3 Systemengineering

Dieses Kapitel zeigt Aspekte auf zu der Frage: „Was ist Systemengineering?“ Wie grenzt sich Systemengineering in verschiedenen Domänen zu den Anforderungen der Automobilindustrie ab. Weiter werden die notwendigen organisatorischen Einflüsse diskutiert. Bei den Prozessmodellen, die allgemein gelehrt werden, gibt es oft keine Antwort auf die Frage, wie kommt man in das V, die Spirale oder in den Wasserfall überhaupt rein. Welche Aspekte müssen bei der Konzeptionierung oder bei den Produktdefinitionsphasen berücksichtigt werden, dass die Aktivitäten innerhalb eines V-Zyklus planbar und damit erst zielgerichtet durchführbar werden. Ziel ist es eine grundsätzliche Vorgehensweisen aufzuzeigen, die in jeder Phase der Produktentwicklung berücksichtigt werden können oder die Grundlage für die Entwicklungsaktivitäten darstellen.

3.1 Geschichtliche und philosophische Hintergründe

Bereits im Altertum gab es Menschen, die nicht alles glaubten, was ihnen gesagt wurde. Auch vor Sokrates nahm man nicht mehr alles als bare Münze, was man gesehen hat. Die Schlüsse, die der ein oder andere Gelehrte zog, wurden in Frage gestellt.

Ob man das Hinterfragen von technischen Zusammenhängen wirklich erst 600 vor Christus in Griechenland begann, es nur von Ägyptern oder den Bewohnern im Zweistromland nicht dokumentiert wurde oder gar viel früher uns von Außerirdischen überliefert wurde, sei dahingestellt. Aber seit man schriftliche Zeugnisse besitzt, versucht man bestimmte Phänomene zu beschreiben und bestimmte Schlüsse daraus zu ziehen.

Die ionischen Philosophen versuchten es schon sehr mathematisch, hier ist Pythagoras zu nennen, der bestimmt nicht geahnt hat, dass wir zur Ansteuerung eines Motors seine Formeln nutzen müssen, um Blind- und Wirkleistung in Beziehung zu bringen.

In der Schule von Elea wird Parmenides nachgesagt, dass er lehrte, man könne bestimmte Dinge beobachten, aber man kann nicht beliebige Schlüsse daraus ziehen. Ob das Beobachtete wirklich den Schluss zulässt, dass es Wahrheit ist, hat er schon vor Sokrates in Frage gestellt. Bis heute, 2600 Jahre später, quälen wir uns mit der Frage herum, ob die Ursache eines negativen Tests tatsächlich eine falsche Testhypothese war oder sogar der Test nicht geeignet war, um eine Aussage zu treffen, ob Anforderungen korrekt und hinreichend umgesetzt werden. Demokrit versuchte den Begriff des Atoms zu definieren als kleinstes Element, aus dem alles besteht, aber selbst Nils Bohr hat zu seinen Lebzeiten noch erfahren müssen, dass es kleinere Elemente als Atome gibt. Nicht nur Albert Einstein hat aufgezeigt, dass es viele Varianten der Interaktion gibt und wir verschiedene Modelle benötigen, um das Beobachtete beschreiben zu können. Aristoteles wusste bereits: „ Das Ganze ist mehr als die Summe seiner Einzelteile.“ Nicht nur die Elemente und ihre Eigenschaften bestimmen, wie sich Elemente zueinander verhalten, sondern es spielt eine große Rolle, in welcher Umgebung die Elemente miteinander interagieren. Heute wissen wir aus eigener Erfahrung, dass eine Schraube und ein Dübel eine andere Haltbarkeit in einer Gipskartonwand als in einer Kalksandsteinwand ergeben. Je nach Design wird es in der einen oder anderen Wand bessere oder schlechtere Ergebnisse geben. Neben dem Beobachten und Schließen hat Aristoteles auch die Induktion ins Gespräch gebracht. Dass heute die vollständige mathematische Induktion als deduktive Methode beschrieben wird, zeigt dass Worte einem Wandel in der Zeit unterworfen sein können. Aber dies wäre nicht die einzige Erfahrung, die die Menschheit über Jahrtausende vergessen hat und erst viel später wieder hart erlernen musste. Roger Bacon beschrieb Anfang des 13ten Jahrhunderts bereits Prinzipien für den Elektromotor bei den Untersuchungen zum Magnetismus. Die Idee des „kontinuierlich laufenden Rades“ führte zu der Erkenntnis dass es ein „Perpetuum mobile“ nicht geben kann. Werner von Siemens hat wohl die Schriften von Roger Bacon später nicht gekannt.

Weitaus aktueller hat Karl Raimund Popper unser Dilemma von heute beschrieben, er sagte mehr oder weniger, man kann gar nichts verifizieren, man kann nur bestimmte Eigenschaften falsifizieren. Dazu wird oft das Beispiel des Schwans genannt. Die Menschheit glaubte immer es gäbe nur weiße Schwäne, bis man in Australien auch schwarze Schwäne entdeckt hat. Wie gehen wir damit um? Das Wort „verifizieren“ beinhaltet die beiden lateinischen Worte „Veritas“ (Wahrheit) und „facere“ (machen). Gut, „im Wein liegt Wahrheit“, aber welche, kann man nur erahnen, wenn man nach reichlichem Genuss am nächsten Tag einen dicken Kopf hat. Scheinbar wird das Wort „verifizieren“ von verschiedenen Menschen und Gruppen gleichwohl auch verschieden genutzt.

Popper hat einige Hinweise zur Falsifizierung hinterlassen. Er sagte, wir können, wenn ein Ergebnis negativ ist, nicht unbedingt die ganze Aussage oder Hypothese

in Frage stellen, er ermutigt uns hieraus neue Erkenntnisse zu gewinnen, die Hinweise geben können, was man verändern muss, damit das Ergebnis positiv wird. Wobei wir aber wohl lernen, selbst wenn alle Tests positiv sind, werden alle unsere Anforderungen noch lange nicht erfüllt sein. Das Positive was wir daraus lernen ist, dass wir unsere negativen Tests analysieren sollten, um Verbesserungen am Produkt finden zu können.

