Tabelle 6.7.

Mögliche Gefährdungen von *Räder drehen* sind zunächst das fehlerhafte Drehen der Räder, also um einen falschen Betrag und/oder in eine falsche Richtung. Zusätzlich können die Räder in einer blockierten Stellung festgehalten werden, sie behalten also starr ihre Richtung bei. Eine dritte mögliche Gefährdungssituation ist, dass die Räder gar nicht gedreht werden, sie hängen lose in ihrer Aufhängung.

Für *Fehlermanagement „Lenken“* ergeben sich nur zwei mögliche Gefährdungen. Zum einen der Fall, dass trotz Vorliegen eines Fehlers im System dieser nicht angezeigt wird, und der umgekehrte Fall, dass ein Fehler im Kombiinstrument angezeigt wird, obwohl es einen solchen gar nicht gibt.

Dabei werden wieder die beiden Situationen berücksichtigt, dass die Lenkung gerade benutzt wird und dass sie gerade nicht in Verwendung ist. Sie sind entsprechend in Spalte vier der Tabellen 6.6 bzw. 6.7 eingetragen.

6.1.5.2. FGA 2 - Ermittlung der Fehlerauswirkungen

Die ermittelten Fehlerauswirkungen sind in der fünften Spalte von Tabelle 6.6 bzw. Tabelle 6.7 wiedergegeben.

6.1.5.3. FGA 3 - Klassifikation der ermittelten Gefährdungen

Anhand der ermittelten Fehlerauswirkungen werden die Gefährdungen klassifiziert, es ergibt sich so die komplette Darstellung von Tabelle 6.6 und von Tabelle 6.7.

6.1.5.4. FGA 4 - Ableitung von quantitativen bzw. qualitativen Anforderungen

Nach Tabelle 5.2 können den in die jeweiligen Sicherheitsstufen klassifizierten Gefährdungen und damit auch den jeweiligen Funktionen quantitative Zuverlässigkeitsanforderungen zugeordnet werden. Die Voraussetzung dafür ist im Beispielsystem gegeben.

Daraus ergibt sich für die Funktion *Räder drehen* (SF 4) eine maximale Fehlerrate von $\lambda_{Lenken} \leq 10^{-9}/h$. Da die Funktion *Fehlermanagement „Lenken“* in SF 0 eingestuft wurde, ist diese als nicht sicherheitsrelevant anzusehen. Es ergibt sich daher auch keine maximale Fehlerrate.

6.1.5.5. FGA 5 - Spezifikation von Methoden für die Verifikation

Für die Verifikation der Sicherheitsanforderungen an die Funktion *Räder drehen* ist die Erfüllung der in FGA 4 festgelegten Sicherheitsanforderungen nachzuweisen, für *Fehlermanagement „Lenken“* entfällt dieser Nachweis (Einstufung in SF 0). Der Nachweis kann beispielsweise in Form eines Fehlerbaums erfolgen.

6.1.5.6. FGA 6 - Vergleich mit bisherigen Erfahrungen

Entfällt im Beispiel.

6.1.5.7. FGA 7 - Wiederholung für niedrigere Hierarchieebenen

In einem nächsten Schritt wird das System im Rahmen des *Funktionalen Entwurfs* weiter verfeinert, dann erfolgt eine erneute *Funktionale Gefährdungsanalyse* für die nächste Hierarchieebene.

Dabei ist zu berücksichtigen, dass mindestens eine Subfunktion der Funktion *Räder drehen* nach Sicherheitsstufe SF 4 zu klassifizieren ist. Subfunktionen von *Fehlermanagement „Lenken“* müssen bei der Funktionalen Gefährdungsanalyse auf der nächsten Hierarchiestufe nicht weiter betrachtet werden, da *Fehlermanagement „Lenken“* in die unkritische Sicherheitsstufe SF 0 eingestuft wurde.

Tabelle 6.6.: Funktionale Gefährdungsanalyse der Funktion *Räder drehen*

Nr.	Funktion	Gefährdung	Randbedingung	Auswirkungen	Klass.
1.3.A.1	Räder drehen	Rad falsch gedreht	Lenkfunktion wird momentan verwendet	Tod oder schwere Verletzung einer oder mehrerer Personen möglich, Situation ist dabei nicht beherrschbar, Auftreten im Regelfall	SF 4
1.3.A.2			Lenkfunktion wird momentan nicht verwendet	Tod oder schwere Verletzung einer oder mehrerer Personen möglich, Situation ist dabei nicht beherrschbar, Auftreten im Regelfall	SF 4
1.3.B.1		Rad blockiert	Lenkfunktion wird momentan verwendet	Tod oder schwere Verletzung einer oder mehrerer Personen möglich, Situation ist dabei nicht beherrschbar, Auftreten im Regelfall	SF 4
1.3.B.2			Lenkfunktion wird momentan nicht verwendet	Tod oder schwere Verletzung einer oder mehrerer Personen möglich, Situation ist dabei nicht beherrschbar, Auftreten im Einzelfall	SF 3
1.3.C.1		Rad nicht gedreht	Lenkfunktion wird momentan verwendet	Tod oder schwere Verletzung einer oder mehrerer Personen möglich, Situation ist dabei nicht beherrschbar, Auftreten im Regelfall	SF 4
1.3.C.2			Lenkfunktion wird momentan nicht verwendet	Tod oder schwere Verletzung einer oder mehrerer Personen möglich, Situation ist dabei nicht beherrschbar, Auftreten im Einzelfall	SF 3

Tabelle 6.7.: Funktionale Gefährdungsanalyse der Funktion *Fehlermanagement „Lenken“*

Nr.	Funktion	Gefährdung	Randbedingung	Auswirkungen	Klass.
1.4.A.1	Fehlermanagement „Lenken“	keine Fehleranzeige, obwohl ein Fehler vorliegt	Lenkfunktion wird momentan verwendet	keine unmittelbare Gefahr (Mehrfachfehler werden hier nicht betrachtet)	SF 0
1.4.A.2			Lenkfunktion wird momentan nicht verwendet	keine unmittelbare Gefahr (Mehrfachfehler werden hier nicht betrachtet)	SF 0
1.4.B.1		Fehleranzeige, obwohl kein Fehler vorliegt	Lenkfunktion wird momentan verwendet	keine unmittelbare Gefahr	SF 0
1.4.B.2			Lenkfunktion wird momentan nicht verwendet	keine unmittelbare Gefahr	SF 0

6.1.6. Funktionaler Entwurf - Hierarchieebene 3

Aus Gründen der Komplexität werden in diesem dritten Iterationsschritt die Funktionen nicht weiter verfeinert, sondern nur kurz funktional erläutert. Grundsätzlich werden aber auch in diesem Schritt alle Funktionen weiter verfeinert, oder, falls eine weitere Verfeinerung nicht mehr möglich ist, abschließend beispielsweise durch CSpecs oder MiniSpecs beschrieben.

6.1.6.1. FE 1 - Identifikation der Systemfunktionen

Im vorigen Schritt wurden für die Funktionen *Lenken* und *Lenkradmoment rückmelden* die Subfunktionen *Lenkradwinkel erfassen*, *Raddrehung ermitteln*, *Räder drehen* und *Fehlermanagement „Lenken“* bzw. *Radmoment erfassen*, *Radmoment in Lenkmoment wandeln*, *Lenkradmoment aufbringen* und *Fehlermanagement Moment* ermittelt. Diese Funktionen werden nun in diesem Schritt erläutert.

Lenkradwinkel erfassen Der Fahrer gibt durch Drehen am Lenkrad einen Lenkradwinkel vor, mit dem er einen Richtungswunsch zum Ausdruck bringt. Dieser Lenkwinkel wird durch geeignete Sensorikmaßnahmen im Rahmen dieser Funktion erfasst und als Eingabeinformation verarbeitet. Diese Information wird an die nächste Funktion in geeigneter Form übermittelt.

Diese Funktion wird aus Gründen der Komplexität hier in diesem Beispiel nicht weiter verfeinert. Eine detailliertere Darstellung könnte in Form eines weiteren Flussdiagramms und damit einer weiteren hierarchischen Gliederung, in Form einer textuellen Beschreibung wie beispielsweise eine sogenannte „MiniSpec“ oder durch eine Kontrollspezifikation (in SA/RT „CSpec“ genannt) wie z. B. einen Zustandsautomaten oder ein Statechart erfolgen.

Raddrehung ermitteln Auf Basis der Eingangsinformation *Lenkwinkel* wird vom System durch diese Funktion ermittelt, in welche Richtung und um welchen Betrag die Vorderräder gedreht werden sollen. Diese Ausgangsinformation *Radwinkel* wird an *Räder drehen* übergeben.

Auch diese Funktion wird aus Gründen der Komplexität hier in diesem Beispiel nicht weiter verfeinert. Es gilt das bei *Lenkradwinkel erfassen* Gesagte.

Räder drehen Auf Basis der Eingangsinformation *Radwinkel* werden im Rahmen dieser Funktion die Vorderräder um einen entsprechenden Betrag gedreht. Diese Funktion beinhaltet die Darstellung einer entsprechenden Aktorik.

Auch diese Funktion wird aus Gründen der Komplexität hier in diesem Beispiel nicht weiter verfeinert. Es gilt das bei *Lenkradwinkel erfassen* Gesagte.

Fehlermanagement „Lenken“ Diese Funktion sammelt Fehlermeldungen der anderen drei Funktionen und bereitet sie entsprechend zur Weitergabe an die übergeordnete Funktion auf.

Diese Funktion wird ebenfalls aus Gründen der Komplexität hier in diesem Beispiel nicht weiter verfeinert. Es gilt das bei *Lenkradwinkel erfassen* Gesagte.

Radmoment erfassen In dieser Funktion wird in einer geeigneten Art und Weise das an den Vorderrädern anliegende Moment sensiert, für die Weiterverarbeitung aufbereitet und in Form der Information *Radmoment* an die Funktion *Radmoment in Lenkmoment wandeln* übergeben.

Diese Funktion wird aus Gründen der Komplexität hier in diesem Beispiel nicht weiter verfeinert. Es gilt das bei *Lenkradwinkel erfassen* Gesagte.

Radmoment in Lenkmoment wandeln Auf Basis der an den Rädern sensierten Momenteninformation wird durch diese Funktion das einzustellende Lenkradmoment errechnet und an die nächste Funktion weitergegeben.

Auch diese Funktion wird aus Gründen der Komplexität hier in diesem Beispiel nicht weiter verfeinert. Es gilt das bei *Lenkradwinkel erfassen* Gesagte.

Lenkradmoment aufbringen Im Rahmen dieser Funktion wird am Lenkrad haptisch ein Moment aufgebracht, das dem Fahrer eine Rückmeldung über das an den Vorderrädern anliegende Lenkmoment geben soll.

Diese Funktion wird ebenfalls aus Gründen der Komplexität hier in diesem Beispiel nicht weiter verfeinert. Es gilt das bei *Lenkradwinkel erfassen* Gesagte.

Fehlermanagement „Moment“ Diese Funktion sammelt Fehlermeldungen der anderen drei Funktionen und bereitet sie entsprechend zur Weitergabe an die übergeordnete Funktion auf.

Wie die anderen Funktionen auch wird diese Funktion aus Gründen der Komplexität hier in diesem Beispiel nicht weiter verfeinert. Es gilt das bei *Lenkradwinkel erfassen* Gesagte.

6.1.6.2. FE 2 - Erstellen einer Funktionsliste

Dieser Schritt entfällt hier.

6.1.6.3. FE 3 - Wiederholung für niedrigere Hierarchieebenen

Prinzipiell wird der *Funktionale Entwurf* so lange wiederholt, bis alle Funktionen umfassend beschrieben sind. Auf jeder Ebene wird für die neu ermittelten Funktionen eine *Funktionale Gefährdungsanalyse* durchgeführt.

Hier im Beispiel wird die Darstellung an dieser Stelle nicht weiter verfeinert.

6.1.7. Funktionale Gefährdungsanalyse - Hierarchieebene 3

In der Darstellung hier im Beispiel wurden keine weiteren Funktionen auf dieser Hierarchieebene identifiziert, die *Funktionale Gefährdungsanalyse* entfällt daher.

6.1.8. Ergebnis

Wie schon an diesem vereinfachten Beispiel erkennbar ist, wird die Darstellung im Rahmen des *Funktionalen Entwurfs* und der *Funktionalen Gefährdungsanalyse* schon auf niedrigen Hierarchiestufen sehr umfangreich und komplex und ist ohne eine geeignete Werkzeugunterstützung nicht mehr zu bewältigen. Ein geeignetes Werkzeug sollte

- graphische Unterstützung bei der Erfassung der Funktionen bieten. Die funktionale Gliederung ist die Grundlage für die *Funktionale Gefährdungsanalyse* und für den *System-Entwurf*, daher ist hier besondere Sorgfalt vonnöten.
- auf die entsprechende Methode der funktionalen Beschreibung (hier im Beispiel SA/RT) zugeschnitten sein.
- das Durchführen der *Funktionalen Gefährdungsanalyse* unterstützen (z. B. durch entsprechende Eingabemasken).
- die Ergebnisse der *Funktionalen Gefährdungsanalyse* den zugehörigen Funktionen zuordnen, sowie auf Basis der ermittelten Gefährdungen die Generierung von Fehlerbäumen bzw. FMEAs unterstützen.
- die Ergebnisse des *Funktionalen Entwurfs* und der *Funktionalen Gefährdungsanalyse* in geeigneter Form zur Weiterverarbeitung beim weiteren Vorgehen speichern.

Zusätzlich ist beim *Funktionalen Entwurf* die Trennung von Funktion und der Umsetzung der Funktionalität in einem Systemkonzept zu beachten. Auf niedrigeren Hierarchieebenen der Systemdarstellung wird diese Trennung zunehmend schwieriger. Kann das System in einer detaillierteren Form nur noch unter Zuhilfenahme von konkreten Beschreibung einer Umsetzung des Systems erfolgen, so ist der *Funktionale Entwurf* an dieser Stelle zu beenden.

6.2. Beispiel 2: System-Entwurf und Entwurfsbegleitende System-Sicherheitsbewertung

In diesem Abschnitt werden die Ergebnisse des *Funktionalen Entwurfs* und der *Funktionalen Gefährdungsanalyse* aus dem vorherigen Beispiel aufgegriffen. Basierend darauf wird exemplarisch die Vorgehensweise beim *System-Entwurf* und der parallelen *Entwurfsbegleitenden System-Sicherheitsbewertung* dargestellt.

Dazu wird ein mögliches beispielhaftes Systemkonzept verwendet und hinsichtlich der Erfüllung der Sicherheitsanforderungen bewertet. Am Beispiel des Bordnetzes wird dann die Darstellung des Systems verfeinert. Zu diesem Zweck wird eine konkrete Gefährdung herausgegriffen und die Bordnetzarchitektur daraufhin analysiert, ob die Sicherheitsanforderungen erfüllt werden.

Dabei ist zu beachten, dass die verwendeten Daten und Inhalte dieses Kapitels als rein hypothetisch anzusehen sind und nur der Veranschaulichung der Vorgehensweise dienen. Rückschlüsse auf reale Systemarchitekturen, konkrete quantitative Anforderungen oder mögliche Lösungsansätze im Automobilbereich sind daher nicht möglich.

6.2.1. System-Entwurf

Als erster Schritt in diesem Beispiel wird ein Systemkonzept für das in Kapitel 6.1 beschriebene System erstellt.

6.2.1.1. SE 1 - Erstellen und Auswahl eines Systemkonzepts

In mehreren Iterationsschritten im Rahmen des *Funktionalen Entwurfs* wurde im vorigen Beispiel im Kapitel 6.1 ein Steer-by-Wire System funktional beschrieben. Diese Funktionale Beschreibung wird nun, gepaart mit zusätzlichen nichtfunktionalen Randbedingungen, die im Rahmen einer *System-Anforderungsanalyse*

festgehalten wurden, auf eine mögliche Systemarchitektur abgebildet. Es entsteht so ein Systemkonzept.

Eine mögliche, beispielhafte Systemarchitektur für ein Steer-by-Wire System zeigt Abbildung 6.5 nach [Nel02]. Alternative, ähnliche Darstellungen finden sich auch in [Bin01] oder [Mah00].

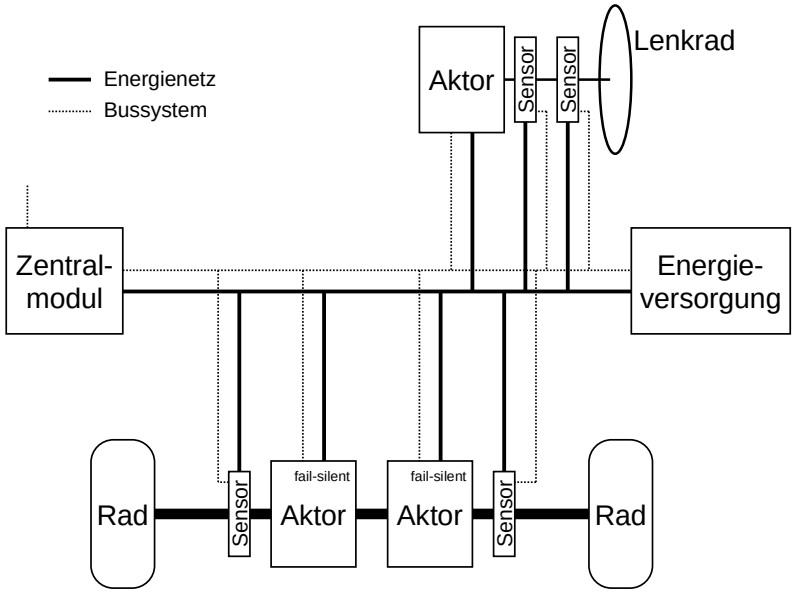

Abbildung 6.5.: Systemkonzept für ein Steer-by-Wire System

Das Systemkonzept sieht eine redundante, fail-silent ausgeführte Aktorik für die Drehung der Räder vor, die Aktorik für die Rückmeldung des Lenkmoments am Lenkrad ist einfach ausgeführt. Zwei redundante Sensoren nehmen den Lenkwinkel auf, die Sensorik für die Stellung der Räder und für die Erfassung des Lenkmoments ist ebenfalls zweifach redundant. Ein Zentralmodul übernimmt übergreifende und koordinierende Aufgaben und stellt eine Schnittstelle zu anderen Anwendungen im Fahrzeug dar. Alle Systemkomponenten sind durch ein hier nicht näher definiertes Bussystem verbunden, zusätzlich wird jede Komponente mit Energie versorgt.

6.2.2. Entwurfsbegleitende System-Sicherheitsbewertung

Die Bewertung des Systemkonzepts erfolgt in der *Entwurfsbegleitenden System-Sicherheitsbewertung*.

6.2.2.1. ESSB 1 - Überprüfung der Anforderungen auf Vollständigkeit

Entfällt im Beispiel.

6.2.2.2. ESSB 2 - Überprüfung des Systems

Dieses Systemkonzept wird nun im Folgenden hinsichtlich der Erfüllung der im Rahmen der *Funktionalen Gefährdungsanalyse* gestellten Sicherheitsanforderungen bewertet. Dabei liegt der Schwerpunkt auf der Betrachtung der Funktion „Räder drehen“ (vgl. Abschnitt 6.1.6.1) unter besonderer Berücksichtigung der Energieversorgung.

Dazu müssen zunächst die quantitativen Anforderungen an die Energieversorgung abgeleitet werden. Zu diesem Zweck wird eine Fehlerbaumanalyse erstellt, das Top-Ereignis sei die in Tabelle 6.6 der Funktion „Räder drehen“ zugeordnete Gefährdung „Rad nicht gedreht“. In Tabelle 6.6 wurde für diese Gefährdung Sicherheitsstufe SF 4 ermittelt, was einer quantitativen Anforderung im Sinne einer maximalen Fehlerrate für diese Gefährdung von $10^{-9}/h$ entspricht. Bei einer Betriebsdauer von 10000 Stunden entspricht dies einer maximalen Auftretenswahrscheinlichkeit für diese Gefährdung von

$$P_{max,\ Gefaehrdung} = 10^{-5}. \tag{6.1}$$

Diese Gefährdung „Rad nicht gedreht“ ist das Top-Ereignis eines Fehlerbaums, mit dem mögliche Ursachen für diese Gefährdung analysiert werden. Es ergibt sich der (vereinfachte und daher unvollständige) Fehlerbaum von Abbildung 6.6. Dabei kann die Gefährdung „Rad nicht gedreht“ durch einen „Defekt in der Aktorik“, einen „mechanischen Defekt in der Lenkung“, einen „Fehler in der Signalübertragung“ oder durch eine „nicht ausreichende Versorgungsspannung“ für das System ausgelöst werden.

Zur Analyse des Systems müssen nun die Auftretenswahrscheinlichkeiten für diese Fehler ermittelt bzw. abgeschätzt werden. Dazu sind weitere (Unter-)Fehlerbäume oder andere Analysemethoden wie z. B. die Markov-Analyse notwendig, was bei kompletter Durchführung den Rahmen dieses Beispiels sprengen würde.

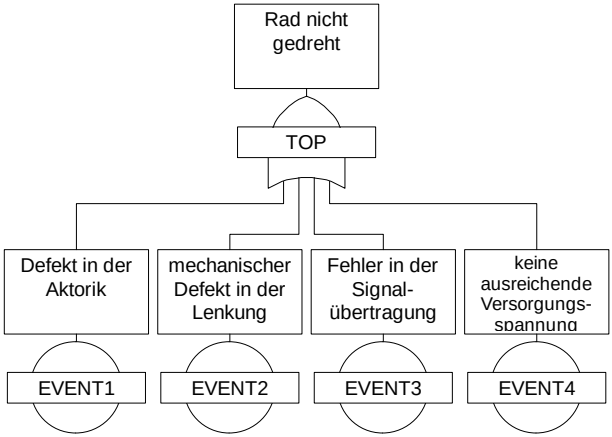

Abbildung 6.6.: Fehlerbaum für die Gefährdung „Rad nicht gedreht“

6.2.2.3. ESSB 3 - Analyse gemeinsamer Fehlerursachen

Entfällt hier.

6.2.2.4. ESSB 4 - Ableitung von Sicherheitsanforderungen

Für die weitere Betrachtung in diesem Beispiel soll angenommen werden, dass sich aufgrund von entsprechenden Analysen des Systemkonzepts eine Anforderung für die maximale Auftretenswahrscheinlichkeit des Fehlers „keine ausreichende Versorgungsspannung“ von

$$P_{Fehler,\ Bordnetz,\ max} = 1\ ppm \tag{6.2}$$

ergibt. Diese Auftretenswahrscheinlichkeit bezieht sich auf die Lebensdauer des Systems (15 Jahre bzw. 10000 Stunden).

Mit dieser Anforderung für $P_{Fehler,\ Bordnetz,\ max}$ können nun verschiedene Konzepte für das Energie-Bordnetz bewertet werden.

6.2.3. System-Entwurf

6.2.3.1. SE 2 - Entwurf des Systems

In einem nächsten Schritt wird nun eine mögliche System-Architektur für das Bordnetz erstellt. Abbildung 6.7 nach [Bau99] zeigt eine einfache, grundlegende Bordnetz-Architektur, bestehend aus einem Generator G, einem Starter St, einer Batterie, einem Kabelbaum inkl. Sicherungen (in der Abbildung durch eine Sicherung repräsentiert) und einem Verbraucher.

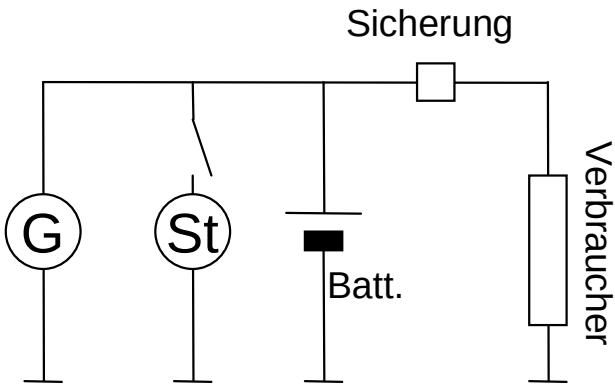

Abbildung 6.7.: Systemarchitektur für das Energiebordnetz

Der Starter wird lediglich dazu verwendet, den Motor des Fahrzeugs zu starten. Er wird dazu von der Batterie mit Energie versorgt, danach wird der dargestellte Schalter geöffnet und der Starter vom System abgeklemmt. Da die Startphase des Fahrzeuges hier nicht betrachtet werden soll, muss der Starter für die weitere Betrachtung der Sicherheitseigenschaften des Systems nicht berücksichtigt werden. Grundsätzlich ist er aber, da er ein Systembestandteil ist, bei Sicherheitsanalysen mittels FMEA einzubeziehen.

Das System besteht somit aus den Energiequellen Generator und Batterie (der Generator lädt im Normalbetrieb die Batterie), dem Kabelbaum (inkl. Leitungen, Stecker, Sicherungen etc.) und einem Verbraucher (dies sei hier im Beispiel das betrachtete Steer-by-Wire System). Weitere Verbraucher könnten in Parallelschaltung zu diesem Verbraucher dargestellt werden.

6.2.4. Entwurfsbegleitende System-Sicherheitsbewertung

6.2.4.1. ESSB 2 - Überprüfung des Systems

Die Systemarchitektur aus Abbildung 6.7 wird nun hinsichtlich der Erfüllung der Sicherheitsanforderungen bewertet. Dazu wird für das dargestellte System ein Fehlerbaum erstellt. Das Top-Ereignis sei „Spannung am Verbraucher zu niedrig“.

Für die Erstellung des Fehlerbaumes wird angenommen, dass entweder der Wegfall der Energieerzeugung (Batterie oder Generator) oder der Wegfall der Energieverteilung (Kabelbaum) zum Top-Ereignis „Spannung am Verbraucher zu niedrig“ führen können. Dabei wird davon ausgegangen, dass bei einem Ausfall der Batterie der Generator noch für eine begrenzte, aber ausreichend lange Zeit genügend Energie für den Betrieb des Verbrauchers aufbringen kann. Dies gilt auch umgekehrt. Allerdings ist ein Starten des Fahrzeugs in diesem Fehlerfall nicht mehr möglich. Bei der Analyse hier wird allerdings nur das Betriebsverhalten des Systems betrachtet.