Diese Gedanken führten dazu, dass das Grundprinzip des Sicherheitsnachweises in den Sicherheitsstandards heute wie folgt betrachtet werden kann:

„Wenn alle erdenkbaren Fehler des Systems beherrschbar sind, gilt das System als sicher."

Problematisch wird diese Sichtweise bei Neuentwicklungen beziehungsweise wenn neue Technologie die traditionelle Technologie für bewährte Fahrzeugsysteme ersetzt. Dies gilt für die gesamten „By wire"-Systeme, aber insbesondere für fernsteuerbare Systeme, die bisher rein vom Fahrer bedient wurden. Hier gilt der Grundsatz „Equivalent Level of Safety", das heißt man muss zum Beispiel nachweisen, dass das neue elektronische System genauso sicher ist, wie das konventionelle hydraulische System. Wird ein bewährtes System nach vergleichbaren Prinzipien realisiert, so reicht es die Einhaltung dieser Sicherheitsprinzipien zu zeigen. Bei einem neuen Produkt in neuer Technologie ist ein systematischer Sicherheitsnachweis notwendig. Diese Grundsätze sind neben den Normen auch weitgehend in alle weltweiten und branchenweiten rechtlichen Standards eingeflossen.

Dies war ein kleiner Exkurs in die Philosophie, jedoch sollte man auch auf einige Ingenieure, Physiker und Mathematiker in dem Kapitel hinweisen.

George Boole (1815–1864) gilt als der Erfinder der Boolschen Algebra. Prinzipiell waren die Regeln schon früher bekannt, aber er hatte diese in seinem Buch „An investigation of the law of thoughts" als „Algebra der Logik" formuliert. Augustus DeMorgan formulierte das DeMorgan´sche Gesetz, dies beeinflusst die deduktive Analyse.

Neben der qualitativen induktiven und deduktiven Sicherheitsanalyse kennt die ISO 26262 auch die quantitative Sicherheitsanalyse. Hierzu sollten noch einige Namen von Herren genannt werden, die wesentliche Grundlagen für die Sicherheitstechnik erarbeitet haben.

Robert Lusser hat vor (oder während) dem zweiten Weltkrieg bereits seine „Gesetzmäßigkeit von Zuverlässigkeitsketten" formuliert. Erich Pieruschka ergänzte die Quantifizierung. Diese beiden Herren kannte wohl auch den Russen Kolmogorov oder zumindest sein Buch in deutscher Sprache von 1933 „Grundbegriffe der Wahrscheinlichkeitsrechnung". Das Axiom von Kolmogorov besagt: „Die Wahrscheinlichkeit einer Vereinigung abzählbar vieler inkompatibler Ereignisse

entspricht der Summe der Wahrscheinlichkeiten der einzelnen Ereignisse“ in etwas verkürzter Form. Aus dem Kolmogorow-Smirnow-Test geht der Beta-Fehler hervor. Beta-Fehler oder der Beta-Faktor werden in der Sicherheitstechnik genutzt um Abhängigkeiten zu beschreiben.

Weiter ist noch Andrei Andrejewitsch Markow zu nennen, dessen Modelle nicht nur für die Spracherkennung wichtig waren, sondern uns auch lehrten, wie man Übergänge von verschiedenen Zuständen quantifizieren kann.

Dieser geschichtliche Exkurs sollte zeigen, dass wir mit der Funktionssicherheit jetzt nicht die Welt neu erfinden, sondern versuchen technische Systeme zu beschreiben und zu analysieren. Hierzu werden Methoden in der Sicherheitstechnik genutzt und weiterentwickelt, die eine lange Historie haben.

3.2 Technische Zuverlässigkeit

Die ersten Untersuchungen zur technischen Zuverlässigkeit in Zusammenhang zum heutigen Mathematikbegriff begannen zu Beginn des industriellen Zeitalters. Eine vollständige Studie über die Lebensdauer eines Rollenlagers ist im Rahmen einer eisenbahntechnischen Entwicklung dokumentiert.

Das Gesetzt von Robert Lusser beschreibt eine Kette von Elementen, wobei sich die Gesamtzuverlässigkeit aus dem Produkt der einzelnen Zuverlässigkeiten ergibt. Dies beschreibt die Grundlage für die Zuverlässigkeit aller technischen Systeme. Im Grunde genommen sagt dieses Gesetz nichts anderes aus als: „Die Kette ist so stark wie ihr schwächstes Glied“. Auch sicherheitsrelevante Funktionen oder Sicherheitsmechanismen können nur so gut wirken wie die Einzelteile, aus denen sie zusammengesetzt sind. Daher ist die Zerlegung und Strukturierung von Wirkmechanismen die wesentliche Aufgabe der Fehler- beziehungsweise Sicherheitsanalyse. Die Identifizierung des Bedarfs zusätzlicher Mechanismen und mit welcher Intensität diese nun auf das System einwirken, gilt es zu analysieren. Welches dann die geeigneten Maßnahmen sind, um das System zuverlässiger und weniger wartungsanfällig zu machen, eine höhere Verfügbarkeit oder eine höhere Sicherheit zu erreichen, ist demnach das Ergebnis dieser Analyse und der entsprechenden Maßnahmen, um die Kette an den schwachen Stellen entsprechend zu stärken. Es gibt zwar zerstörende Analysen oder Analysen, bei denen ein Stimulus in das Produkt implementiert oder injiziert wird. Diese Analysen verändern das Produkt um den Zweck der Analyse. Aber das Grundprodukt wird durch die Analyse nicht verändert, erst die Maßnahmen, die ergänzt oder durch eine Modifikation (z.B. im