Zur quantitativen Analyse sind die Ausfalldaten der jeweiligen Komponenten notwendig. Bei der Beschaffung entsprechender Daten zeigt sich die schon auf Seite 25 beschriebene Problematik, dass gute Zahlen nur schwer zu ermitteln sind. Um hier im Beispiel trotzdem mit möglichst realistischen Daten rechnen zu können, wird auf die ADAC-Pannenstatistik (vgl. [ADA01]) zurückgegriffen. Die dort verfügbaren Ausfalldaten beziehen sich auf liegengebliebene Fahrzeuge im Jahre 2000, deren Ausfallursache für Fahrzeuge unterschiedlicher Baujahre nach Bordnetzkomponenten aufgeschlüsselt ist. Durch Mittelung und Rundung der Ausfalldaten von bis zu 3 Jahre alten Fahrzeugen zu etwas ungünstigeren Werten hin wurden die mittleren Ausfallwahrscheinlichkeiten in Tabelle 6.8 ermittelt. Abbildung 6.8 zeigt den unter Verwendung dieser Zahlen entstandenen Fehlerbaum.

Tabelle 6.8.: Ausfallursachen und -wahrscheinlichkeiten für das Energiebordnetz im Kfz; Bezugszeitraum: 1 Jahr

Ausfallursache	Ausfallwahrscheinlichkeit
Batterie	2000 ppm
Generator	370 ppm
Kabelbaum	70 ppm

Nach Daten des ADAC (vgl. [ADA01])

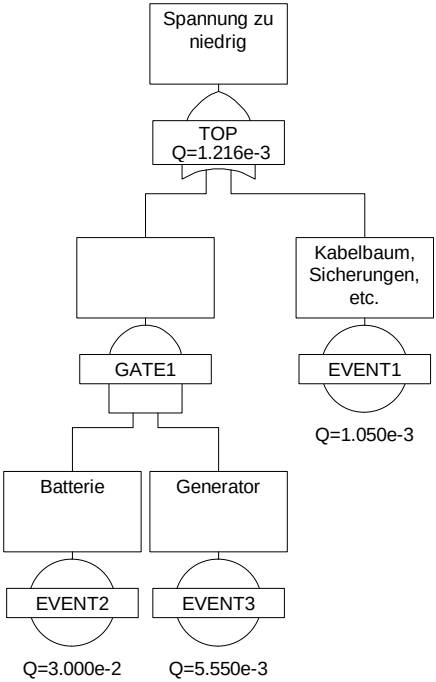

Abbildung 6.8.: Fehlerbaum der Bordnetzarchitektur aus Abbildung 6.7

Für die weitere Betrachtung sei nun angenommen, dass die Ausfälle der Komponenten expontentialverteilt seien. Dies kann in Realität nur für elektrische oder elektronische Systeme angenommen werden, ein chemisches System wie die Batterie wäre für eine reale Analyse sicherlich auf eine andere Art und Weise zu modellieren. Aber zum Aufzeigen des Grundprinzips beim Vorgehen bei der Sicherheitsanalyse soll diese Annahme der Exponentialverteilung hier gelten.

Dann gilt mit Formel 2.10 und der Näherung nach Formel 2.16 für die Funktion der Ausfallwahrscheinlichkeit

$$F(t) \approx \lambda \cdot t. \tag{6.3}$$

Bei einer angenommen Betriebsdauer von 667h pro Jahr (dies entspricht 10000 Betriebsstunden in 15 Jahren) ergeben sich die Ausfallraten λ von Tabelle 6.9.

Tabelle 6.9.: Ausfallraten von Bordnetzkomponenten

Komponente	Ausfallrate
Batterie	3,00 ppm/h
Generator	0,56 ppm/h
Kabelbaum	0,11 ppm/h

Für die Betrachtung von Ausfallwahrscheinlichkeiten innerhalb der Lebensdauer eines Automobils werden 10000 Betriebsstunden zugrunde gelegt. Es ergeben sich unter Anwendung von Formel 6.3 die Werte von Tabelle 6.10.

Tabelle 6.10.: Ausfallwahrscheinlichkeiten von Bordnetzkomponenten; Bezugszeitraum Fahrzeugslebensdauer

Komponente	Ausfallwahrscheinlichkeit q_i
Batterie	30000 ppm
Generator	5550 ppm
Kabelbaum	1050 ppm

Die Fehlerbaumanalyse (vgl. Abbildung 6.8) ergibt für die Funktion der Ausfallwahrscheinlichkeit

$$F_{Bordnetz} = q_{Batt} \cdot q_{Gen} + q_{Kabel} - q_{Batt} \cdot q_{Gen} \cdot q_{Kabel}. \tag{6.4}$$

Es ergibt sich mit den Zahlen aus Tabelle 6.10 eine Auftretenswahrscheinlichkeit für das Top-Ereignis „Spannung zu niedrig“ von

$$P_{Spannung\ zu\ niedrig} = 1216\ ppm. \tag{6.5}$$

Mit diesem Wert wird die Anforderung von $P_{Fehler,\ Bordnetz,\ max} = 1\ ppm$ deutlich verfehlt, Verbesserungen im System oder der Systemkomponenten sind notwendig. Daher werden zur Analyse des Systems die strukturelle Importanz, die marginale Importanz und die fraktionale Importanz der Komponenten dieses Systems als Kenngrößen errechnet.

Die strukturelle Importanz $I_{\bar{\Phi}}(i)$ ist das Maß für die Wichtigkeit einer Komponente i aufgrund ihrer logischen Anordnung ($\bar{\Phi}$ ist die Ausfallfunktion des betrachteten Systems). Die marginale Importanz $I_m(i)$ berücksichtigt neben dem struktu-

rellen auch den probabilistischen Einfluss, den eine Komponente i auf die System-Ausfallwahrscheinlichkeit ausübt. Die fraktionale Importanz $I_f(i)$ identifiziert diejenige Komponente, deren relative Änderung der Ausfallwahrscheinlichkeit den größten Einfluss auf das System hat (vgl. [MP03]).

Nach [MP03] errechnet sich die marginale Importanz $I_m(i)$ einer Komponente i zu

$$I_m(i) = \frac{\partial F}{\partial q_i}, \tag{6.6}$$

mit F als der Funktion der Ausfallwahrscheinlichkeit des System und q_i als der Ausfallwahrscheinlichkeit der betrachteten Komponente i.

Es ergeben sich folgende Formeln für die marginalen Importanzen:

$$I_m(Batt) = \frac{\partial F}{\partial q_{Batt}} = q_{Gen} \cdot (1 - q_{Kabel}) \tag{6.7}$$

$$I_m(Gen) = \frac{\partial F}{\partial q_{Gen}} = q_{Batt} \cdot (1 - q_{Kabel}) \tag{6.8}$$

$$I_m(Kabel) = \frac{\partial F}{\partial q_{Kabel}} = (1 - q_{Batt} \cdot q_{Gen}) \tag{6.9}$$

Die strukturelle Importanz $I_{\bar{\Phi}}(i)$ ergibt sich gemäß [MP03] aus der marginalen Importanz nach Formel 6.6 mit $q = \frac{1}{2}$, also

$$I_{\bar{\Phi}}(i) = I_m(i) \mid_{q=\frac{1}{2}} . \tag{6.10}$$

Nach [MP03] ergibt sich die fraktionale Importanz $I_f(i)$ aus der marginalen Importanz nach Formel 6.6 zu

$$I_f(i) = I_m(i) \cdot q_i. \tag{6.11}$$

Setzt man die Werte aus Tabelle 6.10 für die Ausfallwahrscheinlichkeiten q_{Batt}, q_{Gen} und q_{Kabel} in Formel 6.7 bis 6.11 ein, so ergeben sich die Werte von Tabelle 6.11.

Aus dieser Analyse wird ersichtlich, dass bei dieser Systemarchitektur für die verwendete Fehlerhypothese das Ausfallverhalten des Bordnetzes von Fehlern des Kabelbaumes dominiert wird. Es ist daher dringend anzuraten, zur Verbesserung

Tabelle 6.11.: Importanzen der Komponenten des Systems aus Abbildung 6.7

Komponente i	strukturelle Importanz $I_{\bar{\Phi}}(i)$	marginale Importanz $I_m(i)$	fraktionale Importanz $I_f(i)$
Batterie	0,25	0,005544	0,000166
Generator	0,25	0,029969	0,000166
Kabelbaum	0,75	0,999969	0,001050

des Fehlerverhaltens Änderungen am Kabelbaum vorzunehmen, welche die Zuverlässigkeit dieser Systemkomponente verbessern. Dies können physikalische Verbesserungen, bessere Stecker und Sicherungen, aber auch Redundanzkonzepte mit entsprechender Überwachung sein.

Für die weitere Analyse wird nun angenommen, dass aufgrund entsprechender Maßnahmen die Ausfallwahrscheinlichkeit des Kabelbaums von 1050 ppm auf 10 ppm verbessert werden kann. Eine erneute Fehlerbaumanalyse mit diesem Wert für die Fehlerwahrscheinlichkeit des Kabelbaums ergibt eine Auftretenswahrscheinlichkeit für das Top-Ereignis „Spannung zu niedrig“ von

$$P_{Spannung\ zu\ niedrig,\ 2} = 177\ ppm, \tag{6.12}$$

was einer weiteren Verbesserung gegenüber dem Wert von $P_{Spannung\ zu\ niedrig}$ um eine Größenordnung entspricht, aber im Vergleich mit $P_{Fehler,\ Bordnetz,\ max}$ immer noch deutlich zu schlecht ist.

Eine kritische Komponente mit sehr schlechten Zuverlässigkeitseigenschaften ist die Batterie. Im Allgemeinen geht man von einer mittleren Lebensdauer für eine Autobatterie von lediglich 3 bis 4 Jahren aus (ein entsprechender Austausch ist bei der Ausfallrate von 2000 ppm/a bereits berücksichtigt, da in der ADAC-Statistik Neufahrzeuge betrachtet wurden).

Im Folgenden soll das Konzept aus Abbildung 6.7 hinsichtlich einer Verbesserung des Ausfallverhaltens der Batterie optimiert werden.

6.2.5. System-Entwurf - Iteration

6.2.5.1. SE 2 - Entwurf des Systems

Im vorigen Schritt wurde, nach einer notwendigen Optimierung des Kabelbaums, die Batterie als kritische Komponente identifiziert. In einem überarbeiteten System-

konzept wird die einfach vorhandene Batterie nun durch ein Batteriemanagement-System (BMS) ersetzt. Die neue Systemarchitektur zeigt Abbildung 6.9.

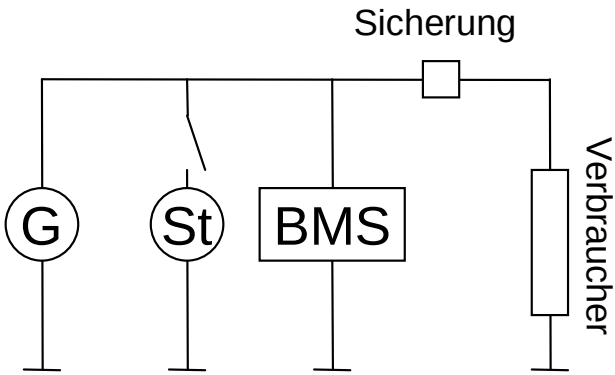

Abbildung 6.9.: Verbesserte Systemarchitektur für das Energiebordnetz

Das Batteriemanagement-System besteht aus zwei identischen Batterien, die über ein Steuergerät SG verschaltet sind (vgl. Abbildung 6.10). Die Funktionalität der Batterie ist somit redundant verfügbar. Das Steuergerät erkennt einen (drohenden) Ausfall einer der beiden Batterien, klemmt diese dann ab und meldet diesen Fehler dem Fahrer, der mit der zweiten Batterie im Notbetrieb weiterfahren kann. Der Fahrer ist in diesem Fall gehalten, die defekte Batterie innerhalb einer Woche (dies entspricht ca. 10 Betriebsstunden) gegen eine neue Batterie zu tauschen.

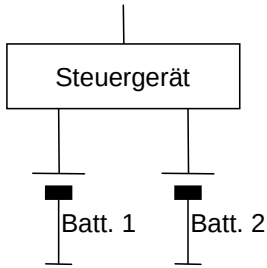

Abbildung 6.10.: Das Batteriemanagementsystem

6.2.6. Entwurfsbegleitende System-Sicherheitsbewertung - Iteration

6.2.6.1. ESSB 2 - Überprüfung des Systems

Das überarbeitete Systemkonzept aus Abbildung 6.9 wird nun wiederum einer Sicherheitsanalyse unterzogen. Dazu wird der Fehlerbaum aus Abbildung 6.8 verwendet, wobei die Komponente „Batterie“ durch das Batteriemanagement-System ersetzt werden muss. Es ergibt sich der modifizierte Fehlerbaum von Abbildung 6.11.

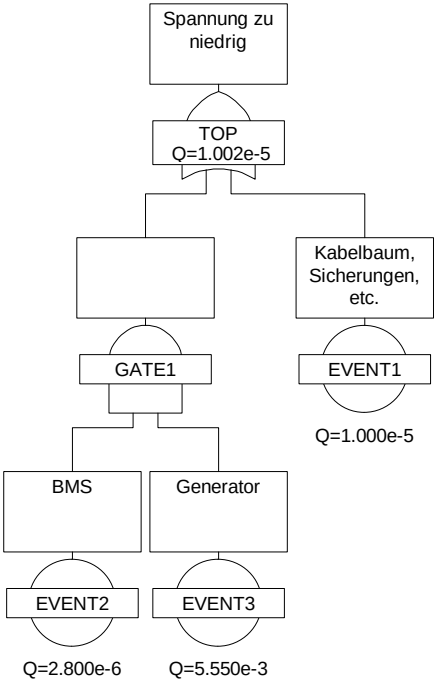

Abbildung 6.11.: Fehlerbaum der Bordnetzarchitektur von Abbildung 6.9

Für die Ermittlung der Auftretenswahrscheinlichkeit eines Fehlers im Batteriemanagement-System muss dieses im Detail betrachtet werden. Dazu sei

angenommen, dass das Batteriemanagement-System BMS ausfällt, wenn das Steuergerät SG oder beide Batterie ausfallen:

$$Ausfall_{BMS} = Ausfall_{SG} \vee Ausfall_{beide\ Batterien} \tag{6.13}$$

Aus Formel 6.13 ergibt sich für die Berechnung der Ausfallwahrscheinlichkeit des Batteriemanagement-Systems

$$P_{BMS} = P_{SG} + P_{b.\ Batt.} - P_{SG} \cdot P_{b.\ Batt.}. \tag{6.14}$$

Eine typische Ausfallwahrscheinlichkeit für das Steuergerät ist $P_{SG} = 1\ ppm$.

Im Gegensatz zum ursprünglichen Systemkonzept von Abbildung 6.7 ist im jetzigen Konzept eine Reparatur des Batteriemanagement-Systems während des Betriebs des Systems möglich und auch gefordert. Das Ausfallverhalten von Systemen mit Reparatur kann nicht mit einer Fehlerbaumanalyse modelliert werden, daher kommt hier die Markov-Analyse zum Einsatz.

Das Markov-Modell für die Betrachtung des Ausfallverhaltens der beiden Batterien des Batteriemanagement-System BMS zeigt Abbildung 6.12. Die Zustände 2 und 3 lassen sich aus Symmetriegründen in einen Zustand zusammenfassen, aus Gründen der Anschaulichkeit soll aber ein Markov-Modell mit vier Zuständen verwendet werden.

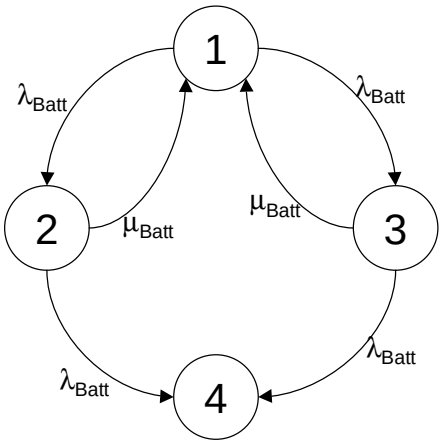

Abbildung 6.12.: Markov-Modell des Ausfallverhaltens der beiden Batterien

Zustand 1 ist der Normalzustand, beide Batterien sind funktionstüchtig. In Zustand 2 ist Batterie 1 ausgefallen, in Zustand 3 Batterie 2. Zustand 4 wird erreicht, wenn beide Batterien ausgefallen sind, d. h. die Reparatur des Systems durch Ersetzen der fehlerhaften Batterie durch eine neue ist nicht rechtzeitig erfolgt. Die Ausfallrate λ_{Batt} der Batterien ergibt sich aus den Daten aus Tabelle 6.9 zu

$$\lambda_{Batt} = 3\ ppm/h. \tag{6.15}$$

Für die Reparaturrate μ_{Batt} gilt bei einer Zeitdauer von 10 Betriebsstunden bis zu einem Austausch und unter Annahme einer Exponentialverteilung

$$\mu_{Batt} = \frac{1}{MTTR} = 10^{-1}/h. \tag{6.16}$$

Bei einer Betriebsdauer des Systems von 10000 Stunden ergibt die Markov-Analyse unter Zuhilfenahme des Software-Werkzeugs CARMS (vgl. [MS04]; eine analytische Herleitung der zugrundeliegenden Gleichungen findet sich in Kapitel 2.4.3, weitergehende Informationen in [Gör97]) eine Wahrscheinlichkeit für den Zustand 4 (also für einen Systemausfall) von

$$P_4 = P_{b.\ Batt.} \approx 1{,}8\ ppm. \tag{6.17}$$

Damit ergibt sich nach Formel 6.14 die Ausfallwahrscheinlichkeit des Batteriemanagement-Systems zu

$$P_{BMS} = 1\ ppm + 1{,}8\ ppm - 1\ ppm \cdot 1{,}8\ ppm \approx 2{,}8\ ppm. \tag{6.18}$$

Durch Einsetzen von P_{BMS} in den Fehlerbaum nach Abbildung 6.11 ergibt sich so eine neue Ausfallwahrscheinlichkeit von

$$P_{Spannung\ zu\ niedrig,\ 3} = 10{,}02\ ppm \tag{6.19}$$

für das gesamte Bordnetz nach Abbildung 6.9.

Durch Einführen des BMS konnte eine weitere Verbesserung des Ausfallverhaltens des Gesamtsystems von einer Größenordnung gegenüber dem ursprünglichen Wert erzielt werden. Das Ziel von $P_{Fehler,\ Bordnetz,\ max} = 1\ ppm$ wird aber immer noch um eine Größenordnung verfehlt. Weitere Systemverbesserungen sind daher notwendig.

6.2.6.2. ESSB 3 - Analyse gemeinsamer Fehlerursachen

Für das verwendete Systemkonzept ist eine Common Mode Analyse notwendig. Aufgrund des verwendeten Fehlerbaums des Bordnetzes muss die Unabhängigkeit von Fehlern des Generators und von Fehlern der Batterie bzw. des BMS gezeigt werden. Außerdem muss ein entsprechender Nachweis für das Steuergerät des BMS und die Batterien im BMS erfolgen. Dies kann beispielsweise durch eine vom Ausfall der Batterien unabhängige Spannungsversorgung des Steuergerätes geschehen, z. B. in Form einer eigenständigen Pufferbatterie für das Steuergerät.

Ein ausführlicher Nachweis im Sinne einer Common Mode Analyse würde aber den Rahmen dieses Beispiels sprengen.

6.2.7. System-Entwurf

Die Schritte „SE 3 - Spezifikation von System-Integration und -Verifikation“, „SE 4 - Ableitung von Software- und Hardware-Anforderungen“, „SE 5 - Software-Entwurf“ und „SE 6 - Hardware-Entwurf“ entfallen hier im Beispiel.

6.2.8. Ergebnis

In diesem Abschnitt wurde an einem einfachen Beispiel das Zusammenspiel zwischen *System-Entwurf* und *Entwurfsbegleitender System-Sicherheitsbewertung* aufgezeigt. Ein besonderes Augenmerk galt dabei der Verfügbarkeit des Bordnetzes.

Auch bei dieser vereinfachten Analyse konnte aufgezeigt werden, dass eine herkömmliche Bordnetzarchitektur aufgrund ihrer deutlich zu schlechten Fehlerwahrscheinlichkeit so nicht für sicherheitsrelevante Anwendungen im Automobil zum Einsatz kommen kann. Entsprechende Verbesserungen durch verbesserte Komponenten (wie im Beispiel der Kabelbaum) oder durch neue Systemkonzepte (wie im Beispiel das BMS) sind notwendig. Diese Einschätzung des Bordnetzes findet sich auch in der Literatur. Bei der Entwicklung von prototypischen X-by-Wire Systemen wurde das Hauptaugenmerk bisher auf die Demonstration der Funktion, der verteilten Regelung, der Kommunikation oder der Sicherheitseigenschaften gelegt. Serienprojekte wurden nicht zuletzt aufgrund der schlechten Zuverlässigkeitseigenschaften des Energie-Bordnetzes im Automobil verschoben (vgl. z. B. [Nel02]).

Die Verknüpfung der Fehlerbaumanalyse und der Markov-Analyse bei der *Entwurfsbegleitenden System-Sicherheitsbewertung* stellt ein weiteres wertvolles Werkzeug für die quantitative Analyse von elektronischen Systemen dar. Dabei

sind aber einige Einschränkungen zu beachten. So kann beispielsweise nicht immer eine Exponentialverteilung der Fehler von Komponenten angenommen werden, zudem ist auch die Verfügbarkeit von verlässlichen Zahlenwerten für die quantitative Analyse nicht immer gegeben. Der Aufbau entsprechender Datenbanken auf Basis öffentlich verfügbarer Datenbasen (vgl. Tabelle 2.4) ist daher dringend angeraten.

Ein wichtiger Schritt bei der Sicherheitsanalyse ist die Common Mode Analyse, um getroffene Annahmen zur Unabhängigkeit von Fehlern nachzuweisen.

Zu beachten ist gerade beim Einsatz redundanter Systemkonzepte, dass im Allgemeinen die Ausfallwahrscheinlichkeit des redundanten Systems sorgfältig neu berechnet werden muss und nicht einfach direkt aus den Systemkomponenten abgeleitet werden kann. Zudem ist auch bei exponentialverteilten Komponenten eine Exponentialverteilung des redundanten Gesamtsystems nicht zwingend gegeben, dies hängt stark vom Redundanzkonzept selbst ab (vgl. [Mül04]) . Zur Analyse redundanter Systeme ist daher eine Markov-Analyse angeraten.

Eine FMEA bzw. FMECA wurde hier in diesem Beispiel nicht näher betrachtet, da sie im Automobilbereich mittlerweile weitgehend eingeführt ist.

7. Zusammenfassung und Ausblick

Im Automobilbereich steigt der Anteil der Elektronik an der Funktionalität und damit auch an der Wertschöpfung eines Automobils stetig an. Dabei macht dieser Trend mittlerweile nicht mehr vor Funktionen halt, die die Sicherheit des Fahrzeug erhöhen, zugleich werden dadurch aber auch neue Gefährdungspotenziale eröffnet. Das Potenzial von passiven Sicherheitssystemen ist dabei weitgehend ausgeschöpft, zukünftige Systeme sind überwiegend im Bereich der aktiven Sicherheit angesiedelt. Solche Systeme stellen aber auch einen merklichen Eingriff in die Sicherheit des Fahrzeugs dar und können kritische Gefährdungen bewirken.

Obwohl im Automobilbereich schon seit etlichen Jahren aktive Sicherheitssysteme wie beispielsweise das Antiblockiersystem ABS im Einsatz sind, existiert bislang keine einheitliche und durchgängige Vorgehensweise zur Betrachtung der Systemsicherheit bei der Entwicklung. Für die Einführung zukünftiger Systeme, die sich anders als das ABS im Fehlerfall nicht ohne schwerwiegende Folgen für die Sicherheit des Fahrzeugs abschalten lassen, wird eine solche aber als unbedingt notwendig erachtet.

In anderen Industriebereichen wie der Luftfahrt oder der Prozessautomatisierung ist mehr Erfahrung mit der Behandlung solcher Sicherheitssysteme vorhanden, dort existieren auch eingeführte Sicherheitsprozesse. In der vorliegenden Arbeit wurde ein verbreiteter Sicherheitsprozess in der Luftfahrtindustrie, das Vorgehen nach SAE ARP 4754 und SAE ARP 4761, herausgegriffen und, an die Randbedingungen in der Automobilbranche angepasst, mit dem V-Modell ’97, dem abstrakten Prozessmodell für die Entwicklung von elektronischen und softwareintensiven Systemen im Automobilbereich, verknüpft.

Die entstandene Entwicklungsmethodik besteht aus den neun einzelnen, miteinander verketteten Schritten *System-Anforderungsanalyse*, *Funktionaler Entwurf*, *Funktionale Gefährdungsanalyse*, *System-Entwurf*, *Entwurfsbegleitende System-Sicherheitsbewertung*, *Implementierung*, *Integration und Test*, *System-Sicherheitsbewertung* und *Zulassung und Inbetriebnahme*. Die prinzipielle Vorgehensweise wurde an einem Steer-by-Wire Beispielsystem demonstriert.

Durch diese Arbeit wurde ein Rahmen aufgezeigt, der durch weitere Forschungsaktivitäten ausgefüllt werden muss. Damit die in dieser Arbeit vorgestellte

Methodik auch in der Realität verwendet werden kann und nicht nur für konzeptionelle Arbeiten in der Forschung, sind weitergehende Aktivitäten notwendig. So ist für eine konkrete Anwendung die Methodik an die existierende Vorgehensweise des jeweiligen Unternehmens bzw. Unternehmensbereiches anzupassen.

Zudem ist eine entsprechende Werkzeugunterstützung notwendig. Schon die Darstellung im Beispiel hat gezeigt, dass die Komplexität der Betrachtung schnell sehr umfangreich wird. Daher ist die Methodik wohl ohne entsprechende Softwarewerkzeuge nicht beherrschbar. Gerade die neu hinzugekommenen Schritte *Funktionale Gefährdungsanalyse*, *Entwurfsbegleitende System-Sicherheitsbewertung* und *System-Sicherheitsbewertung* sind in entsprechende Werkzeuge abzubilden.

Konkret ist für das iterative Vorgehen in den Schritten *Funktionaler Entwurf* und *Funktionale Gefährdungsanalyse* eine Werkzeugunterstützung notwendig. Dabei sollte für den *Funktionalen Entwurf* die auf den Automobilbereich zugeschnittene funktionale Beschreibungssprache CARTRONIC (vgl. Kapitel 3.7.4) zum Einsatz kommen. Ein Werkzeug für die *Funktionale Gefährdungsanalyse* muss dafür entsprechende Schnittstellen bieten.

Die Interaktion zwischen dem Schritt *System-Entwurf* und der *Entwurfsbegleitenden System-Sicherheitsbewertung* sollte in ein weiteres Werkzeug abgebildet werden. Für den *System-Entwurf* ist im Automobilbereich das Software-Werkzeug ASCET-SD der Firma ETAS GmbH (vgl. [ETA04]) weit verbreitet, eine entsprechende Anbindung an ASCET-SD ist daher notwendig.