Rahmen eines Änderungsprozesses) am Produkt neue Verhalten oder veränderte Eigenschaften ausbilden, sind das Ziel einer Analyse. Bis ca. 1930 waren die Aktivitäten auf dem Gebiet der Zuverlässigkeit im Wesentlichen begrenzt auf mechanische Systeme. Der Schwerpunkt der Bestrebungen bei elektrischen Systemen bestand zunächst darin, elektrische Energiequellen sicher zu machen, das heißt, ihre Verfügbarkeit zu erhöhen. Parallele elektrische Schaltungen von Transformatoren und Übertragungseinheiten, also das Einbringen von Redundanzen, waren ein bedeutender Fortschritt in der elektrischen Zuverlässigkeit. Auch in der Luftfahrt entstanden erste Konzepte, die die technische Zuverlässigkeit betrachteten. So zum Beispiel, indem man durch Ermittlung und Auswertung von statistischen Daten das Ausfallverhalten verschiedener Flugzeugkomponenten betrachtete. Insbesondere durch Einbringen von Redundanzen wurde hier dann die Erhaltung der Funktionsfähigkeit gesehen. Der Begriff der technischen Verfügbarkeit und mögliche Maßnahmen zu deren Erhöhung wurden systematisch erarbeitet. Die im Grunde rein qualitative Wirkkettenanalyse wurde dann durch das Team rund um den Mathematiker Erich Pieruschka auch durch statistische Betrachtungen quantifizierbar. Er definierte folgende Grundsätze:

R1, R2, ... Rn seien die Überlebenswahrscheinlichkeiten der einzelnen Kettenglieder.

Da zur Funktion der Kette alle Glieder notwendig sind und die Überlebenswahrscheinlichkeit eines jeden einzelnen Kettengliedes voneinander unabhängig ist, berechnet sich gemäß den Regeln der Wahrscheinlichkeitstheorie die Überlebenswahrscheinlichkeit der gesamten Kette als Produkt der Einzelwahrscheinlichkeiten wie folgt:

Gesamte Überlebenswahrscheinlichkeit der Kette: $Rg = R1 \cdot R2 \cdot ... \cdot Rn$

Hieraus entstand die Erkenntnis, dass die Zuverlässigkeit der einzelnen Komponenten eines Systems um ein Vielfaches höher sein muss als die des Gesamtsystems. Es entstand eine neue, vorwiegend technisch ausgerichtete wissenschaftliche Disziplin: die Zuverlässigkeitstheorie. Diese beschäftigt sich mit der Messung, Vorhersage, Erhaltung und Optimierung der Zuverlässigkeit technischer Systeme.

In den 50er Jahren erlebte die Zuverlässigkeitstechnik in den Vereinigten Staaten von Amerika durch die wachsende Komplexität elektronischer Systeme insbesondere im militärischen Bereich einen rasanten Aufschwung. Die Analyse der Fehler und deren Ursachen sowie die Instandsetzung der defekten Komponenten wurden immer aufwendiger. Das US-Verteidigungsministerium gründete deshalb 1952 die Advisory Group on Reliability of Electronic Equipment (AGREE).

Untersuchungen ergaben, dass für elektronische Systeme das Doppelte der Beschaffungskosten jährlich für die Instandhaltung aufgebracht wurde. Hieraus wurde abgeleitet, dass die Zuverlässigkeitstechnik ein integraler Bestandteil der Entwick-

lung und Konstruktion sein muss. AGREE bestand darauf, dass neue Systeme und Bauteile vor ihrer Produktion erst umfangreichen Tests unter erschwerten Bedingungen (Temperatur, Spannungen, Vibration usw.) unterzogen werden müssen, um so Schwachpunkte in der Konstruktion zu entdecken und zu beheben. Weiter wurde empfohlen, dass ein mittlerer Ausfallabstand MTBF (Mean Time Between Failures) und dessen Vertrauensbereich zu berechnen ist. Es ist weiterhin nachzuweisen, dass der mittlere Ausfallabstand über dem geforderten Wert liegt.

Die Zielrichtungen der Zuverlässigkeitstechnik wurden auch für elektrische Bauelemente übernommen und von dort innerhalb von 20 Jahren praktisch auf alle technischen Fachsparten übertragen.

Die Notwendigkeit zur Steigerung der Zuverlässigkeit technischer Produkte ist heutzutage vor allem vorgegeben durch einen stärker gewordenen Wettbewerb und einen daraus resultierenden Marktdruck, der nicht mehr allein über den Preis reguliert werden kann. Der rasch steigende technologische Fortschritt führt zu kürzer werdenden Produktzyklen. Es verbleibt keine Zeit mehr für eine umfangreiche praktische Erprobung vor der Markteinführung. Durch den verstärkten Kostendruck werden kostengünstigere Entwicklungs- und Fertigungsverfahren verlangt, durch die jedoch die Qualität und Zuverlässigkeit der Produkte nicht beeinträchtigt werden darf. Dies alles erhöht die Risiken einer Produktentwicklung. Die Zuverlässigkeitstechnik liefert Methoden zur Risikobegrenzung in Konzeption, Entwicklung und Fertigung von technischen Produkten.

Die Zuverlässigkeit ist ein Aspekt technischer Unsicherheit. Die Zuverlässigkeitstheorie beschäftigt sich mit der Vorhersage, Messung, Optimierung und Erhaltung der Zuverlässigkeit technischer Systeme. Dies erfordert die Anwendung statistischer und wahrscheinlichkeitstheoretischer Methoden. Ob ein Produkt eine bestimmte Zeit funktionsfähig bleibt, ist nur als Wahrscheinlichkeit darstellbar und beherrschbar. Zuverlässigkeit und Qualität liegen begrifflich nah beieinander. Wesentlich in diesem Zusammenhang ist die Notwendigkeit, die gewünschte Funktion eines Produktes und damit die Funktionsfähigkeit an sich exakt zu definieren. Daneben ist Zuverlässigkeit nur dann quantitativ beschreibbar, wenn ein Zeitbezug hergestellt wird. Mit Hilfe der Zuverlässigkeitstechnik sollen Aussagen über das Systemverhalten während der Nutzungszeit gewonnen werden. Die Zuverlässigkeit eines technischen Produktes hängt jedoch nicht nur von der Nutzungszeit ab. Nutzungshäufigkeit und die Intensität der Nutzung sowie Nutzungsumfeld und -umwelt sind wesentliche Einflussgrößen auf die Zuverlässigkeit. Daher bildet auch hier die Umgebung und das Nutzungsprofil eine wesentliche Rolle für die Zuverlässigkeit.