Abbildung 7.1 auf der nächsten Seite als Ausschnitt von Abbildung 4.2 auf Seite 77 zeigt die notwendige Werkzeugunterstützung.

Darüber hinaus ist die zukünftige Entwicklung der Vorschriften für die Typzulassung sicherheitsrelevanter Elektroniksysteme im Automobilbereich noch ungewiss. Trockene X-by-Systeme, also X-by-Wire Systeme ohne eine Rückfallebene in konventioneller Technologie, können nach dem heutigen Stand der Vorschriften nicht zugelassen werden, für die Zukunft ist daher eine Änderung der Vorschriften zu erwarten. Für den nach UN ECE R 13 und UN ECE R 79 bereits geforderten Sicherheitsnachweis (vgl. Kapitel 5.9.1) stellt die vorgestellte Entwicklungsmethodik mit der *System-Sicherheitsbewertung* eine unterstützende Methode zur Verfügung.

Im Zusammenhang mit der Norm DIN EN 61508 ist eine Diskussion um einen Entwicklungsstandard und um einen Standard für den Sicherheitsprozess im Automobilbereich entfacht. Dabei ist die DIN EN 61508 - auch wenn sie eine generische Norm ist - aufgrund ihrer Historie aus dem chemisch-prozesstechnischen Bereich und wegen ihrer expliziten Ausrichtung auf sicherheitsgerichtete Systeme so nicht für Automobilsysteme geeignet. Sie bietet bei der Übertragung auf andere Industriebereiche sehr viel Interpretationsspielraum. Allerdings hat der seit

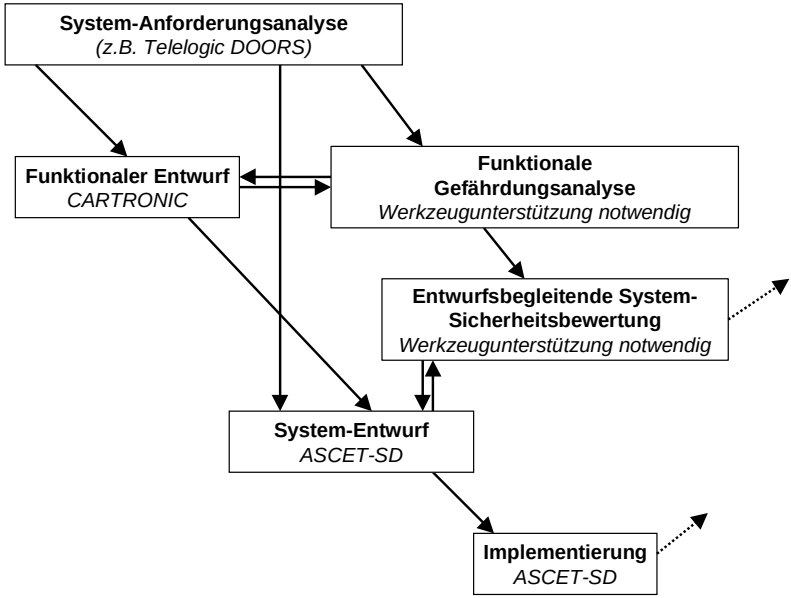

Abbildung 7.1.: Notwendige Werkzeugunterstützung

Sommer 2003 normative Status von DIN EN 61508 auch Einfluss auf den Automobilbereich. Daher arbeitet seit Dezember 2003 der Verband der deutschen Automobilindustrie VDA an einer automobilspezifischen Ausgestaltung der Norm DIN EN 61508.

Aufgrund einer fehlenden standardisierten Vorgehensweise für die Entwicklung sicherheitsrelevanter Systeme im Automobil gibt es zur Zeit einige Forschungs- und Entwicklungstätigkeiten auf diesem Gebiet, wie beispielsweise in [JW03] dargestellt. Diese Tätigkeiten orientieren sich nicht wie diese Arbeit an der Luftfahrtindustrie, sondern im Fall von [JW03] an der genannten Norm DIN EN 61508. Auch wenn es im Detail Unterschiede zur hier vorgestellten Vorgehensweise gibt, so sind die dort vorgeschlagenen Ergebnisse in ihren Grundzügen mit denen dieser Arbeit vergleichbar. Dies bestätigt die grundsätzliche Richtigkeit des in dieser Arbeit gewählten Ansatzes, der aber wie oben dargestellt für die praktische Umsetzung weiter ausgestaltet und mit Hilfe einer entsprechenden Werkzeugunterstützung auf die jeweiligen organisatorischen Randbedingungen angepasst werden muss.

Die Zukunft von trockenen X-by-Wire Systemen im Automobil ist weiter ungewiss. Spätestens in der nächsten Dekade werden X-by-Wire Systeme im Automobil sehr wahrscheinlich zum Einsatz kommen, darin sind sich die Experten einig. Aber gerade die Problematik der Produkthaftung insbesondere in Ländern wie den USA wird zukünftig bei solchen Systemen eine große Rolle spielen. Dies ist neben technischen Schwierigkeiten eine der Ursachen, dass solche Systeme nicht kurzfristig zu erwarten sind. Kein Hersteller und kein Zulieferer möchte ein zu großes wirtschaftliches und technisches Wagnis bei der Einführung solcher Systeme eingehen. Sicherheitstechnisch überdimensionierte Systeme werden aber aus Kostengründen keine Käufer finden, sollen sie nicht über Gebühr subventioniert werden.

Es wird spannend sein, zu verfolgen, in welcher Richtung die Entwicklung in der nächsten Zukunft voranschreiten wird. Gerade die Frage, ob sich das Vorgehen bei der Systementwicklung an der DIN EN 61508 orientieren wird, oder ob es wie hier in dieser Arbeit beschrieben der Luftfahrtbereich sein wird, ist noch offen. Auch eine ganz neue, branchenspezifische Lösung ist denkbar, aber wenig wahrscheinlich. Fakt ist aber, dass es bislang keine einheitliche genormte Vorgehensweise bei der Systementwicklung v. a. für die Betrachtung der Systemsicherheit im Automobil gibt. Dies wird sich mit Sicherheit in Zukunft ändern.

A. V-Modell ’97

Das V-Modell wurde ursprünglich im Auftrag des Bundesministeriums für Verteidigung und in Zusammenarbeit mit dem Bundesamt für Wehrtechnik und Beschaffung von der Industrieanlagen-Betriebsgesellschaft mbH (kurz IABG) erstellt. Im Sommer 1992 wurde es vom Bundesministerium des Innern für den Bereich der Bundesverwaltung übernommen und ist seit Juni 1996 auch dort eine verbindlich einzusetzende Vorschrift.

Im Jahre 1997 wurde es von der IABG als V-Modell ’97 (vgl. [IAB02a]) überarbeitet. Mittlerweile kommt es nicht nur für IT-Systeme des Bundes, sondern auch für etliche kommerzielle Systeme zum Einsatz.

Die Dokumentation des V-Modell ’97 umfasst das eigentliche Vorgehensmodell, eine Methodenzuordnung und Funktionale Werkzeuganforderungen. Das Vorgehensmodell besteht aus den vier Submodellen Projektmanagement (PM), Qualitätssicherung (QS), Konfigurationsmanagement (KM) und Systemerstellung (SE). Die vier Submodelle zeigt Abbildung A.1 auf der vorherigen Seite. Diese sind eng miteinander vernetzt und beeinflussen sich über den Austausch von Produkten bzw. Ergebnissen gegenseitig.

Das Submodell Projektmanagement plant, kontrolliert und informiert die Submodelle Systemerstellung, Qualitätssicherung und Konfigurationsmanagement. Das Submodell Systemerstellung erstellt das System bzw. die Software. Das Submodell Qualitätssicherung gibt Qualitätsanforderungen, Prüffälle und -kriterien vor und untersucht die Produkte und die Einhaltung der Standards. Das Submodell Konfigurationsmanagement schließlich verwaltet die erzeugten Produkte.

Das Vorgehensmodell zeichnet sich durch Allgemeingültigkeit sowie durch Firmen- und Projektunabhängigkeit aus. Um es für ein konkretes Projekt einzusetzen, kann individuell entschieden werden, welche Aktivitäten und Entwicklungsdokumente für das Projekt aus sachlichen Gründen erforderlich sind. In jedem Fall sollen übermäßige Papierflut, sinnlose Dokumente, aber auch das Fehlen wichtiger Dokumente verhindert werden. Diese projektspezifische Anpassung wird „Tailoring“ genannt

Besondere Bedeutung wird im Vorgehensmodell der Kritikalität der Software beigemessen. Dies ist ein Einstufungsmaß bezüglich Zuverlässigkeit, Sicherheit

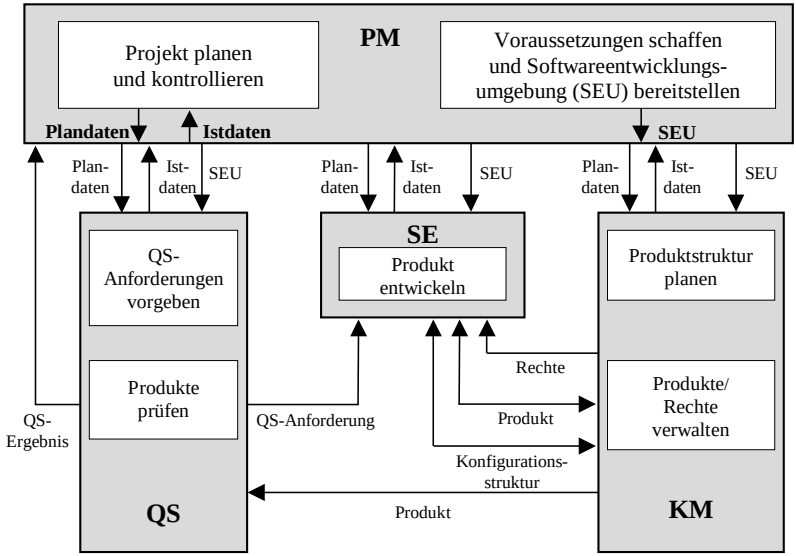

Abbildung A.1.: Die vier Submodelle des V-Modell ’97

oder allgemein der Wichtigkeit bestimmter Softwareanteile. Die Kritikalität wird im Vorgehensmodell als Qualitätsanforderung betrachtet und genau geregelt. Es werden Mechanismen vorgeschlagen, wie der Erstellungsaufwand und der Prüfaufwand den unterschiedlichen Kritikalitätsstufen der Software angepasst werden können.

Das V-Modell enthält Regelungen, die für die Erstellung sicherheitskritischer Software erforderlich sind. Die derzeit geltenden Sicherheitskriterien („ITSEC“ genannt) werden hinsichtlich ihrer Vorschriften zum Entwicklungsprozess durch die Anwendung des V-Modells erfüllt. Eine Zertifizierung der so entwickelten Software wird dadurch wesentlich erleichtert.

Im Anhang „SI - Sicherheit und Kritikalität“ werden Möglichkeiten zur Kritikalitätseinstufung und -zuordnung dargestellt und damit verbunden entsprechende Maßnahmen zur Abwehr der Auswirkung von Fehlverhalten vorgeschlagen.

A.1. SE - Submodell Systemerstellung

Während die drei Submodelle Qualitätssicherung, Projektmanagement und Konfigurationsmanagement die begleitenden Aktivitäten in einem Entwicklungsprojekt beschreiben, wird die Entwicklung selbst im Submodell Systemerstellung durchgeführt. Im Submodell Systemerstellung sind alle unmittelbar der Systemerstellung dienenden Aktivitäten und die jeweiligen Entwicklungsdokumente zusammengefasst.

Abbildung A.2 auf Seite 184 zeigt das Submodell Systemerstellung. Das Submodell Systemerstellung umfasst folgende Hauptaktivitäten:

SE 1 - System-Anforderungsanalyse Beschreibung der Anforderungen an das zu erstellende System und seine technische und organisatorische Umgebung; Durchführung einer Bedrohungs- und Risikoanalyse; Erarbeiten eines fachlichen Modells für Funktionen/Daten/Objekte.

Im Einzelnen besteht diese Hauptaktivität aus folgenden Aktivitäten:

- SE 1.1: Ist-Aufnahme/-Analyse durchführen
- SE 1.2: Anwendungssystem beschreiben
- SE 1.3: Kritikalität und Anforderungen an die Qualität definieren
- SE 1.4: Randbedingungen definieren
- SE 1.5: System fachlich strukturieren
- SE 1.6: Bedrohung und Risiko analysieren
- SE 1.7: Forderungscontrolling durchführen
- SE 1.8: Software-Pflege- und -Änderungs-Konzept erstellen

SE 2 - System-Entwurf Zerlegung des Systems in Segmente sowie Software- und Hardware-Einheiten.

Im Einzelnen besteht diese Hauptaktivität aus folgenden Aktivitäten:

- SE 2.1: System technisch entwerfen
- SE 2.2: Wirksamkeitsanalyse durchführen

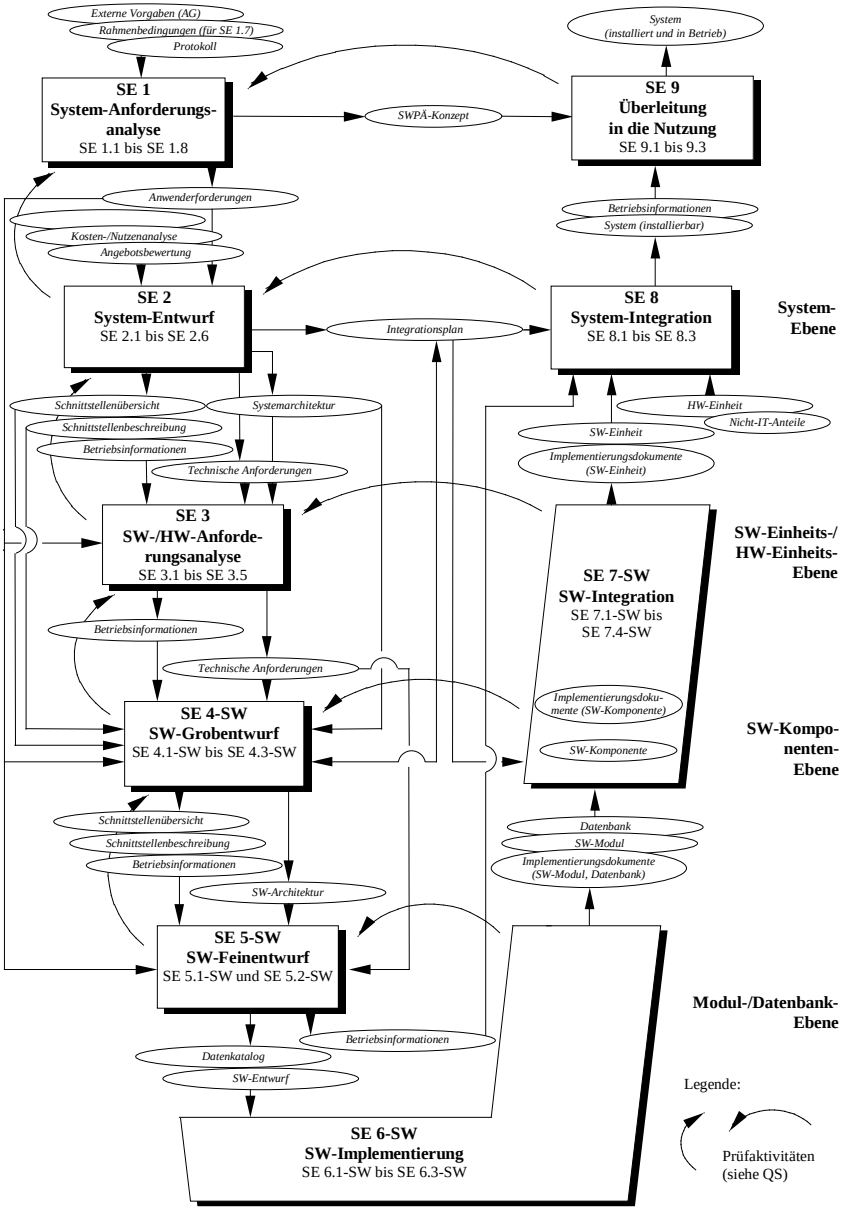

Abbildung A.2.: Das Submodell Systemerstellung

- SE 2.3: Realisierbarkeit untersuchen
- SE 2.4: Anwenderforderungen zuordnen
- SE 2.5: Schnittstellen beschreiben
- SE 2.6: System-Integration spezifizieren

SE 3 - SW-/HW-Anforderungsanalyse Die technischen Anforderungen an die SW- und gegebenenfalls HW-Einheiten werden präzisiert. Von hier ab spaltet sich der weitere Fortgang in die SW-Entwicklung und gegebenenfalls in die HW-Entwicklung.

Im Einzelnen besteht diese Hauptaktivität aus folgenden Aktivitäten:

- SE 3.1: Allgemeine Anforderungen aus Sicht der SW-/HW-Einheit definieren
- SE 3.2: Anforderungen an die externen Schnittstellen der SW-/HW-Einheit präzisieren
- SE 3.3: Anforderungen an die Funktionalität definieren
- SE 3.4: Anforderungen an die Qualität der SW-/HW-Einheit definieren
- SE 3.5: Anforderungen an Entwicklungs- und Software-Pflege- und Änderungs-Konzept-Umgebung definieren

Softwareerstellung

Die SW-Erstellung einer SW-Einheit wird in vier Schritten (SE 4-SW bis SE 7-SW) durchgeführt.

Im Einzelnen bestehen diese Hauptaktivitäten aus folgenden Aktivitäten:

SE 4-SW - SW-Grobentwurf

- SE 4.1-SW: SW-Architektur entwerfen
- SE 4.2-SW: SW-interne und -externe Schnittstellen entwerfen
- SE 4.3-SW: SW-Integration spezifizieren

SE 5-SW - SW-Feinentwurf

- SE 5.1-SW: SW-Komponente/-Modul/Datenbank beschreiben
- SE 5.2-SW: Betriebsmittel- und Zeitbedarf analysieren

SE 6-SW - SW-Implementierung

- SE 6.1-SW: SW-Module codieren
- SE 6.2-SW: Datenbank realisieren
- SE 6.3-SW: Selbstprüfung des SW-Moduls/der Datenbank durchführen

SE 7-SW - SW-Integration

- SE 7.1-SW: Zur SW-Komponente integrieren
- SE 7.2-SW: Selbstprüfung der SW-Komponente durchführen
- SE 7.3-SW: Zur SW-Einheit integrieren
- SE 7.4-SW: Selbstprüfung der SW-Einheit durchführen

Hardwareerstellung

Auch die HW-Erstellung einer HW-Einheit erfolgt in vier Schritten (SE 4-HW bis SE 7-HW).

Im Einzelnen bestehen diese Hauptaktivitäten aus folgenden Aktivitäten:

SE 4-HW - HW-Grobentwurf

- SE 4.1-HW: Lösungsvorschläge erarbeiten und bewerten
- SE 4.2-HW: Grobentwurf einer HW-Einheit erarbeiten
- SE 4.3-HW: HW-interne Schnittstellen spezifizieren
- SE 4.4-HW: HW-Integration spezifizieren

SE 5-HW - HW-Feinentwurf

- SE 5.1-HW: Detailentwürfe für HW-Komponenten/HW-Module erstellen
- SE 5.2-HW: Zeichnungssatz erstellen
- SE 5.3-HW: Analysen und Nachweise durchführen

SE 6-HW - HW-Realisierung

- SE 6.1-HW: HW-Komponente/HW-Modul anfertigen
- SE 6.2-HW: Selbstprüfung durchführen

SE 7-HW - HW-Integration

- SE 7.1-HW: Zur HW-Teilstruktur integrieren
- SE 7.2-HW: HW-Teilstrukturen-Selbstprüfung durchführen
- SE 7.3-HW: Zur HW-Einheit integrieren
- SE 7.4-HW: HW-Einheiten-Selbstprüfung

SE 8 - System-Integration Integration der verschiedenen Software- und Hardwareeinheiten zu einem Segment und Integration der Segmente (falls vorhanden) zum System.

Im Einzelnen besteht diese Hauptaktivität aus folgenden Aktivitäten:

- SE 8.1: Zum System integrieren
- SE 8.2: Selbstprüfung des Systems durchführen
- SE 8.3: Produkt bereitstellen

SE 9 - Überleitung in die Nutzung Beschreibung aller Tätigkeiten, die notwendig sind, um ein fertiggestelltes System an der vorgesehenen Einsatzstelle zu installieren und in Betrieb zu nehmen.

Im Einzelnen besteht diese Hauptaktivität aus folgenden Aktivitäten:

- SE 9.1: Beitrag zur Einführungsunterstützung leisten
- SE 9.2: System installieren
- SE 9.3: In Betrieb nehmen

A.2. QS - Submodell Qualitätssicherung

Das Submodell Qualitätssicherung regelt die Aufgaben und Funktionen der Qualitätssicherung innerhalb des System- bzw. Softwareentwicklungsprozesses. Im Gegensatz zu den informellen Prüfungen im Submodell Systemerstellung wird hier im Rahmen einer Nachweisführung objektiv nachvollziehbar die Erfüllung vorgegebener Anforderungen gezeigt. Diese Anforderungen finden sich in den Dokumenten „Anwenderforderungen“ und „Technische Anforderungen“ des Submodells Systemerstellung. Die Regelungen berühren jedoch wie bei den anderen Submodellen in keiner Weise organisatorische Festlegungen.

Abbildung A.3 auf der vorherigen Seite zeigt das Submodell Qualitätssicherung. Das Submodell Qualitätssicherung umfasst folgende Hauptaktivitäten:

QS 1 - QS-Initialisierung Die QS-Initialisierung legt den organisatorischen und abwicklungstechnischen Rahmen im QS-Plan und in Prüfplänen fest.

QS 2 - Prüfungsvorbereitung Zur Prüfungsvorbereitung gehören die Erstellung von Prüfspezifikation und -prozedur und die Vervollständigung des Prüfplans sowie Anforderungen an die Prüfumgebung. Die Prüfkriterien müssen so festgelegt werden, dass der Erfolg oder Misserfolg einer Prüfung eindeutig und nachvollziehbar entschieden werden kann.

QS 3 - Prozessprüfung von Aktivitäten Bei der Prozessprüfung wird festgestellt, ob vorgegebene Vorgehensweisen bei der Durchführung bestimmter Aktivitäten eingehalten werden.

QS 4 - Produktprüfung Die Produktprüfung erfolgt in zwei Stufen: Prüfung der formalen Kriterien und inhaltliche Prüfung des Produktes. Für die Prüfungen sind die Prüfspezifikationen zu verwenden. Das Ergebnis wird in einem Prüfprotokoll festgehalten.

QS 5 - QS-Berichtswesen Hier sind in regelmäßigen Abständen die Prüfprotokolle nach vorgegebenen Kriterien auszuwerten und die Ergebnisse dem Projektmanagement vorzulegen.

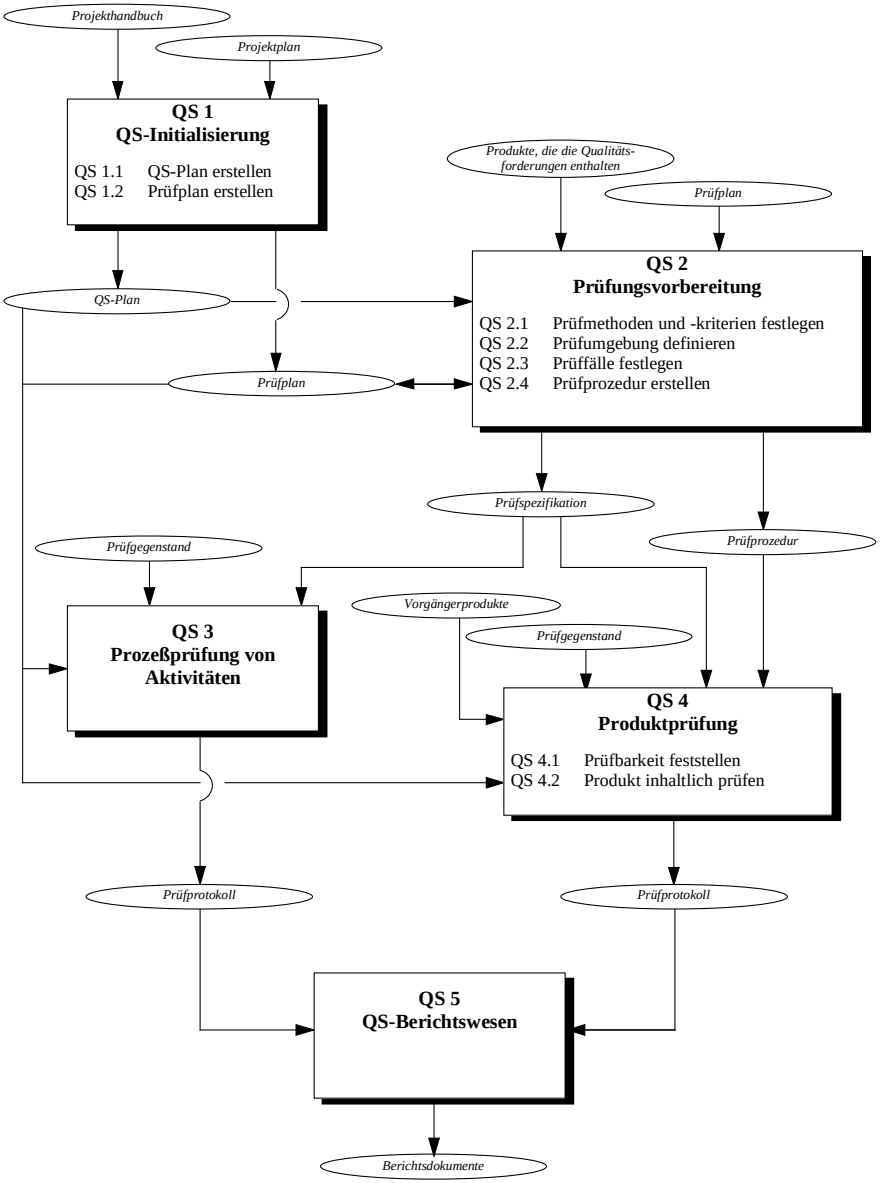

Abbildung A.3.: Das Submodell Qualitätssicherung

A.3. KM - Submodell Konfigurationsmanagement

Das Submodell Konfigurationsmanagement stellt sicher, dass Produkte eindeutig identifizierbar sind, Zusammenhänge und Unterschiede von verschiedenen Versionen einer Konfiguration erkennbar bleiben und Produktänderungen nur kontrolliert durchgeführt werden können.

Abbildung A.4 auf der vorherigen Seite zeigt das Submodell Konfigurationsmanagement. Das Submodell Konfigurationsmanagement umfasst folgende Hauptaktivitäten:

KM 1 - KM-Initialisierung Die KM-Initialisierung regelt den organisatorischen und abwicklungstechnischen Rahmen im KM-Plan. Des weiteren sind die Einsatzmittel (Produktbibliothek, Werkzeuge) bereitzustellen.

KM 2 - Produkt- und Konfigurationsverwaltung Die Produkt- und Konfigurationsverwaltung umfasst das Verwalten von Produkten, Konfigurationen und Rechten. Die Verwaltung einer Konfiguration geschieht über das Konfigurationsidentifikationsdokument, das einen Überblick über die Struktur und alle zu einer Konfiguration gehörenden Anteile aufzeigt.

KM 3 - Änderungsmanagement Über das Änderungsmanagement werden eingehende Fehlermeldungen, Problemmeldungen, Verbesserungsvorschläge usw. erfasst und über die im KM-Plan festgeschriebenen Änderungsprozeduren einer kontrollierten Bearbeitung zugeführt. Die Durchführung einer Änderung selbst wird gemäß den Regelungen des V-Modells abgewickelt.

KM 4 - KM-Dienste Unter KM-Dienste werden allgemeine Serviceleistungen wie Produkte katalogisieren, Daten administrieren, Schnittstellen koordinieren usw. zusammengefasst.

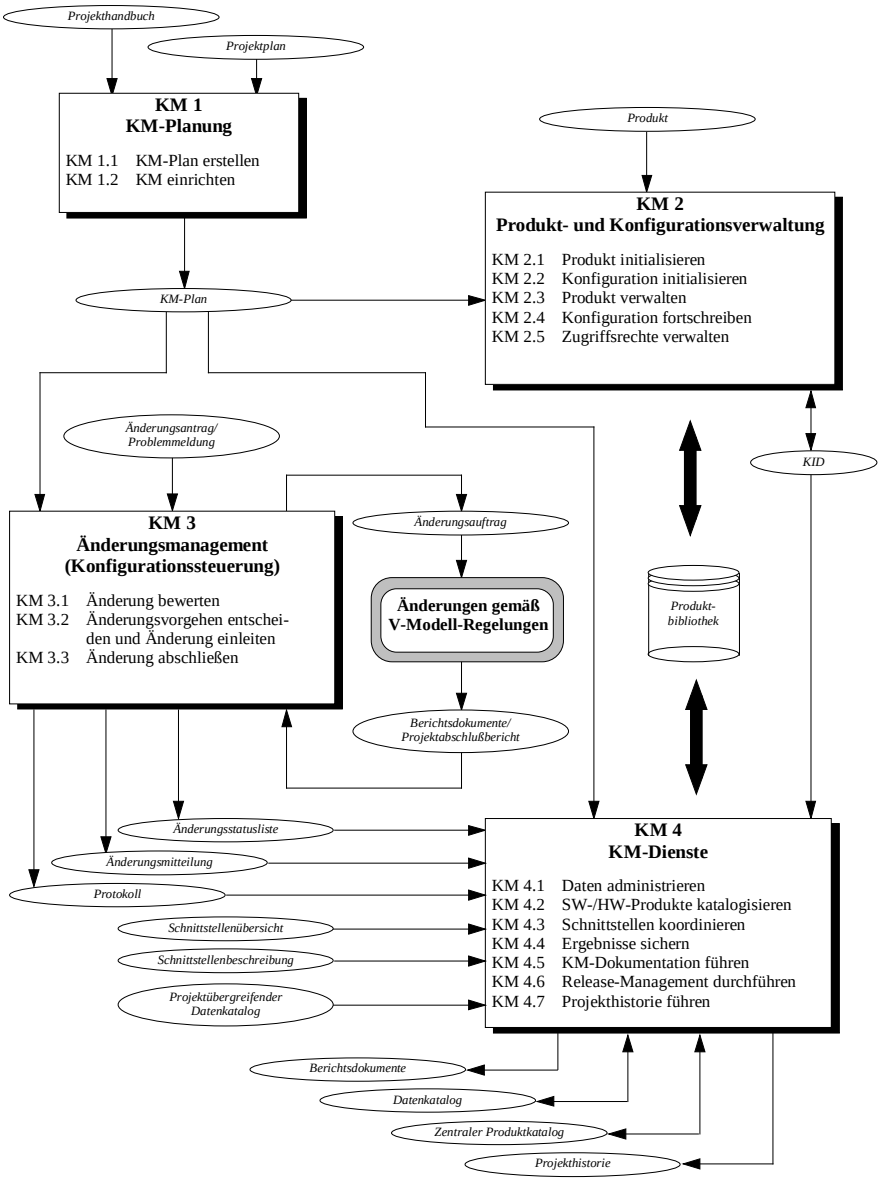

Abbildung A.4.: Das Submodell Konfigurationsmanagement

A.4. PM - Submodell Projektmanagement

Das Submodell Projektmanagement regelt die Aufgaben und Funktionen des technischen Projektmanagements innerhalb des Entwicklungsprozesses. Diese Regelungen berühren keinerlei organisatorische Festlegungen. Die im Submodell Projektmanagement festgelegten Tätigkeiten umfassen Planung, Kontrolle und Steuerung projektinterner Tätigkeiten, die Zuordnung projektinterner Rollen und die Einrichtung einer Schnittstelle zu projektexternen Einheiten (Auftragnehmer).

Abbildung A.5 auf Seite 193 zeigt das Submodell Projektmanagement.

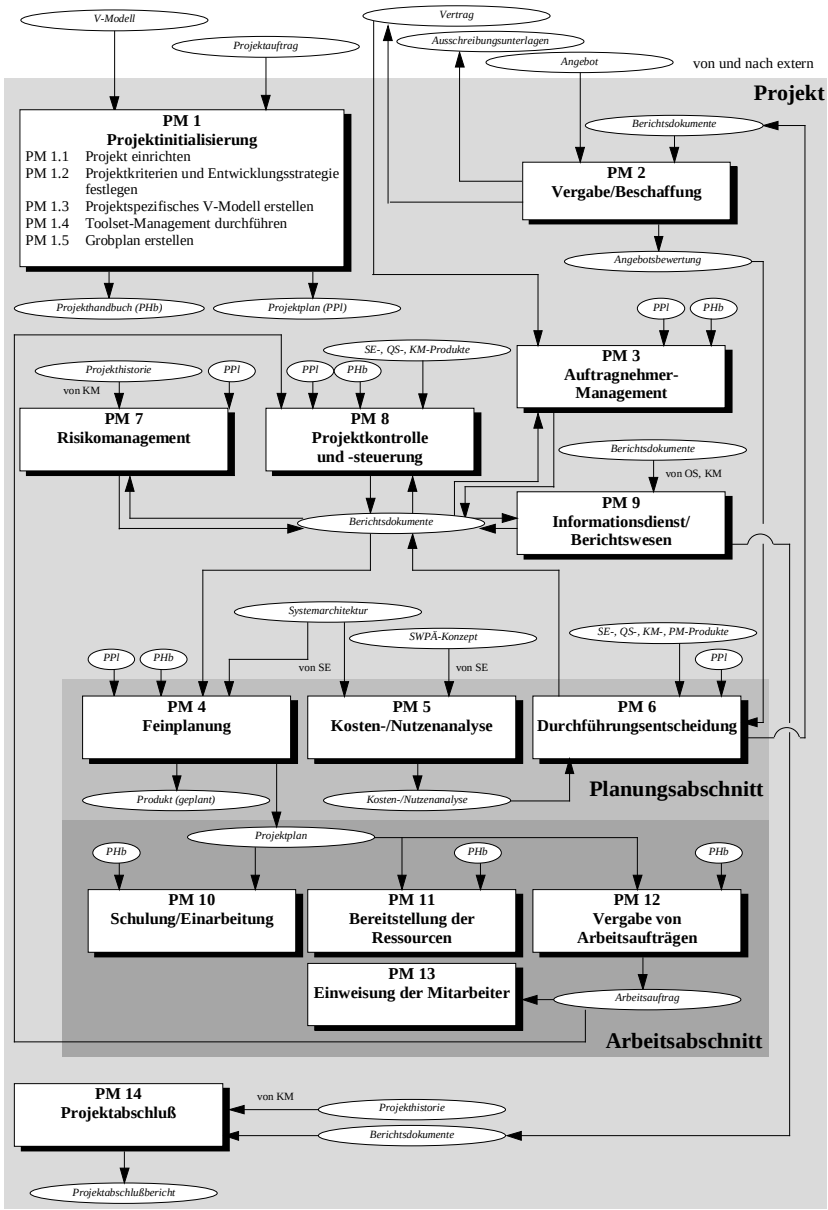

Abbildung A.5.: Das Submodell Projektmanagement

B. SAE ARP 4754 und SAE ARP 4761

In „SAE Aerospace Recommended Practice 4754: Certification Considerations for highly-integrated or complex aircraft systems“ und „SAE Aerospace Recommended Practice 4761: Guidelines und methods for conducting the safety assessment process on civil airborne systems and equipment“ wird ein einheitlicher Prozess zur Entwicklung und Zulassung ziviler Luftfahrtsysteme beschrieben.

Die beiden Standards wurden von der der Society of Automotive Engineers SAE, die Standards sowohl für Luftfahrt- als auch für Automobiltechnik-Anwendungen herausgibt, im Auftrag der Federal Aviation Administration FAA, der amerikanischen Bundesluftfahrtbehörde, entwickelt. Die aktuellen Fassungen stammen aus dem Jahr 1996. Durch die bestehende Harmonisierungsbestrebungen zwischen der FAA und den Joint Aviation Authorities JAA, dem europäischen Zusammenschluss nationaler Luftfahrtbehörden, sind sie implizit weltweit gültig.

SAE ARP 4754 behandelt Zertifizierungsaspekte hochintegrierter oder komplexer Elektroniksysteme an Bord eines großen Verkehrsflugzeugs. Es werden dabei keine exakten Handlungsanweisungen gegeben, im Vordergrund steht die Übersicht über die Anwendung bestehender Standards und Normen. Ein Systementwicklungsprozess wird nicht detailliert beschrieben, es wird auf andere Standards verwiesen (beispielsweise auf DO-178B, vgl. [DO 92]), für die Beschreibung des Sicherheitsprozesses wird der Standard SAE ARP 4761 referenziert (vgl. dazu auch Abbildung B.1 auf der nächsten Seite). In SAE ARP 4761 sind darüber hinaus Hinweise auf geeignete Methoden für jede Stufe des Sicherheitsprozesses enthalten.

Gemäß SAE ARP 4754 laufen der Sicherheitsprozess von SAE ARP 4761 und der jeweilige Systementwicklungsprozess parallel ab (vgl. Abbildung B.2 auf Seite 198). Der Sicherheitsprozess beeinflusst dabei frühzeitig die Entwicklung der Architektur des Systems, wodurch eine Dimensionierung des Systems entsprechend der erforderlichen Sicherheit erreicht wird.

Der Sicherheitsprozess von SAE ARP 4761 beginnt mit der Formulierung von Sicherheitsanforderungen an das Gesamtsystem „Flugzeug“. Dazu gehören Zielwerte für mögliche Gefährdungen, denen maximale Auftretenswahrscheinlichkeiten zugeordnet sind.

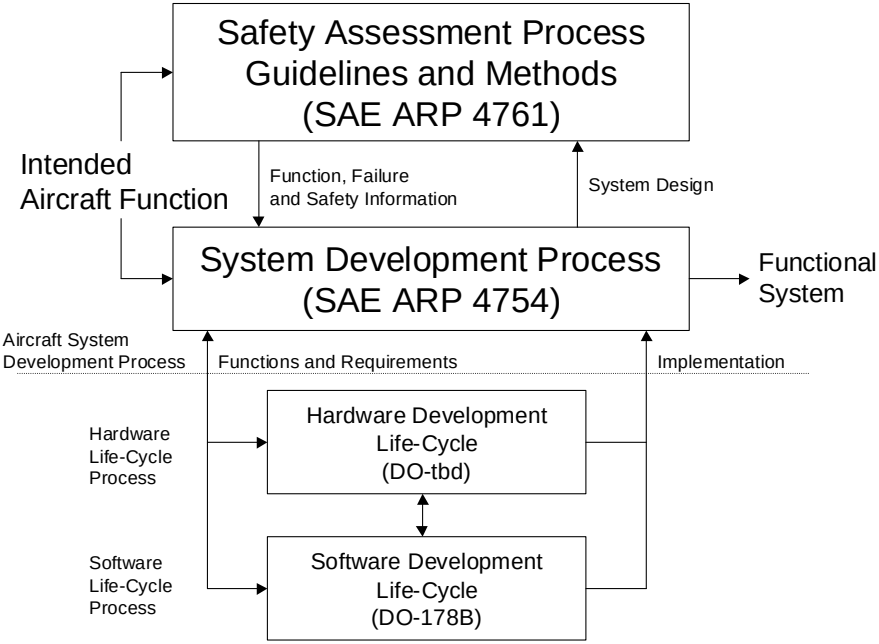

Abbildung B.1.: Systementwicklung im Luftfahrtbereich gemäß SAE ARP 4754

Diese sind auf Basis von FAR 25.1309[1] „Equipment, Systems and Installation“ bzw. JAR 25.1309[2] „Equipment, Systems and Installation“ als Mindestanforderungen festgelegt. In Tabelle B.1 auf der nächsten Seite sind diese Zuordnungen dargestellt.

Der Sicherheitsprozess nach SAE ARP 4761 findet sich in Abbildung B.3 auf Seite 199. Die wesentlichen Elemente der Sicherheitsprozesses sind FHA („Functional Hazard Assessment“), PSSA („Preliminary System Safety Assessment“) und SSA („System Safety Assessment“). Diese Prozesselemente sowie die verwendete Methode CCA („Common Cause Analysis“) werden im Folgenden kurz vorgestellt.

[1] FAR: Federation Aviation Regulations, herausgegeben von der Federal Aviation Administration FAA
[2] JAR: Joint Aviation Requirements, herausgegeben von den Joint Aviation Authorities JAA

Tabelle B.1.: Auswirkungen von Gefährdungen bezogen auf ihre Auftretenswahrscheinlichkeit

		pro Flugstunde				
Wahrscheinlichkeit (quantitativ)		10^{-3}		10^{-5}	10^{-7}	10^{-9}
Wahrscheinlichkeit (beschreibend)	FAR	probable (wahrscheinlich)		improbable (unwahrscheinlich)		extremely improbable (nahezu unmöglich)
	JAR	frequent (häufig)	reasonably probable (möglich)	remote (unwahrscheinlich)	extremely remote (sehr unwahrscheinlich)	extremely improbable (nahezu unmöglich)
Fehlerauswirkungsklassifikation	FAR	minor (unbedeutend)		major (bedeutend)	severe major (schwerwiegend)	catastrophic (katastrophal)
	JAR	minor (unbedeutend)		major (bedeutend)	hazardous (gefährlich)	catastrophic (katastrophal)
Fehlerauswirkung	FAR & JAR	leichte Beeinträchtigung der Sicherheitsfaktoren leichte Erhöhung der Arbeitsbelastung der Besatzung leichte Unannehmlichkeiten für die Passagiere		Beeinträchtigung der Sicherheitsfaktoren oder der Funktionalität bedeutende Erhöhung der Arbeitsbelastung der Besatzung oder Verschlechterung der Arbeitseffektivität größere Unannehmlichkeiten für die Passagiere	starke Beeinträchtigung der Sicherheitsfaktoren oder der Funktionalität Überlastung der Besatzung, die zu unvollständiger oder unzureichender Aufgabenerfüllung führen kann widrige Zustände für die Passagiere	alle Gefährdungen, die eine Fortsetzung eines sicheren Fluges inklusive einer Landung verhindern

Quelle: SAE ARP 4761, Übersetzung [MP03]

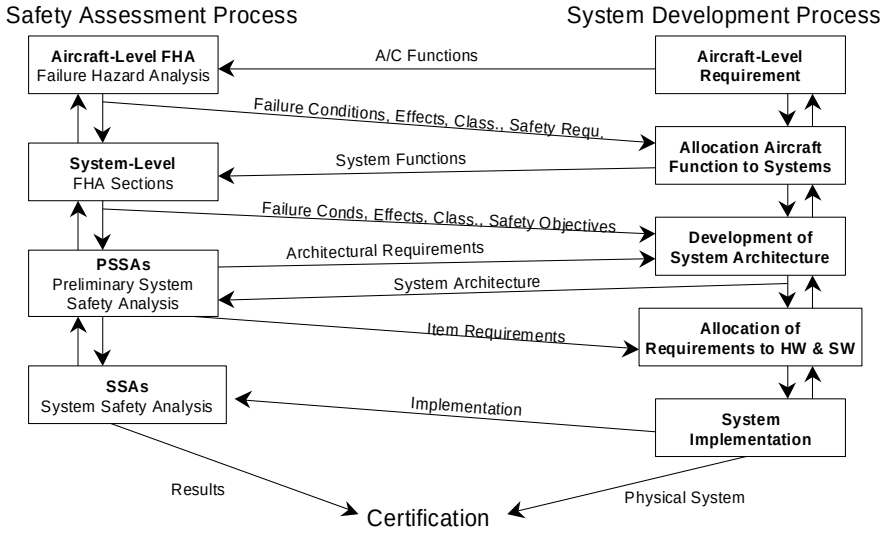

Abbildung B.2.: Der „Safety Assessment Process“ und der „System Development Process“ nach SAE ARP 4754

B.1. FHA: Functional Hazard Assessment

Die FHA Functional Hazard Assessment (dt. „Funktionale Gefährdungsanalyse“) ist eine systematische, umfassende Untersuchung von Funktionen eines Systems mit dem Ziel, Gefährdungen dieser Funktionen zu identifizieren und entsprechend ihrer Schwere zu klassifizieren. Üblicherweise wird eine FHA auf zwei Ebenen durchgeführt. Die zugehörigen FHAs werden entsprechend als „aircraft level FHA“ und „system level FHA“ bezeichnet.

Die „aircraft level FHA“ ist eine abstrakte, qualitative Bewertung der Grundfunktionen eines Flugzeugs, wie sie zu Beginn der Flugzeugentwicklung festgelegt werden. Die Klassifikation dieser Gefährdungen legt die Sicherheitsanforderungen fest, die das Flugzeug erfüllen muss. Ziel bei der Durchführung der „aircraft level FHA“ ist es, jede Gefährdung und die Ursachen für ihre Klassifikation zu identifizieren. Für jede potenziell auftretende Gefährdung einer bestimmten Schwere wird eine maximal tolerierbare Auftretenswahrscheinlichkeit angegeben (vgl. Tabelle B.1).

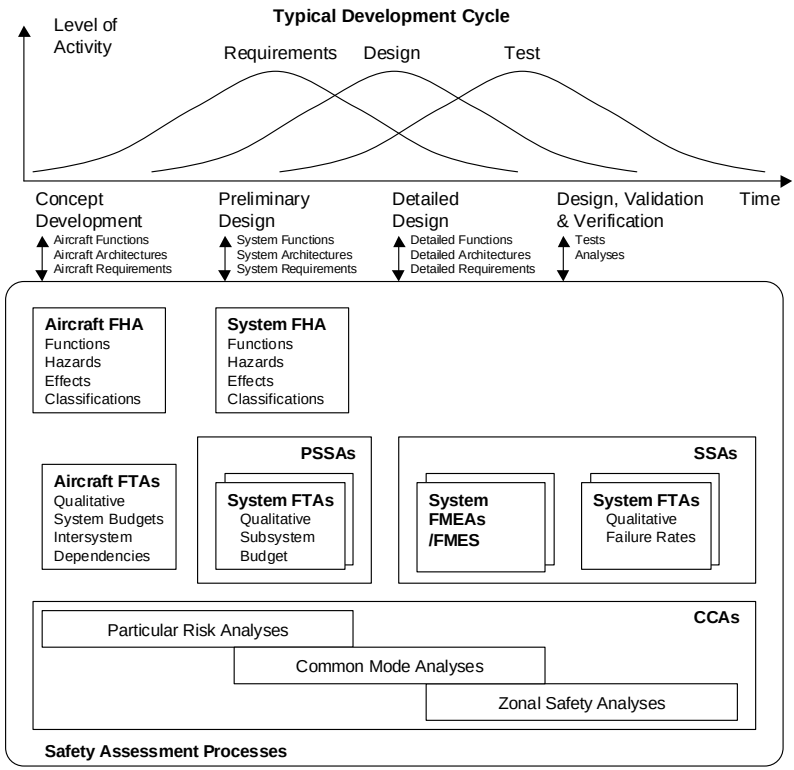

Abbildung B.3.: Der Sicherheitsprozess nach SAE ARP 4761

Die „system level FHA“ ist eine ebenfalls qualitative Bewertung iterativer Art. Sie wird immer detaillierter, je weiter sich das System entwickelt. Sie berücksichtigt einen Fehler oder Kombinationen von Systemfehlern, die eine Flugzeugfunktion beeinflussen. Eine Bewertung einer bestimmten Hardware oder Software ist nicht Ziel dieser Bewertung.

Das Ergebnis der „aircraft level FHA“ bzw. „system level FHA“ ist der Anknüpfungspunkt für die Zuordnung von Sicherheitsanforderungen. Beispielsweise kann ein Fehlerbaum dazu verwendet werden, Anforderungen einer niedrigeren Ebene abzuleiten.

B.2. PSSA: Preliminary System Safety Assessment

Gegenstand der Preliminary System Safety Analysis PSSA (dt. „Vorläufige System-Sicherheitsbewertung“) ist die Beantwortung der Frage „Wie sicher ist die Systemarchitektur?“.

Die PSSA besteht aus zwei Kernelementen, der Validierung der Systemarchitektur und der Ableitung von Sicherheitsanforderungen. Das Ziel der Validierung der Systemarchitektur ist es, zu zeigen, dass die Systemarchitektur mit ausreichender Wahrscheinlichkeit die Sicherheitsziele des Systems erfüllen kann, wie sie in der FHA festgelegt wurden. Ziel der Ableitung der Sicherheitsanforderungen ist, Anforderungen an die Systemelemente festzulegen, die, wenn sie erfüllt werden, eine Einhaltung der in der FHA festgelegten Sicherheitsziele gewährleisten.

Diese beiden Punkte sind miteinander verknüpft. Die Systemarchitektur kann nur die Sicherheitsziele der FHA erfüllen, wenn die Architekturelemente ihre Sicherheitsanforderungen erfüllen. Die PSSA unterstützt den Entwicklungsprozess, so dass die gewählte Systemarchitektur aus Sicht von Sicherheit und Zuverlässigkeit sinnvoll ist.

Dazu werden die funktionalen Sicherheitsanforderungen aus der FHA zuerst auf Teilsysteme und im weiteren auf Hardware und Software zugeordnet sowie die Notwendigkeit alternativer Schutzmechanismen (wie z. B. Überwachung, Redundanz oder Diversität) ermittelt. Zu diesem Zweck werden verschiedene Methoden wie z. B. Fehlerbaumanalyse oder Markov-Analyse verwendet. Dabei werden die Ursachen für Gefährdungen ermittelt sowie Abhängigkeiten zwischen Funktionen und Gefährdungen untersucht.

Die PSSA ist die Grundlage für die Zuordnung von Zuverlässigkeitsanforderungen an Architekturelemente. So wird durch die PSSA der Nachweis erbracht, dass das Gesamtsystem die Sicherheitsanforderungen aus der FHA erfüllt.

B.3. SSA: System Safety Assessment

Gegenstand der System Safety Analysis SSA (dt. „System Safety Assessment“) ist die Beantwortung der Frage „Wie sicher ist das implementierte System?“.

Die SSA umfasst den Sicherheitsnachweis für das System. Durch die SSA wird nachgewiesen, dass die Anforderungen aus FHA und PSSA durch das System erfüllt werden.

Die SSA ist ein ständiger Prozess, der während der gesamten Implementierungsphase der Systemelemente, der Systemintegration und auch den nachfolgenden

Phasen parallel durchgeführt wird. Die SSA beginnt, sobald das detaillierte Design des Systems vorliegt und wird immer wieder aktualisiert, wenn sich das System im Laufe der Entwicklung verändert. Sie endet erst mit der Außerbetriebnahme des Systems. Ziel der SSA ist es dabei sicherzustellen, dass jedes Element in seiner Implementierung ausreichend sicher für seine spezifizierte Aufgabe ist.

Innerhalb der SSA gibt es zwei Teile, Safety Analysis und Safety Engineering genannt.

Die Sicherheitsanalyse („Safety Analysis“) in der SSA ist ähnlich der in der PSSA, verfolgt aber ein anderes Ziel. Während in der PSSA Anforderungen an Systemelemente ermittelt werden, wird durch die SSA sichergestellt, dass diese Anforderungen erfüllt werden. Für jedes Systemelement wird durch eine solche Analyse die Auswirkung ihrer normalen Funktion und eines Ausfalls auf das Gesamtsystem bewertet.

Wie in der PSSA werden die Ergebnisse dieser Analysen dazu benutzt, Mittel zur Risikoabschwächung zu identifizieren und Designänderungen im Hinblick auf die Systemsicherheit zu evaluieren („Safety Engineering“). Dies erfolgt durch Einsatz geeigneter Entwicklungsstandards oder durch Verifikation, ob die Sicherheitsanforderungen an das System eingehalten werden.

Vor der Inbetriebnahme des Systems wird durch die SSA demonstriert, dass alle Risiken im System eliminiert oder ausreichend minimiert wurden und dass das System die Sicherheitsanforderungen erfüllt.

B.4. CCA: Common Cause Analysis

Die meisten der in SAE ARP 4761 referenzierten Methoden sind in Kapitel 2 vorgestellt worden, so z. B. die Fehlerbaumanalyse FTA in Kapitel 2.4.2 oder die FMEA in Kapitel 2.4.1. Hier wird daher nur die noch nicht erwähnte Common Cause Analysis CCA kurz dargestellt.

Die Common Cause Analysis CCA, auf deutsch „Analyse von Fehlern gemeinsamer Ursachen“ genannt, besteht aus den drei Teilen Zonal Safety Analysis ZSA (dt. „Zonensicherheitsanalyse“), Particular Risk Analysis PRA (dt. „Analyse besonderer Risiken“) und Common Mode Analysis CMA (dt. „Analyse redundanzüberbrückender Fehler“) (vgl. [MP03]).

Die ZSA wird für bestimmte Zonen eines Flugzeugs durchgeführt, um sicherzustellen, dass die System- und Geräteinstallation die Sicherheitsanforderungen erfüllt. Hierzu zählen die Überprüfung der Basisinstallation, Wechselwirkungen

zwischen Systemen bzw. Geräten, Auswirkungen von Fehlern auf benachbarte Systeme und Geräte sowie Strukturen sowie Instandhaltungsfehler.