3.2.1 Grundlage der Zuverlässigkeit

Zuverlässigkeit wird also allgemein als erwartete Funktionserfüllung über einen definierten Zeitraum beschrieben. Die Fehlereintrittserwartungszeit (MTBF, mean time between failure) ist klassisches Maß für die Zuverlässigkeit von Komponenten (zum Beispiel Bauelemente, Baugruppen, Geräte, Anlagen). Hier unterscheidet man zwischen reparierbaren und nicht reparierbaren Komponenten. Bei nicht reparierbaren Komponenten wird die Erstfehlerwahrscheinlichkeitszeit (MTTF, mean time to failure) definiert, über unterschiedliche Wartungsmodelle kann man dann wieder auf die MTBF schließen. Bei der MTTF wird der statistische Erwartungszeitraum eines Fehlers ermittelt. Gemäß IEC 60050 wird die MTTF wie folgt definiert: „Erwartungswert der Zeit bis zum Ausfall". Bei einer Lebensdauerverteilung mit konstanter Ausfallrate (meist einer Exponentialverteilung angenähert) ist der Kehrwert der MTTF die Ausfallrate (R). Diese wird in FIT (Failure in Time) mit der Einheit „Ausfall pro 10E-9 Stunden" angegeben und gilt auch in der Sicherheitstechnik als Maß für die Ausfallwahrscheinlichkeit von elektrischen Bauelementen.

MTTF = 1 / R

Typisch für die Automobilindustrie ist eine durchschnittliche statistische Lebenserwartung von 15 Jahren bei einer jährlichen Fahrleistung von 300 Stunden. Tritt bei einer Million Komponenten ein Fehler im Fahrzeugleben auf, so ergäbe sich folgende Fehlerrate:

MTTF bei kontinuierlicher Beanspruchung über Lebensdauer:

MTTF = 15 Jahre x 1 Million Teile / 1 Ausfall = 15.000.000 Jahre

MTTF bei Beanspruchung nur während Fahrzeit über Lebensdauer:

MTTF = 15 x 300h x 1 Million Teile / 1 Ausfall = 1.500.000.000 Stunden

Ausfallrate als Kehrwert der MTTF in FIT:

R = 1 / 1,5h ·10E-9 = 0,67 Fit.

Für die kontinuierliche Beanspruchung käme ein anderer Wert für die Ausfallrate heraus, da der Beanspruchungszeitraum wesentlich größer ist. Als Konsequenz sollte man hier sehen, dass auch die Ausfallrate nie ein eindeutiger Wert für eine Komponente sein kann. Einsatzart und Einsatzumgebung müssen auch für die Ermittlung der Werte berücksichtigt werden. Die Reparaturzeiten in den Modellen zu ergänzen hängt auch sehr von den Einsatzbedingungen ab. In der Automobilindustrie legt man grundsätzlich auf die erwartete Lebensdauer aus, was aber im Einzelfall eine enorme Herausforderung bedeuten kann. Ein Kabelbaum im Auto wird oft mit 6000Fit angegeben, somit ist er immer das schwächste Glied

in der Kette. Man neigt immer dazu hier das Optimierungspotential für die Kette zu suchen. Für eine quantitative Funktionsbetrachtung würde jedoch dieser hohe Anteil der Fehlerrate für dieses schwache Glied alle anderen Fehler in der Funktionskette dominieren, so dass die Quantifizierung an der Stelle nicht die gewünschten Effekte zeigt. In die Verbesserung der Kabelführung und auch die Verbesserung der Verbindungen wird sehr viel Energie hineingesteckt. Durch die heterogene Nutzung und Verlegung im Fahrzeug oder auch durch unsachgemäße Wartung kommt es immer zu Fehlern in den Kabelbäumen. Daher werden solche Verbindungen oft nur rein formal mit einem Fit in die Betrachtungen von Sicherheitsanwendungen einfließen. Diese Empfehlung geben auch die meisten Zuverlässigkeitshandbücher.

In der Realität gibt es keine konstante Fehlerrate über die gesamte Lebensdauer, weil es meistens keine kontinuierliche gleichförmige Beanspruchung gibt. Durch eine konservative Auslegung kann oft vermieden werden, dass Bauelemente über die Elastizitätsgrenzen beansprucht werden. Werden diese Grenzen jedoch häufig oder intensiv überschritten, ist ein statistisch verteiltes Alterungsverhalten nicht mehr argumentierbar. Kein Material weist eine konstante Alterungskurve auf und eine Materialstreuung je nach Beanspruchung führt auch zu Unterschieden im Alterungsverhalten. Dieser Umstand führte zu der Definition der Badewannenkurve, die in der Sicherheitstechnik meist als Referenzmodell betrachtet wird. Dies vereinfacht die Betrachtungen und deren Umfang wesentlich und gleicht Varianzen im Wesentlichen hinreichend besonders für elektrische Bauelemente aus.

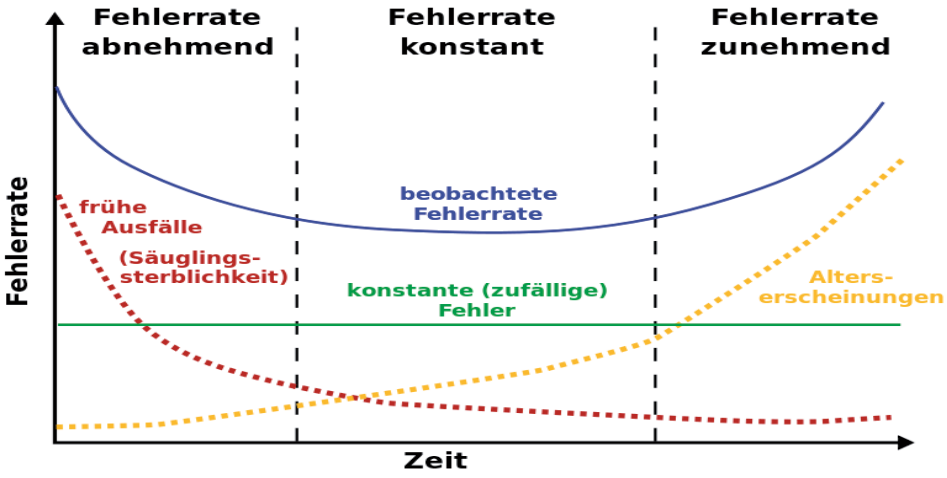

Bild 3.1 Badewannenkurve (Beispiel aus der Demografie)

Die Badewannenkurve zeigt drei Bereiche über die Zeit. Die Frühausfallphase beschreibt den Zeitraum, wo das Ausfallverhalten durch unbekannte Einflüsse, Umgebungsparameter, korrekte Materialien, Arbeitspunkte noch nicht hinreichend ausgereift ist. Dies sollte bei Komponentenentwicklungen so weit im Rahmen der Designverifikation untersucht sein, dass man beim Start der Serienfertigung in die Phase 2, die Nutzungsphase kommt. Die Nutzungsphase soll so ausgelegt sein, dass die Ausfallrate erst nach Ablauf der statistischen Lebenserwartung der Komponente beginnt. In der Praxis wird die Fehlerrate so weit unterhalb der Badewannenkurve platziert, dass man einen alterungsbedingten Anstieg zwar schon sieht, aber die Robustheit hinreichend gewählt ist, sodass die statistische Lebenserwartung erreicht wird. Die ISO 26262 selbst stellt keine Anforderungen um zum Beispiel Frühausfallverhalten zu vermeiden.

Die Umgebungsbedingungen versucht man mit den sogenannten Pi-Faktoren zu standardisieren bzw. anzupassen oder zu korrigieren.

Typischerweise orientieren sich die Pi-Faktoren an der Arrhenius-Gleichung.

$$k = A \cdot e^{\frac{-E_A}{R \cdot T}} \quad \text{Arrhenius - Gleichung}$$

A: präexponentieller Faktor oder Frequenzfaktor

EA: Aktivierungsenergie (Einheit: J · mol-1)

R: = 8,314 J · K-1 · mol-1 universelle Gaskonstante

T: absolute (thermodynamische) Temperatur (Einheit: K)

k: Reaktionsgeschwindigkeitskonstante

Besteht eine Temperaturabhängigkeit von A, wird diese Formel verwendet.

$$k = B \cdot T^n \cdot e^{\frac{-E_A}{R \cdot T}}$$

Da bereits die Badewannenkurve und die Materialabhängigkeit in solchen Gleichungen eine starke Abstrahierung der messbaren Ergebnisse von technischen Systemen darstellen, bezieht man sich insbesondere bei elektrischen Bauelementen auf anerkannte Tabellenbücher. Eines der am weitesten verbreiteten Tabellenbücher für die Zuverlässigkeit elektronischer Bauelemente ist die Siemensnorm SN 29500. Sie beschreibt einen einfacheren Ansatz zur Handhabung der Korrekturfaktoren, dieser Ansatz wurde später in die DIN EN 61709 übernommen.

> In DIN EN 61709 bzw. SN 29500ff. wird der temperaturbedingte Beschleunigungsfaktor π_T für 2 Ausfallmechanismen (z. B. bei diskreten Halbleiterbauelementen, IC's, optoelektronischen Bauelementen) wie folgt angegeben:

$$\pi_T = \frac{A \cdot EXP(E_{a1} \cdot Z) + (1-A) \cdot EXP(E_{a2} \cdot Z)}{A \cdot EXP(E_{a1} \cdot Z_{ref}) + (1-A) \cdot EXP(E_{a2} \cdot Z_{ref})}$$

Mit A = 1 und E_{a2} = 0 lässt sich die obige Beziehung auf das unter 3.3 beschriebene Basismodell für einen Ausfallmechanismus (z. B. bei Widerständen, Kondensatoren, Induktivitäten) zurückführen.

$$\pi_U = EXP\left\{C_1 \cdot \left(U^{C2} - U_{ref}^{c2}\right)\right\}$$

oder

$$\pi_U = EXP\left\{C_3 \cdot \left[\left(U / U_{rat}\right)^{c2} - \left(U_{ref} / U_{rat}\right)^{c2}\right]\right\}$$

Beanspruchungsfaktoren für Spannungsabhängigkeit π_U gem. DIN EN 61709 / SN 29500ff.

$$\pi_I = EXP\left\{C_4 \cdot \left[\left(I / I_{rat}\right)^{C5} - \left(I_{ref} / I_{rat}\right)^{C5}\right]\right\}$$

Beanspruchungsfaktoren für Stromabhängigkeit π_I gem. DIN EN 61709 / SN 29500ff.

Neben den Umfeldfaktoren spielt auch die Art der Fehlerverteilung und wie sie für die unterschiedlichen technischen Elemente statistisch beschrieben werden kann eine Rolle.

Am bekanntesten ist die Normalverteilung, auch Gaußverteilung. Die Gauß'sche Glockenkurve war früher auf den 10-DM-Scheinen abgebildet.