Die PRA analysiert besondere Risiken, die in der Regel von außen auf das System wirken, wie Feuer, Hagel, Schnee, Blitzschlag, Bersten einer Radfelge, Leckage (Kraftstoff, Hydraulik, Batteriesäure, Wasser etc.), u. a.

Hingegen wird durch die CMA verifiziert, dass die z. B. in einer Fehlerbaumanalyse als unabhängig angenommenen Basisereignisse auch tatsächlich unabhängig sind. Hierzu zählen Soft- und Hardwarefehler, Instandsetzungsfehler, Umgebungsfaktoren (Temperatur, Vibrationen etc.), Spezifikationsfehler, u. a.

C. DIN EN 61508

Die Norm DIN EN 61508 „Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme (E/E/PES)“ behandelt die Entwicklung sicherheitsbezogener Systeme. Sie ist die deutsche Entsprechung der internationalen Norm IEC 61508 „Functional safety of electrical/electronic/programmable electronic safety-related systems“.

Der Ursprung der Norm liegt in der chemisch-prozesstechnischen Industrie, was sich auch in der Begriffswahl der Norm niederschlägt. So wird in der Norm beispielsweise stillschweigend von der in der Prozessautomatisierung üblichen Systemstruktur nach Abbildung C.1 ausgegangen.

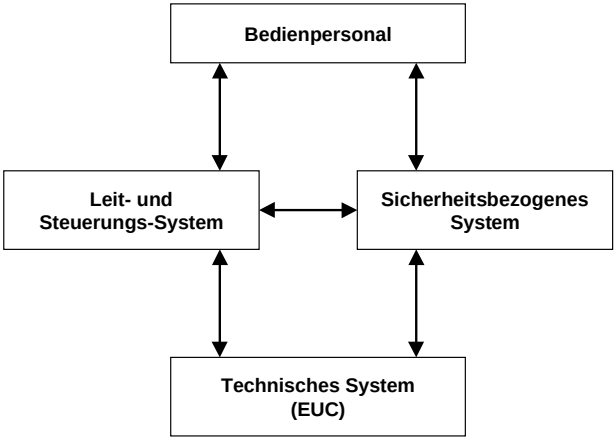

Abbildung C.1.: Bestandteile eines Prozessautomatisierungssystems

Ein sicherheitsbezogenes System nach dem Verständnis der Norm ist „... ein System, das sowohl die erforderlichen Sicherheitsfunktionen ausführt, die notwendig sind, um einen sicheren Zustand, für das Technische System (sog. EUC[1]) zu

[1] EUC: Equipment Under Control

erreichen oder aufrechtzuerhalten, als auch dazu vorgesehen ist, selbst, oder mit anderen E/E/PE-Systemen, sicherheitsbezogenen Systemen anderer Technologie oder externen Einrichtungen zur Risikominderung, die notwendige Sicherheitsintegrität für die geforderten Sicherheitsfunktionen zu erreichen“.

Unter Funktionaler Sicherheit wird „... der Teil der Gesamtsicherheit verstanden, bezogen auf die EUC und das EUC-Leit- oder Steuerungssystem, der von der korrekten Funktion des E/E/PE-sicherheitsbezogenen Systems anderer Technologie und externer Einrichtungen zur Risikominderung abhängt“.

Grundlage jeglicher Sicherheitsbetrachtung ist eine Einstufung der jeweiligen EUC in vier sogenannte Sicherheitsintegritätslevel (SIL). Unter Sicherheitsintegrität versteht die Norm „... die Wahrscheinlichkeit, dass ein sicherheitsbezogenes System die geforderten Sicherheitsfunktionen unter allen festgelegten Bedingungen innerhalb eines festgelegten Zeitraumes ordnungsgemäß ausführt“.

Die Norm IEC 61508 entwickelte sich aus IEC 1508 und ist eine generische Norm, aus der anwendungs- bzw. branchenspezifische Entsprechungen abgeleitet werden sollen. IEC 61508 ist somit nicht zur direkten Anwendung gedacht, allerdings soll die Norm jeweils zum Einsatz kommen, wenn es keine branchenspezifische Entsprechung gibt. Existierende branchenspezifische Ausprägungen der IEC 61508 sind beispielsweise EN 50126 für die Eisenbahn, IEC 61513 im Kerntechnikbereich, IEC 60601 für die Medizintechnik oder IEC 50156 für Feuerungseinrichtungen. Für den Automobilbereich existiert eine solche Entsprechung bislang nicht.

IEC 61508 wurde 1998 von der International Electrotechnical Commission IEC veröffentlicht, im Jahr 2002 wurde sie als DIN EN 61508 ins deutsche Normenwerk aufgenommen. Im August 2004 werden Vorgänger-Normen bzw. ihr entgegenstehende Normen wie DIN V VDE 0801 „Grundsätze für Rechner in Systemen mit Sicherheitsaufgaben“, DIN V 19250 „Grundlegende Sicherheitsbetrachtungen für MSR-Schutzeinrichtungen“ und DIN V 19251 „MSR-Schutzeinrichtungen - Anforderungen und Maßnahmen zur gesicherten Funktion“ zurückgezogen.

Die Norm DIN EN 61508 (bzw. IEC 61508) besteht aus insgesamt sieben Teilen. „Teil 1: Allgemeine Anforderungen“ stellt das Grundgerüst der Norm, den sogenannten Sicherheitslebenszyklus vor und erklärt zugrunde liegende Grundprinzipien. Er ist in Abbildung C.2 auf Seite 205 dargestellt.

Der Sicherheitslebensyklus gliedert sich im Einzelnen in folgende 16 Schritte: *Konzept, Definition des gesamten Anwendungsbereiches, Gefährdungs- und Risikoanalyse, Gesamte Sicherheitsanforderungen, Zuweisung der Sicherheitsanforderungen, Planung des Gesamtbetriebs und der gesamten Wartung, Planung der Sicherheits-Gesamtvalidierung, Planung der*

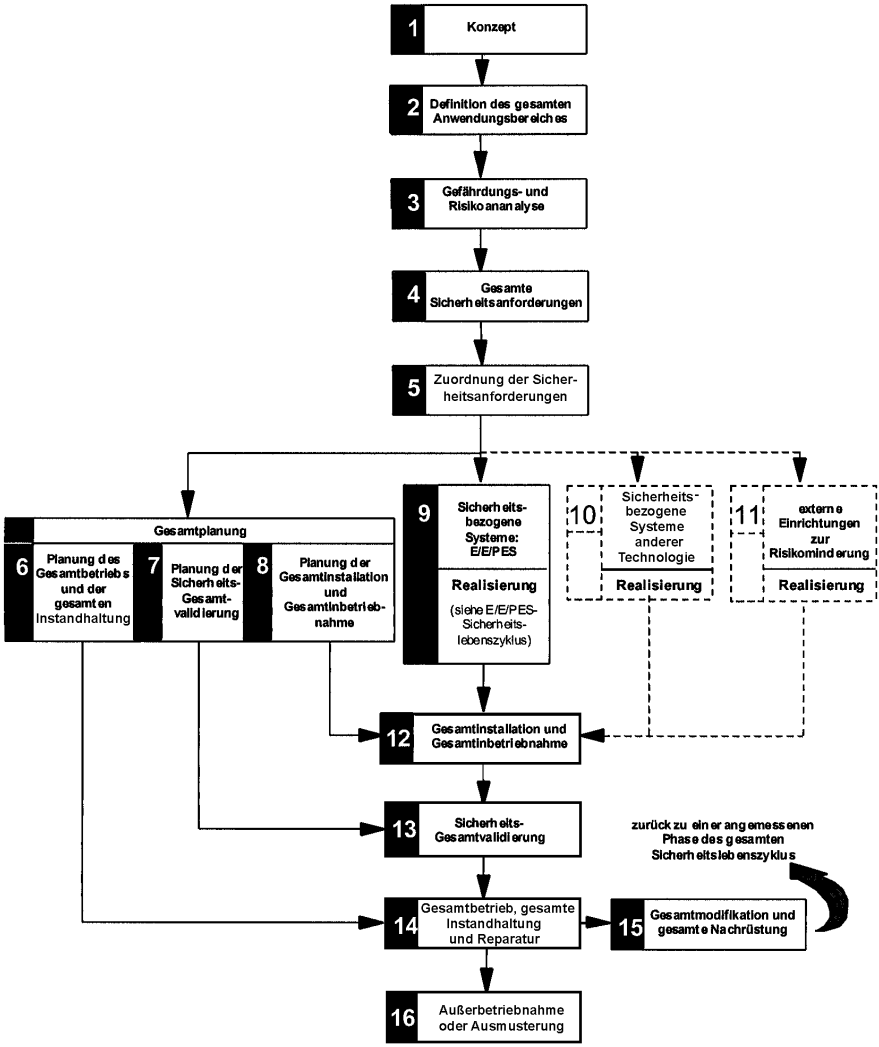

Abbildung C.2.: Der Sicherheitslebenszyklus von DIN EN 61508

Gesamtinstallation und Gesamtinbetriebnahme, *Realisierung der elektrischen/elektronischen/programmierbar elektronischen sicherheitsbezogenen Systeme*, *Realisierung der sicherheitsbezogenen Systeme anderer Technologie*, *Realisierung der externen Einrichtungen zur Risikominderung*, *Gesamtinstallation und Gesamtinbetriebnahme*, *Sicherheits-Gesamtvalidierung*, *Gesamtbetrieb*, *gesamte Instandhaltung und Reparatur*, *Gesamtmodifikation und gesamte Nachrüstung* sowie *Außerbetriebnahme oder Ausmusterung*.

Der zweite Teil der Norm „Teil 2: Anforderungen an sicherheitsbezogene elektrische/elektronische/programmierbar elektronische Systeme“ formuliert Anforderungen an die Hardware sicherheitsbezogener Systeme. Dabei werden je nach gefordertem Sicherheitsintegritätslevel unterschiedliche Maßnahmen vorgeschrieben. Eine Kenngröße ist neben der Fehlerwahrscheinlichkeit „Probability of Failure on Demand“ (kurz PFD) auch die sogenannte „Safe Failure Fraction“ (kurz SFF), das sehr interpretationsbedürftige und oft falsch angewandte Verhältnis unkritischer Fehler zu den Gesamtfehlern einer Systemkomponente. In „Teil 3: Anforderungen an Software“ werden Software-spezifische Randbedingungen und Maßnahmen dargestellt. Begriffsdefinitionen und Erläuterungen zum Grundverständnis sind Inhalt von „Teil 4: Begriffe und Abkürzungen“. In „Teil 5: Beispiele zur Ermittlung der Stufe der Sicherheitsintegrität“ werden unterschiedliche Verfahren zur Ermittlung eines Sicherheitsintegritätslevels dargestellt. Hier werden auch die Grundlagen der Funktionalen Sicherheit im Verständnis der Norm erläutert. „Teil 6: Anwendungsrichtlinien für IEC 61508-2 und IEC 61508-3“ gibt zusätzliche, teilweise beispielhafte Hinweise zu Teil 2 und Teil 3, „Teil 7: Anwendungshinweise über Verfahren und Maßnahmen“ erläutert die in den anderen Teilen empfohlenen bzw. vorgeschriebenen Verfahren und Maßnahmen.

C.1. Der Sicherheitslebenszyklus von DIN EN 61508

Im Folgenden werden die 16 Schritte des Sicherheitslebenszyklus kurz beschrieben. Der Inhalt ist Teil 1 von [DIN02] entnommen, für nähere Erläuterungen siehe dort.

C.1.1. Konzept

Ziel dieses Abschnitts ist es, einen Grad an Verständnis der EUC und deren Umgebung (physikalische, gesetzgebende usw.) zu entwickeln, der ausreichend ist, um

die anderen Tätigkeiten des Sicherheitslebenszyklus zufriedenstellend ausführen zu können.

Dazu gehört die Erarbeitung gründlicher Kenntnisse der EUC, ihrer erforderlichen Leit- oder Steuerungsfunktionen und ihrer physikalischen Umgebung. Außerdem müssen mögliche Gefährungsquellen bestimmt und Informationen über die ermittelten Gefährdungen (Giftigkeit, Explosivität, Korrosivität, Reaktionsfreudigkeit, Brennbarkeit usw.) sowie über die aktuellen Sicherheitsvorschriften (national und international) eingeholt werden. Die Gefährdungen durch Wechselwirkung mit anderen EUCs (installiert oder noch zu installieren) in der Nähe der EUC werden betrachtet und schlussendlich alle gewonnenen Informationen dokumentiert.

C.1.2. Definition des gesamten Anwendungsbereichs

Ziel dieses Abschnitts ist es, die Grenze der EUC und des EUC-Leit- oder Steuerungssystems festzulegen. Außerdem ist hier der Anwendungsbereich der Gefährdungs- und Risikoanalyse (zum Beispiel Prozessgefährdungen, Umweltgefährdungen usw.) festzulegen.

Dazu müssen die physikalischen Einrichtungen, einschließlich der EUC und des EUC-Leit- oder Steuerungssystems, die in den Anwendungsbereich der Gefährdungs- und Risikoanalyse eingeschlossen werden, die externen Ereignisse, die in der Gefährdungs- und Risikoanalyse berücksichtigt werden müssen, die Untersysteme, die mit den Gefährdungen in Verbindung stehen, sowie die Art von Unfall auslösenden Ereignissen, die betrachtet werden müssen (zum Beispiel Bauteilausfälle, Verfahrensfehler, menschliches Versagen, abhängige Ausfallmechanismen, die das Auftreten von Unfallketten verursachen können), festgelegt werden.

C.1.3. Gefährdungs- und Risikoanalyse

Das Ziel dieses Abschnitts ist, die Gefährdungen und gefährlichen Vorfälle der EUC und des EUC-Leit- oder Steuerungssystems (in allen Betriebsarten) für alle vernünftigerweise vorhersehbaren Umstände, einschließlich Fehlerbedingungen und Fehlanwendung, zu bestimmen. Zudem werden die Abläufe von Ereignissen bestimmt, die zu den bestimmten gefährlichen Vorfällen führen. Weiterhin sollen die mit den bestimmten gefährlichen Vorfällen verbundenen EUC-Risiken ermittelt werden.

Es muss eine Gefährdungs- und Risikoanalyse durchgeführt werden, die die Informationen aus der Phase der Definition des gesamten Anwendungsbereiches berücksichtigen muss. Wenn zu späteren Zeitpunkten in den Phasen des gesamten

Sicherheitslebenszyklus, des E/E/PES-Sicherheitslebenszyklus oder des Software-Sicherheitslebenszyklus Entscheidungen getroffen werden, die die Grundlage ändern, auf der frühere Entscheidungen getroffen wurden, muss eine weitere Gefährdungs- und Risikoanalyse durchgeführt werden. Die Beseitigung von Gefährdungen muss in Erwägung gezogen werden.

Die Gefährdungen und gefährlichen Vorfälle der EUC und des EUC-Leit oder Steuerungssystems müssen unter allen vernünftigerweise vorhersehbaren Umständen (einschließlich Fehlerbedingungen und vernünftigerweise vorhersehbarer Fehleranwendung) festgelegt werden. Dieses muss alle relevanten menschlichen Faktoren einschließen und muss ungewöhnlichen und selten genutzten Betriebsarten der EUC besondere Aufmerksamkeit schenken.

Die Abläufe von Ereignissen, die zu den bestimmten gefährlichen Vorfällen führen, müssen festgelegt werden. Die Wahrscheinlichkeiten der gefährlichen Vorfälle für die festgelegten Bedingungen müssen bewertet werden. Die möglichen Auswirkungen, die mit den festgelegten gefährlichen Vorfällen verbunden sind, müssen bestimmt werden. Das EUC-Risiko muss für jeden festgelegten gefährlichen Vorfall bewertet oder abgeschätzt werden.

Die Anforderungen können durch die Anwendung von entweder qualitativen oder quantitativen Methoden der Gefährdungs- und Risikoanalyse erfüllt werden. Die Angemessenheit der Methoden und der Umfang in dem diese Methoden angewendet werden müssen, hängen von einer Anzahl von Faktoren ab. Diese schließen die speziellen Gefährdungen und deren Auswirkungen, den Anwendungsbereich und dort bewährte Praktiken, die Anforderungen zu gesetzlichen und Sicherheitsbestimmungen, das EUC-Risiko sowie die Verfügbarkeit korrekter Daten, auf denen die Gefährdungs- und Risikoanalyse basieren muss, ein.

Die Gefährdungs- und Risikoanalyse muss jeden ermittelten gefährlichen Vorfall und die Komponenten, die dazu beitrugen, die Auswirkungen und die Wahrscheinlichkeit der Abläufe von Ereignissen, mit denen jeder gefährliche Vorfall verbunden ist, die notwendige Risikominderung für jeden gefährlichen Vorfall, die ergriffenen Maßnahmen zur Reduzierung oder Beseitigung von Gefährdungen und Risiken, die Annahmen, die während der Analyse der Risiken getroffen wurden, einschließlich geschätzter Anforderungsraten und Ausfallraten der Einrichtungen (jedes Zugeständnis, das für betriebliche Beschränkungen oder menschliches Eingreifen gewährt wird, muss genau beschrieben werden) und die Hinweise auf Schlüsselinformationen, die sich auf die sicherheitsbezogenen Systeme in jeder Phase des E/E/PES-Sicherheitslebenszyklus beziehen (zum Beispiel Verifikations- und Validierungstätigkeiten) betrachten.

Die Informationen und Ergebnisse, die die Gefährdungs- und Risikoanalyse bilden, müssen dokumentiert und während des gesamten Sicherheitslebenszyklus, von der Phase der Gefährdungs- und Risikoanalyse bis hin zur Phase der Außerbetriebnahme oder Ausmusterung für die EUC und das EUC-Leit- oder Steuerungssystems gepflegt werden.

C.1.4. Gesamte Sicherheitsanforderungen

Ziel dieses Abschnitts ist es, die Spezifikation der gesamten Sicherheitsanforderungen für die sicherheitsbezogenen E/E/PE-Systeme, sicherheitsbezogene Systeme anderer Technologie und externen Einrichtungen zur Risikominderung, im Hinblick auf die Anforderungen zu den Sicherheitsfunktionen und den Anforderungen zur Sicherheitsintegrität zu entwickeln, um die erforderliche funktionale Sicherheit zu erreichen.

Die Sicherheitsfunktionen, die zur Sicherstellung der erforderlichen funktionalen Sicherheit für jede ermittelte Gefährdung notwendig sind, müssen festgelegt werden. Diese Spezifikation muss die Spezifikation der gesamten Anforderungen zu den Sicherheitsfunktionen bilden. Die notwendige Risikominderung muss für jeden festgelegten gefährlichen Vorfall bestimmt werden. Die notwendige Risikominderung kann in quantitativer und/oder qualitativer Art und Weise bestimmt werden.

In Situationen, in denen eine anwendungsspezifische Internationale Norm vorhanden ist, die angemessene Methoden für die direkte Bestimmung der notwendigen Risikominderung enthält, können solche Normen verwendet werden, um die Anforderungen dieses Unterabschnitts zu erfüllen.

In Fällen, in denen Ausfälle des EUC-Leit oder Steuerungssystems eine Anforderung an ein oder mehrere sicherheitsbezogene E/E/PE-Systeme oder sicherheitsbezogene Systeme anderer Technologie und/oder externe Einrichtungen zur Risikominderung stellen und es nicht beabsichtigt ist, das EUC-Leit oder Steuerungssystem als sicherheitsbezogenes System zu bezeichnen, müssen die folgenden Anforderungen zutreffen:

1. Die Rate gefahrbringender Ausfälle, die für das EUC-Leit oder Steuerungssystem in Anspruch genommen wird, muss sich auf Daten stützen, die durch aktuelle Betriebserfahrung zum EUC-Leit oder Steuerungssystem in einer gleichartigen Anwendung, eine Zuverlässigkeitsanalyse, die nach einem anerkannten Verfahren ausgeführt wurde, oder eine industrielle Datenbank zur Zuverlässigkeit von allgemeinen Einrichtungen erworben wurden.

2. Die Rate gefahrbringender Ausfälle, die für das EUC-Leit oder Steuerungssystem in Anspruch genommen werden kann, darf nicht kleiner als 10^{-5} gefahrbringende Ausfälle pro Stunde sein.

3. Alle vernünftigerweise vorhersehbaren gefahrbringenden Ausfallarten des EUC-Leit oder Steuerungssystems müssen bestimmt werden und bei der Entwicklung der Spezifikation der gesamten Sicherheitsanforderungen in Betracht gezogen werden.

4. Das EUC-Leit oder Steuerungssystem muss getrennt und unabhängig von den sicherheitsbezogenen E/E/PE-Systemen, sicherheitsbezogenen Systemen anderer Technologie und externen Einrichtungen zur Risikominderung sein.

Falls diese Anforderungen nicht erfüllt werden können, muss das EUC-Leit oder Steuerungssystem als sicherheitsbezogenes System bezeichnet werden. Der dem EUC-Leit oder Steuerungssystem zugeordnete Sicherheits-Integritätslevel muss auf der Ausfallrate basieren, die für das EUC-Leit- oder Steuerungssystem in Übereinstimmung mit den in den Tabelle C.1 auf der nächsten Seite festgelegten Ausfallgrenzwerten angegeben wird. In solchen Fällen müssen die Anforderungen dieser Norm entsprechend dem zugeordneten Sicherheits-Integritätslevel auf das EUC-Leit oder Steuerungssystem angewendet werden.

Die Anforderungen zur Sicherheitsintegrität müssen in Hinsicht auf die notwendige Risikominderung für jede Sicherheitsfunktion festgelegt werden. Dies muss die Spezifikation der gesamten Anforderungen zur Sicherheitsintegrität ergeben.

Die Spezifikation der Anforderungen zu den Sicherheitsfunktionen und die Spezifikation der Anforderungen zur Sicherheitsintegrität müssen zusammen die Spezifikation der gesamten Sicherheitsanforderungen ergeben.

C.1.5. Zuordnung der Sicherheitsanforderungen

Ziel dieses Abschnitts ist es, die in der Spezifikation der gesamten Sicherheitsanforderungen (sowohl die Anforderungen zu den Sicherheitsfunktionen, als auch die Anforderungen zur Sicherheitsintegrität) enthaltenen Sicherheitsfunktionen den vorgesehenen sicherheitsbezogenen E/E/PE-Systemen, sicherheitsbezogenen Systemen anderer Technologie und externen Einrichtungen zur Risikominderung zuzuordnen. Zudem ist jeder Sicherheitsfunktion ein Sicherheitsintegritätslevel zuzuordnen.

Die vorgesehenen sicherheitsbezogenen Systeme, die zum Erreichen der erforderlichen funktionalen Sicherheit eingesetzt werden müssen, müssen festgelegt werden. Die notwendige Risikominderung kann erreicht werden durch externe Einrichtungen zur Risikominderung, sicherheitsbezogene E/E/PE-Systeme oder durch sicherheitsbezogene Systeme anderer Technologie.

Im Rahmen der Zuordnung der Sicherheitsfunktionen zu den vorgesehenen sicherheitsbezogenen E/E/PE-Systemen, sicherheitsbezogenen Systemen anderer Technologie und externen Einrichtungen zur Risikominderung müssen die vorhandenen Fertigkeiten und Ressourcen in allen Phasen des gesamten Sicherheitslebenszyklus betrachtet werden.

Jede Sicherheitsfunktion mit ihrer zugehörigen Anforderung zur Sicherheitsintegrität muss den vorgesehenen sicherheitsbezogenen E/E/PE-Systemen zugeordnet werden. Dabei muss die Risikominderung, die durch die sicherheitsbezogenen Systeme anderer Technologie und externen Einrichtungen zur Risikominderung erzielt wird, berücksichtigt werden, so dass die notwendige Risikominderung für diese Sicherheitsfunktion erreicht wird. Diese Zuordnung ist iterativ und wenn sich herausstellt, dass die notwendige Risikominderung nicht erreicht werden kann, muss die Architektur geändert und die Zuordnung wiederholt werden. Die Zuordnung muss so erfolgen, dass alle Sicherheitsfunktionen zugeordnet werden und die Anforderungen zur Sicherheitsintegrität für alle Sicherheitsfunktionen erreicht werden.

Die Anforderungen zur Sicherheitsintegrität jeder Sicherheitsfunktion müssen geeignet sein, anzuzeigen, ob der Zielparameter der Sicherheitsintegrität entweder die mittlere Wahrscheinlichkeit eines Ausfalls der entworfenen Funktion bei Anforderung (für eine Betriebsart mit niedriger Anforderungsrate), oder die Wahrscheinlichkeit eines gefahrbringenden Ausfalls pro Stunde (für eine Betriebsart mit hoher Anforderungsrate oder kontinuierlicher Anforderung) ist. Die Zuordnung der Anforderungen zur Sicherheitsintegrität muss unter Verwendung angemessener Methoden zur Kombination von Wahrscheinlichkeiten ausgeführt werden.

Die Zuordnung muss unter Berücksichtigung der Möglichkeit von Ausfällen infolge gemeinsamer Ursache fortgeführt werden. Wenn die sicherheitsbezogenen E/E/PE-Systeme, die sicherheitsbezogenen Systeme anderer Technologie und die externen Einrichtungen zur Risikominderung für die Zuordnung als unabhängig behandelt werden müssen,

- dann müssen sie funktional verschiedenartig sein (d. h. Verwendung vollständig verschiedener Lösungswege, um die gleichen Ergebnisse zu erzielen);

- dann müssen sie auf verschiedenartigen Technologien basieren (d. h. Verwendung unterschiedlicher Typen von Einrichtungen, um die gleichen Ergebnisse zu erzielen);

- dann dürfen sie keine gemeinsamen Teile, Dienste oder Versorgungssysteme (zum Beispiel Stromversorgungen) verwenden, deren Ausfall zu einem gefahrbringenden Ausfallmodus aller Systeme führen kann;

- dann dürfen keine gemeinsamen Betriebs-, Instandhaltungs- oder Prüfverfahren verwendet werden;

- dann müssen sie physikalisch getrennt sein, so dass voraussehbare Ausfälle nicht die redundanten sicherheitsbezogenen Systeme und externen Einrichtungen zur Risikominderung beeinflussen.

Können nicht alle diese Anforderungen erfüllt werden, dürfen die sicherheitsbezogenen E/E/PE-Systeme, die sicherheitsbezogenen Systeme anderer Technologie und die externen Einrichtungen zur Risikominderung nicht als unabhängig in Bezug auf die Zuordnung der Sicherheitsintegrität behandelt werden, es sei denn, es ist eine Analyse durchgeführt worden, die zeigt, dass sie ausreichend unabhängig voneinander sind (unter dem Gesichtspunkt der Sicherheitsintegrität).