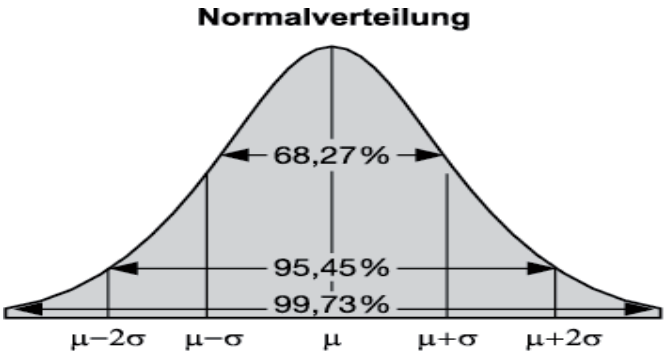

Bild 3.2 Normalverteilung oder Gauß'sche Glockenkurve

Oft wird der Wert von 6 Sigma (six sigma) insbesondere in der Produktionstechnik betrachtet. Bei 6 Sigma betrachtet man 3,4 Defekte pro einer Million Fehlermöglichkeiten, eine Fehlerwahrscheinlichkeit von 0,00034%, eine Fehlerfreiheit von 99,99966% im Bezugszeitraum oder auch die Prozessfähigkeit kurzfristig von Cpk=2 oder langfristig von Cpk=1,5.

Bei Zählbarem, basierend auf natürlichen Zahlen von elektrischen Bauteilen, spricht man oft von einer Chi-Quadrat-Verteilung, weiter werden auch oft Binomialverteilungen, logarithmische oder Weibullverteilungen für die Fehlerwahrscheinlichkeit betrachtet.

In der Automobilindustrie wendet man für komplexere Bauelemente die AEC (Q) 100 an, dies ist ein Standard zur Qualifikation von elektrische Bauelemente. Einfache Bauelemente wie Widerstand und Kondensatoren werden in diesem Standard nicht adressiert. Da diese einfachen Bauelemente meist durch die Vielzahl der Elemente jede statistische Grenze sprengen würden, sind solche statistischen Betrachtungen für die Sicherheitstechnik nicht hinreichend. Das Risiko bei einfachen Bauelementen besteht darin, dass unentdeckt schadhafte Bauelemente zur Produktion geliefert werden. Daher wird die Eignung und ob die Bauelemente in ihrem Einsatzfall tatsächlich hinreichend dimensioniert werden im Rahmen der Qualifikation der gesamten elektronischen Baugruppe durchgeführt. Der Wert für die Fehlerraten nimmt man aus Tabellenbücher. Wobei man bei korrekter Qualifikation inklusive des Nachweis der Lebensdauertauglichkeit der gesamten elektronischen Baugruppe annimmt, dass die einfachen Bauelemente in der in der konstanten Phase der Fehlerraten der Badewannenkurve liegen.

3.2.2 Zuverlässigkeit und Sicherheit

Zuverlässigkeit wird allgemein als eine Komponenteneigenschaft in der Literatur beschrieben, im Gegensatz dazu wird Sicherheit als eine Systemeigenschaft gesehen.

Grundsätzlich gilt der Ansatz, dass Zuverlässigkeit eine Komponenteneigenschaft nur dann, wenn die Umgebungsbedingungen eindeutig definiert sind. Hier stellt sich die Frage, ob nicht für Komponenten in komplexen dynamischen Systemen bezüglich Zuverlässigkeit, dieselben Herausforderungen gelten wie für die Sicherheit.

Als Erstes ergibt sich die Frage, ob man eine Komponentenumgebung vollständig spezifizieren kann. Bei vielen, insbesondere rein mechanischen Komponenten kann man von normierten immer gleichbleibenden Umgebungsbedingungen ausgehen. Betrachtet man jedoch die Zuverlässigkeit über die Zeit, so wird man bereits Einflussfaktoren finden, die nur sehr schwer spezifizierbar sind oder nur meist aus negativer Erfahrung überhaupt als Einflussgröße bekannt sind. Dies gilt für das Thema Materialverträg-

lichkeit bei den Werkstoffen Kupfer / Zink oder Edelstahl und Salzatmosphäre, die ab einer bestimmten Konzentration zu galvanischen Elementen werden können und somit zum Beispiel zu Korrosion führen. Weiter kennt man bei Berührung, Schlägen und Reibung den Effekt, dass unterschiedliche Härten, Oberflächenbeschaffenheiten und Materialkombinationen zu minder starken oder schwachen Materialabnutzungen bis hin zu Rissen in den Materialien führen. Ebenso spielt die Intensität oder der Impuls, mit dem Komponenten miteinander interagieren, eine große Rolle für deren Lebensdauerzuverlässigkeit. Es gibt Materialien, die sehen einen Stoß oder Schlag mit einer bestimmten Kraft (auch eine bestimmte Anzahl pro Zeiteinheit) als elastischen Stoß an, so dass es keine signifikanten Alterungseffekte gibt (sprich das Material oder seine Struktur ist vor dem Stoß und nach dem Stoß unverändert). Oder es gibt bestimmte Veränderungen an den bei der Interaktion beteiligten Materialien. Dies kann auch noch von Schmutz, Feuchtigkeit oder anderen chemischen Stoffen abhängig sein. Als signifikantes Beispiel wird der Vergleich einer Kraft, die hydraulisch auf eine Komponente einwirkt, oft als weicher Impuls angesehen, da die Hydraulikflüssigkeit selber abfedert und der Druckaufbau der Hydraulik meist recht träge verläuft. Wird die Kraft jedoch rein mechanisch, womöglich auch noch basierend auf einer elektromotorischen Kraft erzeugt, so wird der Impuls für die Komponenten wesentlich härter wirken können. Dies kann für Festigkeitsanforderungen bis hin zur Lebensdauerzuverlässigkeit einen wesentlichen Einfluss bedeuten.

Für die Funktionssicherheit gibt es verschiedene Anknüpfungspunkte, wo sich diese beiden Themen überlappen. Hierzu zählen alle externen oder äußeren Schnittstellen und die Komponentenschnittstellen.

Bei der Definition des Fahrzeugsystems (ITEM Definition, ISO 26262, Teil 3, Kapitel 5) werden bereits externe Maßnahmen, Umgebungsbedingungen, Verhalten mit externen Fahrzeugsystemen, Betriebsbedingungen, dynamisches Verhalten und so weiter spezifiziert. Das heißt, bereits bei den Zielen für die Funktionalität und den technischen Schnittstellen zum Fahrzeug müssen wesentliche Einflussfaktoren für Zuverlässigkeit und Sicherheit betrachtet werden. Diese Parameter gehen schon als Input in die Gefahren- und Risikoanalyse ein. Hier wird es bei der Betrachtung der potentiellen Fehlfunktionen, die dann zu einer Gefährdung führen, sehr unterschiedliche Ergebnisse geben insbesondere für den Parameter S (Schweregrad) und den Parameter C (Beherrschbarkeit durch den Fahrer (oder andere betroffene Personen). Ein prägnantes Beispiel hierzu ist die Sollbruchstelle im Getriebe, diese soll ein Blockieren der Antriebsräder verhindern, wenn ein Fehler des Getriebes den Antriebsstrang blockiert. Dieses Abreißen darf natürlich nicht bei entsprechender Last passieren und muss auch über den gesamten Nutzungsraum und die Lebensdauer gewährleistet sein. Bei modernen Getrieben werden die Schaltzeiten immer kürzer, um Energieverluste zu minimieren und

auch bessere Beschleunigungswerte zu erzielen. Dadurch werden die Gänge natürlich auch härter eingelegt, das heißt, der Impuls und auch die Energie, die aufgebracht wird, ist wesentlich höher. Als Konsequenz dazu kann man diese Sollbruchstelle nicht mehr über Lebensdauer auslegen und man ist gezwungen eine E/E-Maßnahme gegen das Blockieren des Antriebsstranges einzuführen. Diese Maßnahme kann wegen der stabilisierenden Wirkung insbesondere der Hinterachse schnell zu einem recht hohen ASIL führen.

In der Systementwicklung wird man weitgehend sinnvollerweise versuchen realisierungsneutral das Design zu beschreiben, sodass die Zuverlässigkeit in erster Linie wieder bei der Komponentenrealisierung relevant wird. Im Software-Design wird oft über Zuverlässigkeit diskutiert, aber zu einer systematischen Methode, um die Zuverlässigkeit von Software zu bestimmen, wurden in der ISO 26262 keine konkreten Anforderungen formuliert. Anders sieht es bei Mechanikkomponenten aus, hier gelten weitgehend die Aussagen, die auch bereits für das Fahrzeugsystem und dessen Integration ins Fahrzeug formuliert wurden.

In der Elektronik gibt es wieder sehr enge Schnittstellen, insbesondere daher, dass die Architekturmetriken (ISO 26262, Teil 5, Kapitel 8) sowie die Topfehlermetrik (ISO 26262, Teil 5, Kapitel 9) auf der Ausfallwahrscheinlichkeit von elektrischen Bauelementen oder der Auftretenswahrscheinlichkeit von zufälligen Hardwarefehlern beruhen. Auf diese quantitativen Sicherheitsanalysen wird im Kapitel 4.4.2.5 detaillierter eingegangen. Oft wird das Kapitel 7 im Teil 5 der ISO 26262 übersehen, wo es um die korrekte und sicherheitsgerechte Auslegung des EE-Hardware-Design und dessen Verifikation geht. Hier werden auf Basis des Systemdesign und den Sicherheitsanforderungen, die aus dem Systemdesign auf die Elektronik heruntergebrochen werden, die entsprechenden Elektronik-Sicherheitsanforderungen und das Elektronikdesign abgeleitet. Natürlich hat ein Widerstand bei einer normierten Umgebung eine bestimmte Zuverlässigkeit. Spiegelt jedoch die Basis aus möglichen Tabellenbüchern tatsächlich die wirkliche Umgebung der Bauelemente wieder? Wenn man keine implementierten Sicherheitsmechanismen betrachtet, werden für die einzelnen Bauelemente reine Zuverlässigkeitsprognosen im Rahmen der Designverifikation vorgenommen. Diese Werte stellen auch die Basis für entsprechende quantitative Metriken der Funktionssicherheit dar. Ein weiterer Aspekt wird hier die Wahrscheinlichkeit sein, mit der sich bestimmte Fehler im Design fortpflanzen. Bei einer rein funktionalen Betrachtung wird es meist keine Hinweise auf solche Anhängigkeiten geben. Betrachtet man jedoch die Realisierung, so werden Größe wie elektrischer Widerstand, Abstande auf der Platine, Durchmesser von Leiterbahnen oder Steckerpins, Materialien und Materialverträglichkeit, Wärmeleitfähigkeit und so weiter, maßgeblichen Einfluss auf die Zuverlässigkeit und unter Umständen Sicherheit der Produkte aufweisen

können. Betrachtet man gefährliche Wärmeentwicklungen bis hin zum Brand als potentielle Fehlfunktion der zu realisierenden Elektronik, so wird die Auslegung der Elektronik zu einer wichtigen Sicherheitsmaßnahme. Im Wesentlichen bilden sich hieraus auch die Designgrenzen für das Gesamtprodukt heraus, das heißt, die Sicherheitsschwellen werden einen signifikanten Einfluss auf die mögliche Performance des Gesamtproduktes haben können.