Wenn die Zuordnung ausreichend ausgearbeitet worden ist, müssen die Anforderungen zur Sicherheitsintegrität für jede dem (den) sicherheitsbezogenen E/E/PE-System(en) zugeordnete Sicherheitsfunktion im Hinblick auf die Sicherheits-Integritätslevel in Übereinstimmung mit Tabelle C.1 auf Seite 213 festgelegt werden. Sie müssen geeignet sein anzuzeigen, ob der Zielparameter der Sicherheitsintegrität entweder die mittlere Wahrscheinlichkeit eines Ausfalls der entworfenen Funktion bei Anforderung (für eine Betriebsart mit niedriger Anforderungsrate), oder die Wahrscheinlichkeit eines gefahrbringenden Ausfalls pro Stunde (für eine Betriebsart mit hoher Anforderungsrate oder kontinuierlicher Anforderung) ist.

Für ein sicherheitsbezogenes E/E/PE-System, das Sicherheitsfunktionen verschiedener Sicherheits-Integritätslevel enthält, müssen diejenigen Teile der sicherheitsbezogenen Hardware und Software, für die unzureichende Unabhängigkeit in der Realisierung besteht, als zu der Sicherheitsfunktion mit dem höchsten Sicherheitsintegritätslevel zugehörig behandelt werden, es sei denn, es kann gezeigt werden, dass eine ausreichende Unabhängigkeit in der Realisierung zwischen diesen bestimmten Sicherheitsfunktionen besteht. Daher müssen die Anforderungen, die

Tabelle C.1.: Ausfallgrenzwerte für Sicherheitsfunktionen

SIL	Betriebsart mit niedriger Anforderungsrate	Betriebsart mit hoher Anforderungsrate bzw. kontinuierlicher Anforderung
4	$\geq 10^{-5}$ bis $< 10^{-4}$	$\geq 10^{-9}/h$ bis $< 10^{-8}/h$
3	$\geq 10^{-4}$ bis $< 10^{-3}$	$\geq 10^{-8}/h$ bis $< 10^{-7}/h$
2	$\geq 10^{-3}$ bis $< 10^{-2}$	$\geq 10^{-7}/h$ bis $< 10^{-6}/h$
1	$\geq 10^{-2}$ bis $< 10^{-1}$	$\geq 10^{-6}/h$ bis $< 10^{-5}/h$

für den höchsten relevanten Sicherheitsintegritätslevel zutreffen, auf alle diejenigen Teile angewendet werden.

Eine Architektur, die nur aus einem einzigen sicherheitsbezogenen E/E/PE-System des Sicherheitsintegritätslevels 4 besteht, ist nur erlaubt, wenn von den unten stehenden Kriterien entweder 1. oder 2. und 3. gemeinsam erfüllt werden:

1. Der Ausfallgrenzwert der Sicherheitsintegrität wurde durch eine Kombination von angemessen analytischen Methoden und Tests eindeutig bewiesen.

2. Es liegen weitläufige Betriebserfahrungen bezüglich der als Teile für das sicherheitsbezogene E/E/PE-System verwendeten Komponenten vor. Diese Erfahrungen müssen in einer vergleichbaren Umgebung gewonnen und mindestens in einem System mit vergleichbarem Komplexitätsgrad angewendet worden sein.

3. Es liegen ausreichende Hardwareausfalldaten bezüglich der als Teile für das sicherheitsbezogene E/E/PE-System verwendeten Komponenten vor, um ausreichendes Vertrauen in Bezug auf den zu erreichenden Ausfallgrenzwert der Sicherheitsintegrität der Hardware zu erreichen. Die Daten sollten bezüglich der vorgesehenen Umgebung, der Anwendung und des Komplexitätsgrads zutreffend sein.

Keinem einzelnen sicherheitsbezogenen E/E/PESystem darf ein kleinerer als der in den Tabelle C.1 angegebene Ausfallgrenzwert der Sicherheitsintegrität zugeordnet werden. Für sicherheitsbezogene Systeme, die in einer Betriebsart mit niedriger Anforderungsrate betrieben werden, ist die untere Grenze eines Ausfalls der entworfenen Funktion bei Anforderung auf eine mittlere Wahrscheinlichkeit von 10^{-5} festgelegt. Für eine Betriebsart mit hoher Anforderungsrate oder kontinuierlicher Anforderung, ist die untere Grenze der Wahrscheinlichkeit eines gefahrbringenden Ausfalls auf 10^{-9} pro Stunde festgelegt.

Die Informationen und Ergebnisse der Zuordnung der Sicherheitsanforderungen müssen zusammen mit allen Annahmen und Begründungen dokumentiert werden.

C.1.6. Planung des Gesamtbetriebs und der gesamten Instandhaltung

Ziel dieses Abschnitts ist es, einen Plan für den Betrieb und die Instandhaltung der sicherheitsbezogenen E/E/PESysteme zu entwickeln, um sicherzustellen, dass die erforderliche funktionale Sicherheit während des Betriebs und der Instandhaltung aufrechterhalten wird.

Es muss ein Plan aufgestellt werden, der Folgendes festlegen muss: Die Handlungen, die routinemäßig durchgeführt werden müssen, um die funktionale Sicherheit der sicherheitsbezogenen E/E/PE-Systeme aufrechtzuerhalten; die Handlungen und Einschränkungen, die notwendig sind, um einen unsicheren Zustand zu verhindern (zum Beispiel während des Einschaltens, des normalen Betriebs, der Routinetests, vorhersehbarer Störungen, bei Fehlern und während des Abschaltens), die Anforderungen an die sicherheitsbezogenen E/E/PE-Systeme zu reduzieren oder die Auswirkungen der gefährlichen Vorfälle zu mindern; die Dokumentation, die über die Ergebnisse der Audits der funktionalen Sicherheit und der Tests geführt werden muss; die Dokumentation, die über gefährliche Vorfälle und alle Vorkommnisse, die zu einem gefahrbringenden Vorfall führen könnten, geführt werden muss; den Umfang der Instandhaltungstätigkeiten (soweit verschieden von den Modifikationstätigkeiten); die Maßnahmen, die im Falle einer Gefährdung getroffen werden müssen sowie die Inhalte der chronologischen Dokumentation der Betriebs- und Instandhaltungstätigkeiten.

Die routinemäßigen Instandhaltungstätigkeiten, die durchgeführt werden müssen, um unerkannte Ausfälle zu erkennen, müssen durch eine systematische Analyse bestimmt werden.

Der Plan für die Instandhaltung der sicherheitsbezogenen E/E/PE-Systeme muss mit denjenigen abgestimmt werden, die für den zukünftigen Betrieb und die Instandhaltung der sicherheitsbezogenen E/E/PE-Systeme, der sicherheitsbezogenen Systeme anderer Technologie, der externen Einrichtungen zur Risikominderung und der nichtsicherheitsbezogenen Systeme, die die Möglichkeit haben, Anforderungen an die sicherheitsbezogenen Systeme zu stellen, verantwortlich sind.

C.1.7. Planung der Sicherheits-Gesamtvalidierung

Ziel dieses Abschnitts ist es, einen Plan zu entwickeln, um die Sicherheits-Gesamtvalidierung der sicherheitsbezogenen E/E/PE-Systeme zu erleichtern.

Es muss ein Plan entwickelt werden, der Folgendes enthalten muss: Einzelheiten, wann die Validierung stattfinden muss; Einzelheiten zu denjenigen Personen, die die Validierung ausführen müssen; Spezifikation der relevanten Betriebsarten der EUC mit deren Beziehung zum sicherheitsbezogenen E/E/PE-System (einschließlich, wenn zutreffend, Vorbereitung für die Verwendung, inklusive Einstellung und Kalibrierung, Anlauf, Lernmodus, Automatikbetrieb, manueller Betrieb, halbautomatischer Betrieb, Dauerbetrieb, Rücksetzen, Abschaltung, Instandhaltung, unter normalen Umständen vorhersehbare, nicht bestimmungsgemäße Zustände); Festlegung der sicherheitsbezogenen E/E/PE-Systeme, die für jede einzelne Betriebsart der EUC validiert werden müssen, bevor die Inbetriebnahme beginnt; die technische Strategie für die Validierung (zum Beispiel analytische Methoden, statistische Tests usw.); die Maßnahmen, Methoden und Verfahren, die zur Bestätigung, dass die Zuordnung der Sicherheitsfunktionen korrekt ausgeführt wurde, verwendet werden müssen (dies muss die Bestätigung einschließen, dass jede Sicherheitsfunktion mit der Spezifikation der gesamten Anforderungen zu den Sicherheitsfunktionen und mit der Spezifikation der gesamten Anforderungen zur Sicherheitsintegrität übereinstimmt); die erforderlichen Umgebungsbedingungen, unter denen die Validierungstätigkeiten stattfinden müssen (für Tests umfasst dieses zum Beispiel kalibrierte Werkzeuge und Einrichtungen); die Kriterien für Bestehen und Nichtbestehen sowie die Vorgehensweise und Verfahren zur Auswertung der Ergebnisse der Validierung, besonders von Ausfällen.

Die Informationen müssen dokumentiert werden und müssen den Plan für die Sicherheits-Gesamtvalidierung der sicherheitsbezogenen E/E/PE-Systeme bilden.

C.1.8. Planung der Gesamtinstallation und Gesamtinbetriebnahme

Ziel dieses Abschnitts ist es, einen Plan für die kontrollierte Installation der sicherheitsbezogenen E/E/PE-Systeme zu entwickeln, um sicherzustellen, dass die erforderliche funktionale Sicherheit erreicht ist. Zudem ist ein Plan für die kontrollierte Inbetriebnahme der sicherheitsbezogenen E/E/PE-Systeme zu entwickeln, um sicherzustellen, dass die erforderliche funktionale Sicherheit erreicht ist.

Es muss ein Plan für die Installation der sicherheitsbezogenen E/E/PE-Systeme entwickelt werden, der Folgendes festlegt: Den Zeitplan für die Installation;

diejenigen Personen, die für die verschiedenen Teile der Installation verantwortlich sind; die Verfahren für die Installation; die Reihenfolge, in der die einzelnen Elemente integriert werden; die Kriterien zur Festsetzung, wann die gesamten sicherheitsbezogenen E/E/PE-Systeme oder Teile für die Installation fertig gestellt und wann Installationstätigkeiten abgeschlossen sind sowie Verfahren für die Behebung von Ausfällen und Inkompatibilitäten.

Außerdem muss ein Plan für die Inbetriebnahme der sicherheitsbezogenen E/E/PE-Systeme entwickelt werden, der Folgendes festlegt: Den Zeitplan für die Inbetriebnahme; diejenigen Personen, die für die verschiedenen Teile der Inbetriebnahme verantwortlich sind; die Verfahren für die Inbetriebnahme; die Beziehungen zu den verschiedenen Stufen der Installation sowie die Beziehungen zur Validierung.

Die Planung der Gesamtinstallation und Gesamtinbetriebnahme muss dokumentiert werden.

C.1.9. Realisierung der sicherheitsbezogenen E/E/PE-Systeme

Ziel dieses Abschnitts ist es, sicherheitsbezogene E/E/PE-Systeme zu erstellen, die der Spezifikation der E/E/PES-Sicherheitsanforderungen (umfasst die Spezifikation der E/E/PES-Anforderungen zu den Sicherheitsfunktionen und die Spezifikation der E/E/PES-Anforderungen zur Sicherheitsintegrität) entsprechen.

DIN EN 61508 enthält in Teil 2 und 3 detaillierte Anweisungen für die Entwicklung von Hardware und Software sicherheitsbezogener E/E/PE-Systeme.

C.1.10. Realisierung der sicherheitsbezogenen Systeme anderer Technologie

Ziel dieses Abschnitts ist es, sicherheitsbezogene Systeme anderer Technologie zu erstellen, um die Anforderungen zu den Sicherheitsfunktionen und Anforderungen zur Sicherheitsintegrität, die für solche Systeme festgelegt sind, zu erfüllen.

Die Spezifikation, um die Anforderungen zu den Sicherheitsfunktionen und die Anforderungen zur Sicherheitsintegrität für sicherheitsbezogene Systeme anderer Technologie zu erfüllen, ist nicht Gegenstand der Norm.

C.1.11. Realisierung externer Einrichtungen zur Risikominderung

Ziel dieses Abschnitts ist es, externe Einrichtungen zur Risikominderung zu erstellen, um die Anforderungen zu den Sicherheitsfunktionen und die Anforderungen zur Sicherheitsintegrität, die für solche Einrichtungen festgelegt sind, zu erfüllen.

Die Spezifikation, um die Anforderungen zu den Sicherheitsfunktionen und die Anforderungen zur Sicherheitsintegrität für die externen Einrichtungen zur Risikominderung zu erfüllen, ist nicht Gegenstand der Norm.

C.1.12. Gesamtinstallation und Gesamtinbetriebnahme

Ziel dieses Abschnitts ist es, die sicherheitsbezogenen E/E/PE-Systeme zu installieren und in Betrieb zu nehmen.

Installationstätigkeiten müssen in Übereinstimmung mit dem Plan für die Installation der sicherheitsbezogenen E/E/PE-Systeme ausgeführt werden. Die während der Installation dokumentierten Informationen müssen die Dokumentation der Installationstätigkeiten sowie die Behebung von Ausfällen und Inkompatibilitäten einschließen.

Inbetriebnahmetätigkeiten müssen in Übereinstimmung mit dem Plan für die Inbetriebnahme der sicherheitsbezogenen E/E/PE-Systeme ausgeführt werden. Die während der Inbetriebnahme dokumentierten Informationen müssen die Dokumentation der Inbetriebnahmetätigkeiten, Hinweise auf Berichte zu Ausfällen sowie die Behebung von Ausfällen und Inkompatibilitäten einschließen.

C.1.13. Sicherheits-Gesamtvalidierung

Ziel dieses Abschnitts ist es zu validieren, dass die sicherheitsbezogenen E/E/PE-Systeme die Spezifikation der gesamten Sicherheitsanforderungen im Hinblick auf die gesamten Anforderungen zu den Sicherheitsfunktionen und die gesamten Anforderungen zur Sicherheitsintegrität, unter Berücksichtigung der Zuordnung der Sicherheitsanforderungen für die entwickelten sicherheitsbezogenen E/E/PE-Systeme, erfüllen.

Validierungstätigkeiten müssen in Übereinstimmung mit dem Plan der Sicherheits-Gesamtvalidierung für die sicherheitsbezogenen E/E/PE-Systeme ausgeführt werden. Alle für quantitative Messungen im Rahmen der Validierungstätigkeiten verwendeten Betriebsmittel müssen nach einer Spezifikation, die auf eine nationale Norm rückführbar ist, oder der Spezifikation des Herstellers kalibriert sein.

Die während der Validierung dokumentierten Informationen müssen die Dokumentation der Validierungstätigkeiten in chronologischer Form, die verwendete Version der Spezifikation der gesamten Sicherheitsanforderungen, die (durch Tests oder Analysen) validierte Sicherheitsfunktion, die verwendeten Werkzeuge und Ausrüstung, zusammen mit den Kalibrierungsdaten, die Ergebnisse der Validierungstätigkeiten, die Identifikation der Konfiguration des geprüften Gegenstands, die angewendeten Verfahren und die Prüfumgebung sowie Unstimmigkeiten zwischen erwarteten und tatsächlichen Ergebnissen einschließen.

Wenn Unstimmigkeiten zwischen den erwarteten und tatsächlichen Ergebnissen auftreten, müssen die durchgeführten Analysen und die getroffenen Entscheidungen, ob die Validierung fortgesetzt oder eine Modifikationsanforderung gestellt und die Rückkehr an einem früheren Punkt der Validierung erfolgt, dokumentiert werden.

C.1.14. Gesamtbetrieb, gesamte Instandhaltung und Reparatur

Ziel dieses Abschnitts ist es, die sicherheitsbezogenen E/E/PE-Systeme zu betreiben, zu warten und zu reparieren, damit die erforderliche funktionale Sicherheit aufrechterhalten wird.

Der Plan für die Instandhaltung der sicherheitsbezogenen E/E/PE-Systeme, die Betriebs-, Instandhaltungs und Reparaturverfahren für die sicherheitsbezogenen E/E/PE-Systeme und die Betriebs- und Pflegeverfahren für Software müssen ausgeführt werden.

Die Ausführung dieser Punkte muss die Ausführung von Verfahren, die Beachtung von Zeitplänen für die Instandhaltung, die Pflege der Dokumentation, die periodische Durchführung von Audits der funktionalen Sicherheit sowie die Aufzeichnung von Modifikationen, die an den sicherheitsbezogenen E/E/PE-Systemen durchgeführt worden sind, einschließen.

Es muss eine chronologische Dokumentation des Betriebs, der Reparatur und der Instandhaltung der sicherheitsbezogenen E/E/PE-Systeme geführt werden, die die Ergebnisse der Audits der funktionalen Sicherheit und der Tests, die Dokumentation des zeitlichen Auftretens und der Gründe für die Anforderungen an die sicherheitsbezogenen E/E/PE-Systeme (im tatsächlichen Betrieb) zusammen mit der Leistungsfähigkeit der sicherheitsbezogenen E/E/PE-Systeme, wenn sie diesen Anforderungen unterworfen sind, und die während der routinemäßigen Instandhaltung gefundenen Fehler sowie die Dokumentation der Modifikationen, die an der EUC, an dem EUC-Leit- oder Steuerungssystem und an den sicherheitsbezogenen E/E/PE-Systemen ausgeführt worden sind, enthalten muss.

Die genauen Anforderungen für eine chronologische Dokumentation hängen von der speziellen Anwendung ab und müssen, wenn zutreffend, in anwendungsspezifischen Normen ausführlich beschrieben werden.

C.1.15. Gesamtmodifikation und gesamte Nachrüstung

Ziel dieses Abschnitts ist es sicherzustellen, dass die funktionale Sicherheit für die sicherheitsbezogenen E/E/PE-Systeme, sowohl während als auch nach der Ausführung der Modifikations- und Nachrüstungsphase, angemessen ist.

Vor der Ausführung jeder Modifikations- und Nachrüstungstätigkeit müssen die Verfahren geplant werden.

Die Modifikations- und Nachrüstungsphase darf nur durch die Stellung einer autorisierten Anforderung unter Anwendung der Verfahren für das Management der funktionalen Sicherheit eingeleitet werden. Die Anforderung muss die festgestellten Gefährdungen, die betroffen sein könnten, die vorgesehene Änderung (sowohl Hardware als auch Software) sowie die Gründe für die Änderung genau beschreiben.

Es muss eine Einflussanalyse durchgeführt werden, die eine Beurteilung der Wirkung der vorgeschlagenen Modifikations- und Nachrüstungstätigkeit auf die funktionale Sicherheit jedes sicherheitsbezogenen E/E/PE =Systems beinhalten muss. Die Beurteilung muss eine Gefährdungs- und Risikoanalyse einschließen, die ausreichend ist, um das Maß und die Tiefe festzulegen, mit der nachfolgende Phasen des gesamten Sicherheitslebenszyklus, des E/E/PES-Sicherheitslebenszyklus oder des Software-Sicherheitslebenszyklus ausgeführt werden müssen. Die Beurteilung muss auch die Einflüsse anderer gleichzeitiger Modifikations- und Nachrüstungstätigkeiten berücksichtigen und muss auch die funktionale Sicherheit sowohl während, als auch nach den stattgefundenen Modifikations- und Nachrüstungstätigkeiten betrachten. Die beschriebenen Ergebnisse müssen dokumentiert werden. Die Erlaubnis zur Ausführung der erforderlichen Modifikations- und Nachrüstungstätigkeit muss von den Ergebnissen der Einflussanalyse abhängig sein.

Alle Modifikationen, die einen Einfluss auf die funktionale Sicherheit eines beliebigen sicherheitsbezogenen E/E/PE-Systems haben, müssen eine Rückkehr zu einer angemessenen Phase des gesamten Sicherheitslebenszyklus, des E/E/PES-Sicherheitslebenszyklus oder des Software-Sicherheitslebenszyklus einleiten. Alle nachfolgenden Phasen müssen dann in Übereinstimmung mit den festgelegten Verfahren für die speziellen Phasen und in Übereinstimmung mit den Anforderungen dieser Norm durchgeführt werden.

Es muss eine chronologische Dokumentation eingerichtet und geführt werden, die alle Modifikationen und Nachrüstungen genau beschreibt und Hinweise auf die Modifikations- oder Nachrüstungsanforderung, die Einflussanalyse, Neuverifikation und Neuvalidierung von Daten und Ergebnissen sowie alle Dokumente, die von der Modifikations- und Nachrüstungstätigkeit beeinflusst werden enthält.

C.1.16. Außerbetriebnahme oder Ausmusterung

Ziel dieses Abschnitts ist es sicherzustellen, dass die funktionale Sicherheit für die sicherheitsbezogenen E/E/PE-Systeme unter den Umständen während und nach den Tätigkeiten der Außerbetriebnahme oder Ausmusterung der EUC angemessen ist.

Vor jeder Außerbetriebnahme- oder Ausmusterungstätigkeit muss eine Einflussanalyse ausgeführt werden, die eine Beurteilung der Einflüsse der vorgeschlagenen Außerbetriebnahme- oder Ausmusterungstätigkeit auf die funktionale Sicherheit jedes sicherheitsbezogenen E/E/PE-Systems, das mit der EUC verbunden ist, einschließen muss. Die Einflussanalyse muss auch benachbarte EUCs und die Einflüsse auf deren sicherheitsbezogene E/E/PE-Systeme betrachten. Die Beurteilung muss eine Gefährdungs- und Risikoanalyse einschließen, die ausreichend ist, um das Maß und die Tiefe nachfolgender Phasen des gesamten Sicherheitslebenszyklus, des E/E/PES-Sicherheitslebenszyklus oder des Software-Sicherheitslebenszyklus festzulegen. Die beschriebenen Ergebnisse müssen dokumentiert werden.

Die Außerbetriebnahme- oder Ausmusterungsphase darf nur durch die Stellung einer autorisierten Anforderung unter Anwendung der Verfahren für das Management der funktionalen Sicherheit eingeleitet werden. Die Erlaubnis zur Durchführung der erforderlichen Außerbetriebnahme oder Ausmusterung muss von den Ergebnissen der Einflussanalyse abhängig sein.

Bevor die Außerbetriebnahme oder Ausmusterung stattfindet, muss ein Plan erstellt werden, der Verfahren in Bezug auf die Stilllegung der sicherheitsbezogenen E/E/PE-Systeme und die Demontage der sicherheitsbezogenen E/E/PE-Systeme enthält.

Falls irgendeine Außerbetriebnahme- oder Ausmusterungstätigkeit einen Einfluss auf die funktionale Sicherheit irgendeines sicherheitsbezogenen E/E/PE-Systems hat, muss dies eine Rückkehr zu einer angemessenen Phase des gesamten Sicherheitslebenszyklus, des E/E/PES-Sicherheitslebenszyklus oder des Software-Sicherheitslebenszyklus einleiten. Alle nachfolgenden Phasen müssen dann in Übereinstimmung mit den in dieser Norm angegebenen Verfahren für den

festgelegten Sicherheits-Integritätslevel der sicherheitsbezogenen E/E/PE-Systeme durchgeführt werden.

Es muss eine chronologische Dokumentation eingerichtet und geführt werden, die die Außerbetriebnahme- oder Ausmusterungstätigkeiten genau beschreibt und Hinweise auf den für die Außerbetriebnahme- oder Ausmusterungstätigkeiten verwendeten Plan und die Einflussanalyse enthält.

D. Glossar

Architektur (engl. architecture)

Architektur nennt man die spezifische Konfiguration von Hardware- und Softwareelementen in einem System (nach [DIN02]).

Ausfallrate (engl. failure rate)

Mit dem Begriff Ausfallrate bezeichnet man die mittlere Anzahl von Ausfällen in einer (kurzen) Zeiteinheit δt (nach [VDI85]), oder auch - anders ausgedrückt - die Wahrscheinlichkeit, dass eine Einheit, die bis zum Zeitpunkt t überlebt hat, im nächsten kleinen Zeitraum δt ausfallen wird (nach [MP03]).

Die Ausfallrate wird in der Regel mit dem griechischen Buchstaben λ abgekürzt.

Ausfall infolge gemeinsamer Ursache (engl. common cause failure)

Common Cause Failures sind Ausfälle, die aufgrund einer einzigen gemeinsamen Ursache entstehen. Zur Beherrschung sind Maßnahmen wie z. B. Diversität, räumliche Trennung oder elektrische Trennung notwendig (nach [Sch92]).

Diversität (engl. diversity)

Diversität nennt man die Verwendung ungleichartiger technischer Mittel (z. B. andere physikalische Prinzipien, andere Lösungswege der gleichen Aufgabe usw.) zur Erreichung nützlicher Redundanz (nach [VDI00]).

Fail-operational System

Ein System wird „fail-operational“ (oft FO abgekürzt) genannt, wenn im Fall eines Fehlers das System eine Grundfunktionalität solange aufrecht erhält, bis ein sicherer Systemzustand erreicht ist. Diese Systemeigenschaft wird verlangt, wenn im Fall eines Fehlers ein solcher sicherer Systemzustand nicht unmittelbar erreicht werden kann (nach [Kop97]).

Fail-safe System

Als „fail-safe“ bezeichnet man die Fähigkeit eines technischen Systems, beim Auftreten bestimmter Ausfälle im sicheren Zustand zu bleiben oder unmittelbar in einen (anderen) sicheren Zustand überzugehen (nach [VDI00]).

Fail-silent System

Ein System nennt man „fail-silent“ (kurz FS), wenn es sich nach Feststellen von Abweichungen vom vorgegeben Verlauf nach außen hin sofort still verhält, d. h. sich in einen für die Außenwelt als unkritisch angesehen Zustand versetzt. Dies kann beispielsweise durch spontanes Selbstabschalten der entsprechenden Komponenten oder des jeweiligen Teilsystems erfolgen.

Fehler (engl. fault)

Der Begriff Fehler im Sinne einer Fehlerursache (Fault) bezeichnet den Auslöser für einen Fehler im Sinne eines Fehlerzustands. Die Fehlerursache muss entweder toleriert oder vermieden werden (nach [Lap92]).

Fehler (engl. error)

Der Begriff Fehler im Sinne eines Fehlerzustands (Error) bezeichnet einen Teil des Systemzustands, der dafür verantwortlich ist, dass ein Ausfall auftritt. Die Fehlerursache im System wird offenbar (nach [Lap92]).

Fehler (engl. failure)

Der Begriff Fehler im Sinne eines Fehlerzustands (Error) bezeichnet die Abweichung der erbrachten Leistung von der in der Systemspezifikation geforderten Leistung.

Die Unterscheidung zwischen Fehlerursache (Fault), Fehlerzustand (Error) und Ausfall (Failure) ist vom betrachteten System abhängig. Ein Ausfall einer Komponente kann die Fehlerursache eines übergeordneten Systems sein (nach [Lap92]).

Fehler (engl. nonconformity, defect)

Nichterfüllung der Spezifikation, Inkorrektheit (nach [VDI00]).

Fehlerhypothese (engl. fault hypothesis)

Annahme über Art und Anzahl der auftretenden Fehler (nach [Kop88]).

Fehlermaskierung (engl. error masking)

Verhinderung der Auswirkung eines Fehlers in einem Subsystem, so dass die ununterbrochene Erfüllung der Funktion des Gesamtsystems ermöglicht wird (nach [VDI00]).

Fehlersemantik

Das Verhalten von Komponenten, Subsystemen oder Systemen im Fehlerfall wird als Fehlersemantik bezeichnet. Der Bedarf für den Einsatz eines Fehlertoleranzmechanismus stellt sich in der Regel durch eine Abweichung zwischen der tatsächlichen Fehlersemantik und der gewünschten Fehlersemantik an einer bestimmten Schnittstelle des Systems dar (nach [Cri91]).

Fehlertoleranz (engl. fault tolerance)

Unter Fehlertoleranz wird die Fähigkeit eines Systems verstanden, auch mit einer begrenzten Zahl fehlerhafter Subsysteme seine spezifizierte Funktion erfüllen zu können (nach [Gör89]).

Fehlervermeidung (engl. fault avoidance)

Fehlervermeidung nennt man die Verwendung von Techniken und Verfahren mit dem Ziel, die Entstehung von Fehlern während jeder Phase des Systemlebenszyklus zu vermeiden (nach [DIN02]).

Funktionale Sicherheit (engl. functional safety)

Funktionale Sicherheit ist der Teil der Gesamtsicherheit, der von der korrekten Funktion des sicherheitsbezogenen Systems abhängt (nach [DIN02]).

Gefahr (engl. danger)

Gefahr ist eine Sachlage, bei der das Risiko größer als das Grenzrisiko ist (nach [VDI00]).

Gefährdung (engl. hazard)

Eine Gefährdung ist eine potenzielle Schadensquelle (nach [DIN02]).

Grenzrisiko (engl. limiting risk)

Das Grenzrisiko ist das größte noch vertretbare Risiko eines bestimmten technischen Vorgangs oder Zustands. Im Allgemeinen lässt sich das Grenzrisiko nicht quantitativ erfassen. Es wird in der Regel durch sicherheitstechnische Festlegungen beschrieben (nach [DIN87]).

Typischerweise entsteht das Grenzrisiko aus einem sozialen Konsens. Im Falle, dass dieser sich ändert, ändert sich auch das Grenzrisiko.

Hardwareeinheit

Element der Erzeugnisstruktur, das ausschließlich aus Hardware besteht (nach [IAB02a]).

Hardwarekomponente

Hardwarebaustein einer Hardwareeinheit. Hardwarekomponenten können ihrerseits andere Hardwarekomponenten und Hardwaremodule enthalten (nach [IAB02a]).

Hardwaremodul

Hardwaremodule sind die kleinsten im V-Modell ’97 betrachteten Bausteine einer Hardwareeinheit (nach [IAB02a]).

Kritikalität

Die Kritikalität einer Einheit drückt aus, welche Bedeutung ihrem Fehlverhalten beigemessen wird. Die Kritikalität wird in Stufen angegeben, wobei die Einstufung umso höher ist, je gravierendere Auswirkungen bei Fehlverhalten zu erwarten sind (nach [IAB02a]).

momentane Verfügbarkeit

Die momentane Verfügbarkeit ist die Wahrscheinlichkeit, eine Einheit zu einem vorgegebenen Zeitpunkt der geforderten Anwendungsdauer unter vorgegebenen Arbeitsbedingungen in einem funktionsfähigen Zustand anzutreffen. (nach [DIN90]).

Nachweis

Information, deren Richtigkeit bewiesen werden kann und die auf Tatsachen beruht, welche durch Beobachtung, Messung, Untersuchung oder durch andere Ermittlungsverfahren gewonnen wird (nach [DIN92]).

Qualität

Qualität ist die Gesamtheit von Merkmalen einer Einheit bezüglich ihrer Eignung, festgelegte und vorausgesetzte Erfordernisse zu erfüllen (nach [DIN92]).

Qualitätssicherung

Qualitätssicherung umfasst alle geplanten systematischen Tätigkeiten, die innerhalb des Qualitätsmanagementsystems verwirklicht sind, und die wie erforderlich dargelegt werden, um angemessenes Vertrauen zu schaffen, das eine Einheit die Qualitätsforderung erfüllen wird (nach [DIN92]).

Redundanz (engl. redundancy)

Redundanz nennt man das funktionsbereite Vorhandensein von mehr als für die vorgesehene Funktion notwendigen technischen Mitteln (nach [VDI00]).

Restrisiko (engl. residual risk)

Restrisiko nennt man das trotz Schutzmaßnahmen verbleibende Risiko (nach [DIN02]).

Risiko (engl. risk)

Das Risiko, das mit einem bestimmten technischen Vorgang oder Zustand verbunden ist, wird zusammenfassend durch eine Wahrscheinlichkeitsaussage beschrieben, die

- die zu erwartende Wahrscheinlichkeit bzw. Häufigkeit des Eintritts eines zum Schaden führenden Ereignisses und
- das beim Ereigniseintritt zu erwartende Schadensausmaß

berücksichtigt.

Mathematisch gesehen ist das Risiko R ein sogenanntes Paar aus Ereignishäufigkeit H und Schadensausmaß S, kurz

$$R = (H, S).$$

Im Allgemeinen ist ein Risiko aber nicht quantitativ erfassbar, nur selten lässt es sich als Kombination der beiden Parameter Häufigkeit (H) und Schadensausmaß (S) quantifizieren. Daher sind zwei Risiken auch nicht direkt miteinander vergleichbar. Eine Ausnahme ist, wenn das Schadensausmaß S bei zwei Risiken identisch ist,

dann kann ein mathematischer Vergleich von H zum Ziel führen. Dies muss bei der Definition des Sicherheitsbegriffs beachtet werden, der auf einer Vergleichbarkeit von Risiken beruht.

Ist das Schadensausmaß S quantifizierbar (z. B. in finanzieller Form) und kann man im konkreten Fall von einem „mittleren Schaden“ bei Eintritt des Ereignisses sprechen, so kann für die Quantifizierung des Risikos das Produkt $R = H \cdot S$ verwendet werden.

Risikoanalyse (engl. risk analysis)

In der Risikoanalyse wird untersucht, wie wahrscheinlich es ist, dass eine der ermittelten Gefährdungen wirksam wird und wie hoch der Schaden ist, der dabei entsteht. Das Risiko wird aus der Eintrittswahrscheinlichkeit und der zu erwartenden Schadenshöhe ermittelt (nach [IAB02a]).

Dabei kommt in der Regel ein Risikograph zum Einsatz.

Risikominderung (engl. risk reduction)

Ist das aktuelle Risiko einer betrachteten Funktion oder eines betrachteten Systems größer als das Grenzrisiko, so muss das Risiko durch geeignete Maßnahmen bis auf mindestens das Grenzrisiko vermindert werden. Diese Risikominderung muss nicht zwingend mit der notwendigen Risikominderung, also der Differenz zwischen dem aktuellen Risiko und dem Grenzrisiko, identisch sein.

Schaden (engl. harm)

Ein Schaden ist die physische Verletzung oder Schädigung der Gesundheit von Menschen, entweder direkt oder indirekt als ein Ergebnis von Schäden von Gütern oder der Umwelt (nach [DIN02])

Schlafender Fehler

Schlafende Fehler in Komponenten sind Fehler, die im System vorhanden sind, aber zunächst nicht erkennbar sind. Beispielsweise kann eine Komponente beschädigt sein oder eine Teilfunktion ausfallen, ohne dass es sofort bemerkt wird (nach [Sch92]).

Sicherer Zustand (engl. safe state)

Zustand einer Betrachtungseinheit, bei der es aufgrund des festgestellten Nichtauftretens von sicherheitsrelevante Fehlfunktionen oder aufgrund der getroffenen

Schutzmaßnahmen gegen mögliche sicherheitsrelevante Fehlfunktionen das Risiko vertretbar gering ist (nach [VDI00]).

Sicherheit (engl. safety)

Sicherheit ist eine Sachlage, bei der das Risiko nicht größer als das Grenzrisiko ist (nach [DIN87]).

Sicherheit (engl. security)

Ein System als sicher im Sinne von englisch „security“ bezeichnet, wenn seine Umwelt nur einen streng kontrollierten und überwachten Zugriff auf das System hat. Damit ist also Sicherheit gegen etwas oder gegen jemanden gemeint, beispielsweise gegen ungewünschte Eindringlinge.

sicherheitsbezogenes System (engl. safety-related system)

Ein sicherheitsbezogenes System ist ein System, das zusammen mit externen Einrichtungen zur Risikominderung dazu vorgesehen ist, die notwendige Risikominderung auf das Restrisiko zu erreichen (nach [DIN02]).

Sicherheitsintegrität (engl. safety integrity)

Sicherheitsintegrität ist die Wahrscheinlichkeit, dass ein sicherheitsbezogenes System die geforderten Sicherheitsfunktionen unter allen festgelegten Bedingungen innerhalb eines festgelegten Zeitraumes anforderungsgemäß ausführt (nach [DIN02]).

Sicherheits-Integritätslevel (SIL) (engl. safety integrity level)

Als Sicherheitsintegritätslevel bezeichnet man eine von vier diskreten Stufen zur Spezifizierung der Anforderung für die Sicherheitsintegrität der Sicherheitsfunktionen in der Norm DIN EN 61508 (nach [DIN02]).

sicherheitsrelevantes System (engl. safety-relevant system)

Ein System ist sicherheitsrelevant, wenn eine Fehlfunktion im System im Regelfall zu einer unmittelbaren Gefahr für Leib und Leben von Verkehrsteilnehmern führen kann. Die Fahrsituation ist durch die Fahrzeuginsassen nicht beherrschbar oder beeinflussbar.

Single Point of Failure

Als Single Point of Failure bezeichnet man eine singuläre Fehlerursache, beispielsweise für einen Common Mode Fehler.

Software (engl. software)

Software ist ein geistiges Produkt, das aus Programmen, Verfahren, Daten und allen dazugehörigen Beschreibungen besteht, die zur Arbeit mit einem Datenverarbeitungssystem gehören (nach [DIN02]).

Softwareeinheit

Element der Erzeugnisstruktur, das ausschließlich aus Software (in der Regel mehrere parallelen Prozesse oder Tasks) besteht. Sie setzt sich meist aus mehreren Softwarekomponenten zusammen (nach [IAB02a]).

Softwarekomponente

Softwarebaustein einer Softwareeinheit. Softwarekomponenten können ihrerseits andere Softwarekomponenten, Softwaremodule und/oder Datenbanken enthalten (nach [IAB02a]).

Softwaremodul

Softwaremodule sind die kleinsten zu programmierenden Softwarebausteine einer Softwareeinheit, deren Behandlung noch durch das V-Modell '97 geregelt wird (nach [IAB02a]). Sie besteht aus Programmteilen und/oder Datendeklarationen.

Ein Softwaremodul wird anhand folgender Kriterien gebildet: Abgeschlossenheit, Geheimnisprinzip, Datenabstraktion, Kapselung, Schnittstellenspezifikation, Schnittstellenminimalität, Überschaubarkeit, Testbarkeit.

stationäre Verfügbarkeit

Mittlere Betriebsdauer zwischen zwei Ausfällen dividiert durch die Summe aus mittlerer Betriebsdauer zwischen zwei Ausfällen und mittlerer Störungsdauer.

Die Verfügbarkeit V ist definiert als

$$V = \frac{MTBF}{MTBF + MDT}$$

mit $MTBF$ als der Mean Time Between Failures (Mittlerer Ausfallabstand) und MDT als Mean Downtime (Mittlere Störungsdauer)

(nach [DIN90])

Störung (engl. malfunction, interference)

(Vorübergehende) Beeinträchtigung einer Funktion (nach [VDI00]).

Submodell (engl. submodel)

Ein aus einer bestimmten Sicht abgeschlossenes Teilmodell des V-Modell ’97. Das V-Modell ’97 besteht aus den Submodellen Systemerstellung, Qualitätssicherung, Konfigurationsmanagement und Projektmanagement.

System (engl. system)

Einheitliches Ganzes, das aus einem oder mehreren Prozessen, Hardware, Software, Einrichtungen und Personen besteht, das die Fähigkeit besitzt, vorgegebene Forderungen oder Ziele zu befriedigen (nach [ISO95]).

Systematischer Fehler (engl. systematic failure)

Ein systematischer Fehler ist eine Fehlerauswirkung, bei der eindeutig auf eine Ursache geschlossen werden kann, die nur durch eine Modifikation des Entwurfs oder des Fertigungsprozesses, der Art und Weise des Betreibens, der Bedienungsanleitung oder anderer Einflussfaktoren beseitigt werden kann (nach [DIN02]).

tolerierbares Risiko (engl. tolerable risk)

Risiko, das basierend auf den aktuellen gesellschaftlichen Wertvorstellungen in einem gegebenen Zusammenhang tragbar ist (nach [DIN02]).

Transienter Fehler

Unter transienten Fehlers versteht man Fehler, die sehr selten auftreten und meist nicht reproduzierbar sind. Diese können beispielsweise durch elektromagnetische Störungen, Strahlungs-, Vibrations- oder Temperatureinflüsse hervorgerufen werden (nach [Sch92]).

Validierung (engl. validation)

Bestätigen aufgrund einer Untersuchung und durch Führung eines Nachweises, dass die besonderen Forderungen für einen speziellen vorgesehenen Gebrauch (Erwartungshaltung des Anwenders) erfüllt worden sind (nach [DIN92]).

Validierung ist die Überprüfung, ob das richtige System entwickelt wurde.

Verfügbarkeit (engl. availability)

Verfügbarkeit ist die Wahrscheinlichkeit für eine korrekte Funktion im Sinne der Zuverlässigkeit (nach [VDI00]).

Verifikation (engl. verification)

Bestätigen aufgrund einer Untersuchung und durch Führung eines Nachweises, dass die festgelegten Forderungen erfüllt worden sind (nach [DIN92]).

Verifikation ist die Überprüfung, ob das System richtig entwickelt wurde.

Zertifizierung (engl. certification)

Verwaltungsmaßnahmen zur Erlangung der Zulassung einer Funktionseinheit durch eine unabhängige Organisation zu einem bestimmten Einsatzzweck (nach [IAB02a]).

Zuverlässigkeit (engl. reliability)

Zuverlässigkeit ist die Fähigkeit eines Systems, während einer vorgegebenen Zeitdauer bei zulässigen Betriebsbedingungen ein funktionsgerechtes Verhalten zu erbringen (d. h. korrekt zu arbeiten).

Voraussetzung ist, dass das System zu Anwendungsbeginn korrekt war und nur Ausfälle zu Inkorrektheit führen können.

Mathematisch ist die Zuverlässigkeit eines Systems die Wahrscheinlichkeit dafür, dass das System die geforderte Funktion unter vorgegebenen Arbeitsbedingungen während einer festgelegten Zeitdauer ausfallfrei ausführt.

Das Auftreten eines Ausfalls kann mit einer Ausfallwahrscheinlichkeit $F(t)$ beschrieben werden. Das Komplement zur Ausfallwahrscheinlichkeit $F(t)$ ist die Überlebenswahrscheinlichkeit $R(t)$. Es gilt

$$R(t) = 1 - F(t).$$

Die Überlebenswahrscheinlichkeit $R(t)$ gibt die Wahrscheinlichkeit dafür an, ein System nach einer Zeit t in funktionsfähigem Zustand anzutreffen, vorausgesetzt das System war zum Startzeitpunkt t_0 korrekt.

Allgemein lautet die Formel für stetige Überlebenswahrscheinlichkeiten $R(t)$

$$R(t) = e^{-\int_{t_0}^{t} \lambda(\xi)\, d\xi}.$$

Die meistverwendete Verteilungsfunktion für die Überlebenswahrscheinlichkeit $R(t)$ von mechanischen oder elektronischen Komponenten, Geräten und Systemen

ist die Weibullverteilung. Die 3-parametrige Weibullverteilung hat die Dichtefunktion

$$f(t) = \frac{b}{T - T_0}\left(\frac{t - T_0}{T - T_0}\right)^{b-1} \cdot e^{-\left(\frac{t-T_0}{T-T_0}\right)^b}.$$

Dabei ist b die sogenannte „Ausfallsteilheit“, T die charakteristische Lebensdauer und T_0 die ausfallfreie Zeit.

Gilt $T_0 = 0$, so spricht man von einer 2-parametrigen Weibull-Verteilung. Es gilt dann für die Überlebenswahrscheinlichkeit

$$R(t) = e^{-(t/T)^b}.$$

Zur Beschreibung von Frühausfällen verwendet man $b < 1$, mit $b > 1$ beschreibt man Verschleißverhalten. Für $b = 1$ reduziert sich die Weibullverteilung auf die Exponentialverteilung. Es gilt dann für konstantes $\lambda = 1/T$

$$R(t) = e^{-\lambda \cdot t}.$$

Charakteristisch für die Exponentialverteilung ist, dass ein neues Objekt das gleiche zukünftige Ausfallverhalten hat wie ein altes Objekt, von dem man weiß, dass es gerade noch lebt. Die Exponentialverteilung ist daher für die meisten elektrischen Komponenten gültig.

Eine weitere Kenngröße für die Zuverlässigkeit ist die mittlere Lebensdauer einer Einheit, meist als MTTF („Mean Time To Failure“) bezeichnet. Die MTTF ist der Erwartungswert $E(\tau)$ der Lebensdauer τ, den man aus dem Integral

$$MTTF = E(\tau) = \int_0^\infty t \cdot f(t)\,dt = \int_0^\infty R(t)\,dt$$

erhält. Für eine Exponentialverteilung mit konstanter Ausfallrate λ gilt dann

$$MTTF = \int_0^\infty e^{-\lambda \cdot t}\,dt = \frac{1}{\lambda} = T.$$

Bei konstantem λ kann man noch die mittlere Ausfallzeit MTBF („Mean Time Between Failures“) definieren. Wenn in einem gegebenen Zeitraum Δt n von N Komponenten ausfallen, so gilt

$$MTBF = \frac{\Delta t \cdot N}{n}.$$

Kann das System repariert werden, so kann man eine mittlere Reparaturzeit MTTR („Mean Time To Repair“) definieren. Damit gilt für die MTBF:

$$MTBF = MTTF + MTTR$$

Literaturverzeichnis

[ADA01] ADAC. *ADAC-Pannenstatistik*. ADAC Allgemeiner Deutscher Automobil-Club e.V., 2001.

[ADM+00] S. AMBERKAR, J. G. D'AMBROSIO, B. T. MURRAY, J. WYSOCKI und B. J. CZERNY. *A System-Safety Process For By-Wire Automotive Systems*. In *SAE World Congress*. 2000.

[AHKSC02] M. AUERSWALD, M. HERRMANN, S. KOWALEWSKI und V. SCHULTE-COERNE. *Entwurfsmuster für fehlertolerante softwareintensive Systeme*. at - Automatisierungstechnik, S. 389–398, August 2002.

[Bal01] H. BALZERT. *Lehrbuch der Software-Technik - Software-Entwicklung*. Spektrum Verlag, 2001.

[Bau99] H. BAUER. *Kraftfahrtechnisches Taschenbuch*. Robert Bosch GmbH, 23. Auflage, 1999.

[BB94] J. B. BOWLES und R. D. BONNELL. *Failure Mode, Effects, and Criticality Analysis*. In *Annual Reliability And Maintainability Symposium*. 1994.

[BDDM04] S. BENZ, E. DILGER, W. DIETERLE und K. D. MÜLLER-GLASER. *A Design Methodology for Safety-Relevant Automotive Electronic Systems*. In *SAE World Congress*. 2004.

[BDM99] T. BERTRAM, P. DOMINKE und B. MÜLLER. *The Safety-Related Aspect of CARTRONIC*. In *SAE World Congress*. 1999.

[Bea00] J. BEAUFAYS. *Air Navigation System Safety Assessment Methodology*. Technischer Bericht, Eurocontrol, 2000.

[BEE+96] T. BELTZ, B. EDENHOFER, J. EILERS, D. GAIDE, A. KRAUSE, G. LASCHET, H.-J. PFEUFER und D. ZANDER. *Qualitätsmanagement in der Automobilindustrie - Band 4 Teil 2: System FMEA*. VDA - Verband der Automobilindustrie, 1996.

[Ben03] S. BENZ. *Eine Entwicklungsmethodik für sicherheitsrelevante Elektroniksysteme im Automobil*. In *15. Workshop Testmethoden und Zuverlässigkeit von Schaltungen und Systemen*. Kooperationsgemeinschaft Rechnergestützter Schaltungs- und Systementwurf von GI, ITG, GMM, Fachgruppe 5 - Testmethoden und Zuverlässigkeit von Schaltungen und Systemen, März 2003.

[BEP+82] B. W. BOEHM, J. F. ELWELL, A. B. PYSTER, E. D. STUCKLE und R. D. WILLIAMS. *The TRW Software Productivity System*. In *IEEE Proc 6th Int. Conf. on Software Engineering*, S. 148–156. Tokyo, September 1982.

[BGH+86] P. BITTER, H. GROSS, H. HILLEBRAND, E. TRÖTSCH und A. WEIHE. *Technische Zuverlässigkeit, mathematische Grundlagen, Untersuchungsmethoden, Anwendungen*. Springer Verlag, Berlin, Heidelberg, New York, 1986.

[BHR90] K. BAREIS, H. HOFFMANN und L. ROSSINELLI. *PAAG-Verfahren (HAZOP) - Risikobegrenzung in der Chemie*. Internationale Sektion der IVSS für die Verhütung von Arbeitsunfällen und Berufskrankheiten in der chemischen Industrie, Heidelberg, 1990.

[BIA97] BIA-REPORT EN 954-1. *Kategorien für sicherheitsbezogene Steuerungen nach EN 954-1*. Hauptverband der gewerblichen Berufsgenossenschaften, 1997.

[Bie03] U. BIEGERT. *Ganzheitliche modellbasierte Sicherheitsanalyse*. atp - automatisierungstechnische Praxis, S. 42–49, August 2003.

[Bin01] M. BINFET-KULL. *Entwicklung einer Steer-by-Wire-Architektur nach zuverlässigkeits- und sicherheitstechnischen Vorgaben*. Dissertation, Bergische Universität - Gesamthochschule Wuppertal, 2001.

[Bir97] A. BIROLINI. *Zuverlässigkeit von Geräten und Systemen*. Springer, Berlin, 1997.

[Bis90] P. G. BISHOP. *Dependability of Critical Computer Systems - Band 3: Techniques Directory*. Elsevier Applied Science, London, New York, 1990.

[BKM04] S. BENZ, M. KÜHL und K. D. MÜLLER-GLASER. *Future Generation Software Architectures in the Automotive Domain*. In *Automotive Software Workshop*. Januar 2004.

[Boe87] B. BOEHM. *A Spiral Modell of Software Development and Enhancement*. In *IEEE Software Engineering Project Management*, S. 513–527. 1987.

[Bor02] J. BORTOLAZZI. *Skriptum zur Vorlesung „Systems Engineering for Automotive Electronics“*. Institut für Technik der Informationsverarbeitung der Universität Karlsruhe, 2002.

[BRI99] *Bosch Research Info: Schnelle Systementwicklung*. Robert Bosch GmbH, Ausgabe 3 1999.

[BTS+02] T. BERTRAM, P. TORRE FLORES, J. SCHIRMER, J. PETERSEN, A. LAPP, D. KRAFT und W. HERMSEN. *Software-Entwicklung für vernetzte Steuergeräte - von der Cartronic-Domänenstruktur zum Steuergerätecode*. Automobiltechnische Zeitschrift ATZ, Band Sonderausgabe Automotive Electronics:S. 32–41, März 2002.

[Bur00] A. BURST. *Rapid Prototyping eingebetteter elektronischer Systeme auf Basis des CDIF-Datenaustauschformats*. Dissertation, Universität Karlsruhe (TH), 2000.

[CDJM03] B. J. CZERNY, J. G. D'AMBROSIO, P. O. JACOB und B. T. MURRAY. *Identifying and Understanding Relevant System Safety Standards for use in the Automotive Industry*. In *SAE World Congress*. 2003.

[Chr92] G. CHROUST. *Modelle der Software-Entwicklung*. Oldenbourg Verlag, München, 1992.

[Cri91] F. CRISTIAN. *Understanding fault tolerant distributed systems*. Communications of the ACM, Band 34:S. 56–78, Februar 1991.

[Dai02] S. DAIS. *Elektronik und Sensorik: Basis der Sicherheit*. In *Technischer Kongress „Sicherheit durch Elektronik“*. VDA - Verband der Automobilindustrie, März 2002.

[DD02] E. DILGER und W. DIETERLE. *Fehlertolerante Elektronikarchitekturen für sicherheitsgerichtete Kraftfahrzeugsysteme*. at - Automatisierungstechnik, S. 375–381, August 2002.

[DeM79] T. DEMARCO. *Structured Analysis and System Specification*. Yourdon Press, New York, 1979.

[Dep95] DEPARTMENT OF DEFENSE. *MIL-Handbook 217: Reliability Prediction of Electronic Equipment*. 1995.

[Dil02] E. DILGER. *x-by-wire-Systeme - Fokus Sicherheit*. In *IIR-Konferenz: Erfolgsfaktor im Auto der Zukunft: x-by-wire*. Robert Bosch GmbH, Stuttgart, März 2002.

[DIN87] DIN VDE 31000, TEIL 2. *Allgemeine Leitsätze für das sicherheitsgerechte Gestalten technischer Erzeugnisse. Begriffe der Sicherheitstechnik, Grundbegriffe*. Dezember 1987.

[DIN90] DIN 40041. *Zuverlässigkeit - Begriffe*. 1990.

[DIN92] DIN EN ISO 8402. *Qualitätsmanagement und Qualitätssicherung - Begriffe*. Norm, DIN, 1992.

[DIN95] DIN V 19250. *Leittechnik - Grundlegende Sicherheitsbetrachtungen für MSR-Schutzeinrichtungen*. Norm, DIN, 1995.

[DIN02] DIN EN 61508. *Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme*. 2002.

[DMF] E. DILGER, B. MÜLLER und T. FÜHRER. *Distributed Fault Tolerant and Safety Critical Applications in Vehicles - A Time Triggered Approach*. Technischer Bericht, Robert Bosch GmbH.

[DO 92] DO 178 B. *Software Considerations in Airborne Systems and Equipment Certification*. RTCA, 1992.

[EM03] B. ELBEL und O. MÄCKEL. *Softwarezuverlässigkeit effizient bewerten - ein Erfahrungsbericht aus der Praxis*. In *15. Workshop Testmethoden und Zuverlässigkeit von Schaltungen und Systemen*. Kooperationsgemeinschaft Rechnergestützter Schaltungs- und Systementwurf von GI, ITG, GMM, Fachgruppe 5 - Testmethoden und Zuverlässigkeit von Schaltungen und Systemen, März 2003.

[EN 96] EN 1050. *Sicherheit von Maschinen - Leitsätze zur Risikobeurteilung*. 1996.

[EN 97] EN 954. *Sicherheit von Maschinen - Sicherheitsbezogene Teile von Steuerungen*. 1997.

[ETA04] ETAS GMBH. *Homepage*. www.etas.de, letzter Abruf: Februar 2004.

[Fri01] A. FRITZ. *Berechnung und Monte-Carlo Simulation der Zuverlässigkeit und Verfügbarkeit technischer Systeme*. Dissertation, Universität Stuttgart, 2001.

[Gom00] H. GOMAA. *Designing Concurrent , Distributed and Real-Time Applications with UML*. Addison Wesley, 2000.

[Gör89] W. GÖRKE. *Fehlertolerante Rechensysteme*. Oldenbourg, 1989.

[Gör97] W. GÖRKE. *Skriptum zur Vorlesung Zuverlässigkeitsprobleme elektrischer Geräte*. Institut für Rechnerentwurf und Fehlertoleranz der Universität Karlsruhe, 1997.

[Gor03] J. GORONCY. *Elektronik-Innovationen um jeden Preis: Interview mit Stephan Wolfsried, Direktor Pkw-Entwicklung bei Mercedes-Benz*. VDI nachrichten, S. 10–11, 21. März 2003.

[GR02] G. GLÖE und G. RABE. *Qualifizierung von Software für Kraftfahrzeuge*. In *11. Aachener Kolloquium für Fahrzeug- und Motorentechnik*. 2002.

[Hed01] B. HEDENETZ. *Entwurf von verteilten, fehlertoleranten Elektronikarchitekturen in Kraftfahrzeugen*. Dissertation, Universität Tübingen, 2001.

[Hen96] V. HENAULT. *Méthodologie de développement des systèmes électroniques embarqués automobiles, matériels et logiciels, sûrs de fonctionnement*. Dissertation, IRESTE, Nantes, 1996.

[HP87] D. J. HATLEY und I. A. PIRBHAI. *Strategies for Real-Time System Specification*. Dorset House, 1987.

[HSE04] HSE - HEALTH AND SAFETY EXECUTIVE. *Reducing risks, protecting people. HSE's decision-making process*. www.hse.gov.uk, letzter Abruf: Februar 2004.

[IAB02a] IABG - INDUSTRIEANLAGEN-BETRIEBSGESELLSCHAFT. *V-Modell '97 Spezifikation: Entwicklungsstandard für IT-Systeme des Bundes - Vorgehensmodell*. www.v-modell-iabg.de, letzter Abruf: November 2002.

[IAB02b] IABG - INDUSTRIEANLAGEN-BETRIEBSGESELLSCHAFT. *V-Modell '97 Spezifikation - Anhang zu Nr. 250/3*. www.v-modell-iabg.de, letzter Abruf: November 2002.

[Inf04] INFORMATION ON FORMAL METHODS. *Homepage*. www.afm.sbu.ac.uk, letzter Abruf: Januar 2004.

[ISO95] ISO/IEC 12207. *Information technology - Software life cycle processes*. Norm, 1995.

[JW03] C. JUNG und M. WOLTERECK. *Vorschlag eines Funktionssicherheitsprozesses für die verteilte Entwicklung sicherheitsrelevanter Systeme*. In *VDI-Berichte: Elektronik im Kraftfahrzeug Baden-Baden*. 2003.

[KBC+03] M. KOORNSTRA, J. BROUGHTON, J.-P. CAUZARD, R. ESBERGER, A. EVANS, C. GLANSDORP, L. HANTULA, W. KÖPPEL, M. PIERS, F. TAYLOR und W. VANLAAR. *Transport Safety Performance in the EU - A Statistical Overview*. European Transport Safety Council, 2003.

[KG02a] H. KIRRMANN und K.-E. GROSSPIETSCH. *Fehlertolerante Steuerungs- und Regelungssysteme*. at - Automatisierungstechnik, S. 362–374, August 2002.

[KG02b] H. KIRRMANN und K.-E. GROSSPIETSCH. *Fehlertolerante Systeme in der Automatisierungstechnik*. at - Automatisierungstechnik, S. 359–361, August 2002.

[Kif03] A. KIFMANN. *Nur Fahren ist sicherer*. Auto & Elektronik, Band 01:S. 3, 2003.

[Kne00] R. KNEPPER. *Der Sicherheits- und Zuverlässigkeitsprozess in der zivilen Luftfahrtindustrie*. In *Sicherheit komplexer Verkehrssysteme*. VDI-Verlag, 2000.

[Kop88] H. KOPETZ. *Fehlermodelle in verteilten Echtzeitsystemen*. Springer-Verlag, 1988.

[Kop97] H. KOPETZ. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Kluwer Academic Publishers, 1997.

[Län03] W. LÄNGST. *Formale Anwendung von Sicherheitsmethoden bei der Entwicklung verteilter Systeme*. Dissertation, Universität Karlsruhe (TH), 2003.

[Lap92] J. C. LAPRIE. *Dependability: Basic Concepts and Terminology in English, French, German, Italian and Japanese*. Springer, 1992.

[Len95] R. LENK. *Zuverlässigkeitsanalyse von komplexen Systemen am Beispiel PKW-Automatikgetriebe*. Dissertation, Universität Stuttgart, 1995.

[Lev95] N. LEVESON. *Safeware*. Addison-Wesley Publishing, 1995.

[LTS+01] A. LAPP, P. TORRE FLORES, J. SCHIRMER, D. KRAFT, W. HERMSEN, T. BERTRAM und J. PETERSEN. *Softwareentwicklung für Steuergeräte im Systemverbund - Von der CARTRONIC-Domänenstruktur zum Steuergerätecode*. VDI-Berichte: Elektronik im Kraftfahrzeug Baden-Baden, 2001.

[Mah00] R. MAHMOUD. *Sicherheits- und Verfügbarkeitsanalyse komplexer Kfz-Systeme*. Dissertation, Universität-Gesamthochschule Siegen, 2000.

[MB97] F. MOSNIER und J. BORTOLAZZI. *Prototyping Car-Embedded Applications*. Advances in Information Technologies: The Business Challenge, S. 744–751, 1997.

[MIL99] MIL-STD-882D. *System Safety Program Requirements*. 1999.

[MIS04] MISRA - MOTOR INDUSTRY SOFTWARE RELIABILITY ASSOCIATION. *Guidelines for the Use of the C Language in Vehicle Based Software*. http://www.misra.org.uk, letzter Abruf: Januar 2004.

[MKB03] K. D. MÜLLER-GLASER, M. KÜHL und S. BENZ. *Management und Verhalten verteilter Systeme im Kfz*. IIR Deutschland, 2003.

[Mon99] S. MONTENEGRO. *Sichere und fehlertolerante Steuerungen - Entwicklung sicherheitsrelevanter Systeme*. Hanser, 1999.

[MP03] A. MEYNA und B. PAULI. *Taschenbuch der Zuverlässigkeits- und Sicherheitstechnik*. Hanser Verlag, 2003.

[MS04] C. C.-A. R. MODELING und SIMULATION. *Homepage*. www.tc.umn.edu/ puk/carms.htm, letzter Abruf: März 2004.

[MSR03] MSR: MANUFACTURER SUPPLIER RELATIONSHIP. *Homepage*. www.msr-wg.de, letzter Abruf: November 2003.

[Mül04] B. MÜLLER. *Sicherheit und Zuverlässigkeit in redundanten Systemen*, 2004. Interner Bericht, Robert Bosch GmbH.

[Nel02] J. NELL. *Steer by Wire: Ein Blick in die Zukunft?*. In *8. Esslinger Forum für Kfz-Mechatronik*. Continental Teves AG, November 2002.

[OMG04] OMG - OBJECT MANAGEMENT GROUP. *Homepage*. www.omg.org, letzter Abruf: Februar 2004.

[Oxf89] *Oxford Andvanced Learner's Dictionary of current English*. Oxford University Press, 4. Auflage, 1989.

[Pau97] B. PAULI. *Zuverlässigkeitsprognosen für elektrische Steuergeräte im Kraftfahrzeug: Modellbildungen und deren praktische Anwendungen*. Dissertation, Universität-Gesamthochschule Wuppertal, 1997.

[RAC04] RAC: RELIABILITY ANALYSIS CENTER. *Homepage*. http://rac.alionscience.com/prism/, letzter Abruf: Februar 2004.

[RAM03] RAMS-CONSULT. *Homepage*. http://www.rams-consult.com/beratung/zuverlaessigkeit.html, letzter Abruf: Dezember 2003.

[Rat88] B. RATCLIFFE. *Early and not-so early prototyping - rationale and support*. In *Proceedings COMPSAC88*. IEEE Computer Society Press, 1988.

[Rei98] G. REICHART. *Sichere Elektronik im Kraftfahrzeug*. at - Automatisierungstechnik, S. 78–83, Februar 1998.

[Rei02] G. REINKING. *General Motors ruft in den USA 1,5 Millionen Autos zurück*. Financial Times Deutschland, 18.11.2002.

[Roy70] W. W. ROYCE. *Managing the Development of Large Software Systems*. In *Proceedings of IEEE WESCON*. August 1970.

[SAE96a] SAE AEROSPACE RECOMMENDED PRACTICE 4754. *Certification Considerations for Highly-Integrated or Complex Aircraft Systems*. Standard, SAE - The Engineering Society for Advancing Mobility Land Sea Air and Space, 1996.

[SAE96b] SAE AEROSPACE RECOMMENDED PRACTICE 4761. *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*. Standard, SAE - The Engineering Society for Advancing Mobility Land Sea Air and Space, 1996.

[Sch92] E. SCHRÜFER. *Lexikon Meß- und Automatisierungstechnik*. VDI-Verlag GmbH, 1992.

[SEI04] SEI - SOFTWARE ENGINEERING INSTITUTE. *Homepage*. www.sei.cmu.edu, letzter Abruf: Januar 2004.

[SET04] SETTA - SYSTEMS ENGINEERING OF TIME TRIGGERED ARCHITECTURES. *Homepage*. www.setta.org, letzter Abruf: Februar 2004.

[SGW94] B. SELIC, G. GULLEKSON und P. T. WARD. *Real-Time Object-Oriented Modeling*. Wiley, 1994.

[SKR+03] H. SCHWAB, A. KLÖNNE, S. RECK, I. RAMESOHL, G. STURTZER und B. KEITH. *Reliability evaluation of a permanent magnet synchronous motor drive for an automotive application*. In *10th European Conference on Power Electronics and Applications*. September 2003.

[Spi02] SPIEGEL ONLINE. *ESP-Einsatz - Die Unfall-Bremse*. www.spiegel.de, Dezember 2002.

[Spi03] SPIEGEL ONLINE. *Immer häufiger Pannen durch mangelhafte Elektronik*. www.spiegel.de, Januar 2003.

[SS01] D. J. SMITH und K. G. L. SIMPSON. *Functional Safety - A Straightforward Guide to IEC 61508 and Related Standards*. Butterworth-Heinemann, 2001.

[Sta01] STATISTISCHES BUNDESAMT. *Statistisches Jahrbuch 2000: Verkehr - Verkehrsunfälle*. Statistisches Bundesamt, Wiesbaden, 2001.

[Sta02] STATISTISCHES BUNDESAMT. *Statistisches Jahrbuch 2001: Verkehr - Verkehrsunfälle*. Statistisches Bundesamt, Wiesbaden, 2002.

[Sta03] STATISTISCHES BUNDESAMT. *Statistisches Jahrbuch 2002: Verkehr - Verkehrsunfälle*. Statistisches Bundesamt, Wiesbaden, 2003.

[STE03] STEP-X. *Struktruierter Entwicklungsprozess am Beispiel von X-by-Wire-Anwendungen*. www.step-x.de, letzter Abruf: August 2003.

[Sto96] N. STOREY. *Safety-Critical Computer Systems*. Addison-Wesley, 1996.

[Stö00] S. STÖLZL. *Fehlertolerante Pedaleinheit für ein elektromechanisches Bremssystem (Brake-by-Wire)*. Dissertation, Technische Universität Darmstadt, 2000.

[Str83] H. STRÖMER. *Mathematische Theorie der Zuverlässigkeit*. Oldenbourg, München, Wien, 1983.

[Tel03] TELELOGIC. *Homepage*. www.telelogic.com, letzter Abruf: Dezember 2003.

[UN 01a] UN ECE-R 13. *Uniform provisions concerning the approval of vehicles of categories M, N and O with regard to braking*. Regulation, 2001.

[UN 01b] UN ECE-R 79. *Uniform provisions concerning the approval of vehicles with regard to steering equipment*. Regulation - Draft, 2001.

[Uni00] UNION TECHNIQUE DE L'ELECTRICITÉ. *RDF 2000: Reliability Data Handbook*. 2000.

[Vah98] A. VAHL. *Interaktive Zuverlässigkeitsanalyse von Flugzeug-Systemarchitekturen*. Dissertation, Technische Universität Hamburg-Harburg, 1998.

[VDA00] VDA - VERBAND DER AUTOMOBILINDUSTRIE. *Zuverlässigkeitssicherung bei Automobilherstellern und Lieferanten*. VDA - Verband der Automobilindustrie, 2000.

[VDI85] VDI 3691. *Erfassung von Zuverlässigkeitswerten bei Prozeßrechnereinsätzen. VDI-Handbuch Technische Zuverlässigkeit*. VDI, VDI-Gesellschaft Systementwicklung und Projektgestaltung, 1985.

[VDI93] VDI-GEMEINSCHAFTSAUSSCHUSS INDUSTRIELLE SYSTEMTECHNIK (HRSG.). *Software Zuverlässigkeit: Grundlagen, konstruktive Maßnahmen, Nachweisverfahren*. VDI-Verlag, Düsseldorf, 1993.

[VDI00] VDI/VDE-RICHTLINIE 3542. *Sicherheitstechnische Begriffe für Automatisierungssysteme. Blatt 1: Qualitative Begriffe. Blatt 2: Quantitative Begriffe und Definitionen. Blatt 3: Anwendungshinweise und Beispiele. Blatt 4: Zuverlässigkeit und Sicherheit komplexer Systeme (Begriffe)*. VDI-Gesellschaft Systementwicklung und Projektgestaltung, Oktober 2000.

[WBD+02] S. WÖRNER, A. BAUER, A. DROSSOS, C. GROSS, A. WILLIKENS, K. EITZENBERGER und F. MARRONE. *Bordnetz und Telematik - Komfort und Sicherheit*. Maybach-Sonderausgabe von ATZ/MTZ, S. 120–129, September 2002.

[Wer02] S. WERY. *Anwendung der Functional Hazard Analysis (FHA) in der Eisenbahnsignaltechnik am Beispiel ETCS Level 2*. Diplomarbeit, Technische Universität Dresden, 2002.

[WJR04] M. WOLTERECK, C. JUNG und G. REICHART. *How to Achieve Functional Safety and What Safety Standards and Risk Assessment Can Contribute*. In *SAE World Congress*. 2004.

[WK] P. J. WILKINSON und T. P. KELLY. *Functional Hazard Analysis for Highly Integrated Aerospace Systems*. Technischer Bericht, Department of Computer Science, University of York.

[WM85] P. T. WARD und S. J. MELLOR. *Structured Development for Real-Time Systems*. Yourdon Press (Prentice Hall), Englewood Cliffs, 1985.

[X-B98] X-BY-WIRE TEAM. *X-By-Wire: Safety Related Fault Tolerant Systems in Vehicles - Final Report*. Technischer Bericht, European Union, http://www.vmars.tuwien.ac.at/projects/xbywire/, 1998.

[Yeh96] Y. C. YEH. *Triple-Triple Redundant 777 Primary Flight Computer*. In *IEEE Aerospace Applications Conference*. Februar 1996.

Verwendete Abkürzungen

Bezeichnung	Bedeutung
a	(Betriebs-)Jahr
ABS	Antiblockiersystem
ADAC	Allgemeiner Deutscher Automobil-Club e. V.
AG	Aktiengeselleschaft
AK	Anforderungsklasse
ALARP	As Low As Reasonably Practicable (so niedrig wie vernünftigerweise möglich)
ARP	Aerospace Recommended Practice
ASIC	Application Specific Integrated Circuit
ATZ	Automobiltechnische Zeitschrift
BMS	Batteriemanagement-System
BMW	Bayrische Motoren-Werke
CCA	Common Cause Analysis (Analyse gemeinsamer Ursachen nach SAE ARP 4761)
CMA	Common Mode Analysis (Analyse redundanzüberbrückender Fehler)
CMM	Capability Maturity Model des SEI
CMMI	Capability Maturity Model Integration des SEI
COMET	Concurrent Object Modelling and Architectural Design Method
CRC	Cyclic Redundancy Check
CSA	CARTRONIC Safety Analysis (CARTRONIC Sicherheitsanalyse)
DIN	Deutsches Institut für Normung

Fortsetzung auf der nächsten Seite

Bezeichnung	**Bedeutung**
DoD	Department of Defense (US-amerikanisches Verteidigungsministerium)
EADS	European Aeronautic Defence and Space Company
ECE	Economic Commission for Europe
ECU	Electronic Control Unit (engl. für Steuergerät)
EHB	Elektrohydraulische Bremse
EMV	Elektromagnetische Verträglichkeit
EN	European Norm (Europäische Norm)
ESP	Elektronisches Stabilitätsprogramm
ESSB	Entwurfsbegleitende System-Sicherheitsbewertung
ETA	Event Tree Analysis (Ereignisbaumanalyse)
ETSC	European Transport Safety Council
EU	Europäische Union
EUC	Equipment Under Control (zu regelndes oder zu steuerndes System)
FAA	Federal Aviation Association (US-amerikanische Luftfahrtbehörde)
FAR	Federal Aviation Regulation (US-amerikanische Luftfahrt-Vorschrift)
FE	Funktionaler Entwurf
FGA	Funktionale Gefährdungsanalyse
FHA	Functional Hazard Assessment (Funktionale Gefährdungsanalyse nach SAE ARP 4761)
fit	failures in time, d. h. $10^{-9}/h$
FMEA	Failure Modes and Effects Analysis (Fehler-Möglichkeits- und Einfluss-Analyse)
FMECA	Failure Modes, Effects and Criticality Analysis (Fehler-Möglichkeits-, Einfluss- und Kritikalitäts-Analyse)
FO	fail-operational
FS	fail-silent
FTA	Fault Tree Analysis (Fehlerbaumanalyse)
GI	Gesellschaft für Informatik

Fortsetzung auf der nächsten Seite

Bezeichnung	**Bedeutung**
GmbH	Gesellschaft mit beschränkter Haftung
GMM	VDE/VDI-Gesellschaft Mikroelektronik, Mikro- und Feinwerktechnik
h	(Betriebs-)Stunde
HAZOP	Hazard and Operability Studies
HDBK	Handbook
HW	Hardware
IABG	Industrieanlagen-Betriebsgesellschaft
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers, Inc.
Impl	Implementierung
ISO	International Organization for Standardization
IT	Informationstechnik
IT	Integration und Test
ITG	Informationstechnische Gesellschaft im VDE
JAA	Joint Aviation Authority (Zusammenschluss der europäischen Luftfahrtbehörden)
JAR	Joint Aviation Regulation (europäische Luftfahrt-Vorschrift)
Kfz	Kraftfahrzeug
KM	Submodell Konfigurationsmanagement im V-Modell ’97
Lkw	Lastkraftwagen
MIL	Military (militärisch)
MISRA	Motor Industry Software Reliability Association
MSR	Messen, Steuern, Regeln
MSR	Manufacturer-Supplier-Relationship
MTBF	Mean Time Between Failures (mittlere Zeit zwischen zwei Ausfällen)
MTTF	Mean Time To Failure (mittlere Zeit bis zum ersten Ausfall)
MTTR	Mean Time To Repair (mittlere Reparaturzeit)
MTZ	Motortechnische Zeitschrift
NASA	National Aeronautics and Space Administration (US-Raumfahrtbehörde)

Fortsetzung auf der nächsten Seite

Bezeichnung	**Bedeutung**
NHTSA	National Highway Traffic Safety Administration (US-Verkehrssicherheitsbehörde)
OOA	Objektorientierte Analyse
PAAG	Prognose von Störungen, Auffinden der Ursachen, Abschätzen der Wirkungen, Gegenmaßnahmen
PAL	Programmable Array Logic
PES	Programmierbares Elektronisches System
PFC	Primary Flight Control
Pkw	Personenkraftwagen
PM	Submodell Projektmanagement im V-Modell ’97
PoF	Physics of Failures (Physik der Ausfälle)
ppm	parts per million, d. h. 10^{-6}
PRA	Particular Risk Analysis (Analyse besonderer Risiken)
PSSA	Preliminary System Safety Assessment (Vorläufige Systems-Sicherheitsbewertung nach SAE ARP 4761)
QS	Submodell Qualitätssicherung im V-Modell ’97
RAC	Reliability Analysis Center
ROOM	Real-Time Object-Oriented Modeling
RPZ	Risikoprioritätszahl
RTCA	Radio Technical Commission for Aeronautics
SA	System-Anforderungsanalyse
SAE	Society of Automotive Engineers
SA/RT	Structured Analysis/Real-Time Analysis
SE	Submodell Systemerstellung im V-Modell ’97
SE	System-Entwurf
SEI	Software Engineering Institute der Carnegie Mellon University, USA
SETTA	Systems Engineering for Time-Triggered Architectures
SEU	Softwareentwicklungsumgebung
SF	Sicherheitsstufe
SG	Steuergerät
SIL	Safety Integrity Level (Sicherheits-Integritätslevel)

Fortsetzung auf der nächsten Seite

Bezeichnung	**Bedeutung**
SMV	Symbolic Model Verifier
SoC	System-on-Chip
SPICE	Software Process Improvement and Capability Determination
SSA	System Safety Assessment (Systems-Sicherheitsbewertung nach SAE ARP 4761)
SSB	System-Sicherheitsbewertung
STD	Standard
STEP-X	Strukturierter Entwicklungsprozess für eingebettete Systeme im Automobilbereich
SW	Software
SWIFI	Software implementierte Fehlerinjektion
TTP/C	Time-Triggered Protocol/SAE Class C
TU	Technische Universität
TÜV	Technischer Überwachungsverein
UML	Unified Modeling Language
UML-RT	UML for Real-Time
UN	United Nations (Vereinte Nationen)
US	United States (Vereinigte Staaten)
USA	United States of America (Vereinigte Staaten von Amerika)
VDA	Verband der Automobilindustrie
VDE	Verband der Elektrotechnik, Elektronik und Informationstechnik e. V.
VDI	Verein Deutscher Ingenieure
VDM	Vienna Development Method
VSE	Verification System Environment
WCET	Worst Case Execution Time
XML	eXtensible Markup Language
ZI	Zulassung und Inebtriebnahme
ZSA	Zonal Safety Analysis (Zonensicherheitsanalyse)

Verwendete Formelzeichen

Bezeichnung	Bedeutung
A	Aufenthaltsdauer im Gefahrenbereich
$\mathbf{A}$	Matrix der Übergangsraten α
$F(t)$	Funktion der Ausfallwahrscheinlichkeit
G	Möglichkeit der Gefahrenabwendung
H	Häufigkeit
I_f	Fraktionale Importanz
I_m	Marginale Importanz
$I_{\bar{\Phi}}$	Strukturelle Importanz
P	Wahrscheinlichkeit
R	Risiko
$R(t)$	Funktion der Überlebenswahrscheinlichkeit
S	Schadensausmaß
T	Zeit(dauer)
V	Verfügbarkeit
W	Wahrscheinlichkeit des Eintritts eines unerwünschten Ereignisses
b	Ausfallsteilheit
$f(t)$	Funktion der Ausfalldichte
t	Zeit(dauer)
$\bar{\Phi}$	Ausfallfunktion

Fortsetzung auf der nächsten Seite

Formelzeichen - Fortsetzung von der vorherigen Seite

Bezeichnung	**Bedeutung**
α	Übergangsrate zwischen zwei Zuständen
β	Element der Matrix **A**
$\lambda(t)$	Ausfallrate bzw. Fehlerrate
μ	Reparaturrate

Index

Eigene Veröffentlichungen

Eine Entwicklungsmethodik für sicherheitsrelevante Elektroniksysteme im Automobil [Ben03]

Beitrag auf dem 15. Workshop über Testmethoden und Zuverlässigkeit von Schaltungen und Systemen der Fachgruppe 5 „Testmethoden und Zuverlässigkeit von Schaltungen und Systemen“ der Kooperationsgemeinschaft Rechnergestützter Schaltungs- und Systementwurf von GI, ITG und GMM in Timmendorfer Strand, März 2003.

Quality assurance and certification of software modules in safety critical automotive electronic control units using a CASE tool integration platform [BKM04]

Beitrag auf dem Automotive Software Workshop „Future Generation Software Architectures in the Automotive Domain - Connected Services in Mobile Networks“ in San Diego, U.S.A., Januar 2004.

A Design Methodology for Safety-Relevant Automotive Electronic Systems [BDDM04]

Beitrag auf dem SAE World Congress 2004, Session AE5 „Safety-Relevant Systems“ in Detroit, U.S.A., März 2004.

Lebenslauf

Stefan Benz
geboren am 17. August 1975 in Böblingen

Schulbildung

1982 - 1986	Grundschule in Böblingen
1986 - 1995	Gymnasium in Böblingen

Studium

Okt 1995 - Sep 1997	Elektrotechnik an der Universität Stuttgart bis zum Vordiplom
Okt 1997 - Apr 2001	Elektrotechnik an der Universität Karlsruhe (TH) Vertiefungsrichtung „Systems Engineering“ Abschluss: Dipl.-Ing.
	Thema der Diplomarbeit: „Entwicklung eines Brake-by-Wire-Systems für Nutz-Kraftfahrzeuge unter Verwendung einer Hardware-in-the-Loop-Entwicklungsumgebung“
während des Studiums Sep 1998 - Jul 1999	Studienaufenthalt an der Oregon State University, U.S.A.

Beruf

seit Mai 2001	Doktorand bei der Robert Bosch GmbH im Bereich Forschung und Vorausentwicklung, Abteilung Informations- und Systemtechnik