3.3 Architekturentwicklung

Architektur wird oft als das Rückgrat eines jeden Produktes gesehen. Die ISO 26262, Teil 1 Kapitel 1.3 beschreibt die Architektur als Repräsentation eines Fahrzeugsystems, von Funktionen, Systemen oder Elementen, die durch Bausteine, deren Abgrenzungen, Schnittstellen und deren Zuordnung zu Elektronikhardware und Software identifizieren werden können. Als Grundlage für die Definition des Fahrzeugsystems wird bereits das Funktionskonzept (ISO 26262, Teil 1 Kapitel 1.50) genannt. Das Funktionskonzept bilden gemäß Glossar die Spezifikation der beabsichtigten Funktionen und deren Interaktionen, um das beabsichtigte Verhalten zu erreichen. Damit sehen wir zwei Aspekte, die die Architektur erfüllen muss. Sie gibt die Produktstruktur vor und damit auch alle Schnittstellen sowie die Grundlage für die Darstellung des technischen Verhaltens. Für jeden Baustein oder jedes Element und deren Schnittstellen werden Anforderungen benötigt. Das beabsichtigte Verhalten als auch das Verhalten im Fehlerfall muss spezifiziert sein. Somit ist man gezwungen, sämtliche Abstraktionsebenen, Perspektiven, Schnittstellen sowie deren gewünschtes technisches Verhalten zu planen und im Voraus zu definieren. Ursprünglich wurde der Begriff der Sicherheitsarchitektur als weiterer Begriff zur Architektur in der ISO 26262 definiert. Hier konnte man sich aber nicht auf eine eindeutige Abgrenzung zur Produktarchitektur einigen. Insbesondere die Schnittstellen im Produkt müssen für den sicherheitsrelevanten Teil wie auch für alle anderen Teile des Produktes konsistent definiert sein. Weiter gab es auch Argumente, dass, wenn man von Architektur spricht, immer die Sicherheitsarchitektur gemeint ist. Dies kann man mit der Definition des Funktionalen Konzeptes jedoch nicht übereinbringen, da sonst alle Teile und damit alle Eigenschaften, die zur Realisierung des sicherheitsrelevanten Produktes notwendig sind, automatisch sicherheitsrelevant sind. Dies gilt es grundsätzlich zu vermeiden. Sicherheitsmechanismen und sicherheitsrelevante Funktionen sollten einfach und eindeutig beschreibbar sein, auch wenn die zu realisierenden Funktionen und ihre Zusammenhänge sehr komplex werden können.

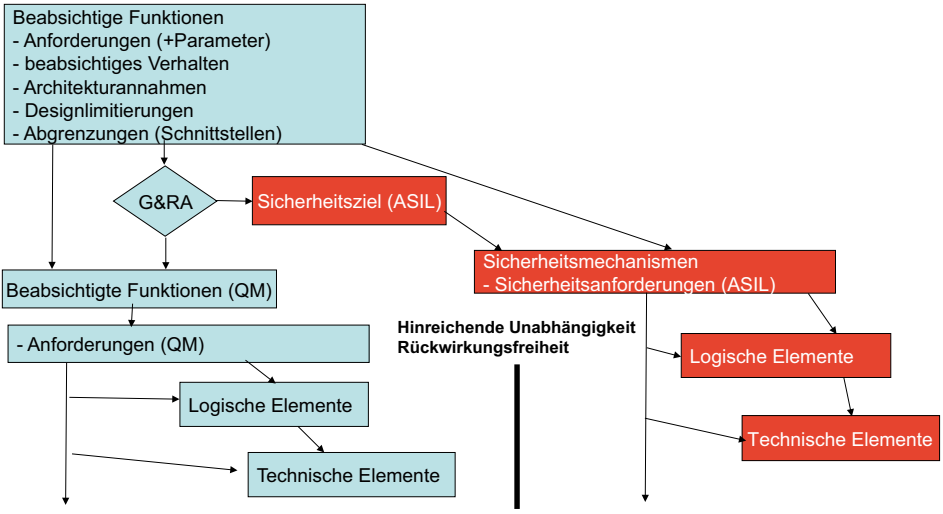

Bild 3.3 Trennung von beabsichtigter Funktion und Sicherheitsmechanismen

Das Bild 3.3 zeigt, wie man eine solche Struktur planen kann, jedoch zeigt es auch, wie stark die Anforderungen aus der Definition des Fahrzeugsystems durch alle Elemente der Architektur vererbt werden. Wird zum Beispiel die beabsichtigte nicht-sicherheitsrelevante Funktion (QM) im selben technischen Element (zum Beispiel Mikrokontroller) umgesetzt, so kann jede Eigenschaft des Mikrokontrollers auch die sicherheitsrelevante Funktion beeinflussen. Daher wurde in älteren Sicherheitsnormen darauf verwiesen, dass alle Funktionen in einem Mikrokontroller gemäß der höchsten Sicherheitsintegrität (hier ASIL) realisiert werden müssen. Bei Systemen mit nur einem Sicherheitsziel war dies durch separate Mikrokontroller umsetzbar, bei mehreren Sicherheitszielen mit verschiedenen Ebenen der Sicherheitsintegrität oder ASILs und verschiedenen sicheren Zuständen wird dies schwierig. Als Konsequenz daraus kommt man nicht umhin, die gesamte Produktarchitektur in der jeweiligen Integrationsumgebung vollständig zu analysieren. Die ISO 26262 lässt auch zu, dass ein Fahrzeugsystem aus mehreren Systemen zusammengesetzt wird. Sollten in dem Fall die Schnittstellen der einzelnen Systeme nicht aufeinander abgestimmt sein, so wird man bei der Integration diese anpassen müssen, da sonst keine systematische Abstimmung der Schnittstellen stattfinden kann. Das heißt, wenn man nicht die Schnittstellen vorab plant, werden sich die Architekturen der verschiedenen Systeme ihre eigenen Schnittstellen festlegen. Es wäre reiner Zufall, wenn diese zu den beteiligten Systemen passen würden oder gemeinsam zu den Schnittstellen des Fahrzeugs, in das das jeweilige System integriert werden soll. Auf der Fahrzeugebene kann man ein elektronisches System inklusive der physischen Erkennung der Sensoren oder der Fahrzeugreaktion aufgrund der Ansteuerung durch einen Aktuator sehen, oder

man betrachtet gemäß ISO 26262 nur die elektrisch oder elektronisch umgesetzten Funktionen. Genauso verhält es sich mit der Software, man kann die Software als eine Komponente in einem Mikrokontroller beschreiben. Alternativ kann man das funktionale Verhalten inklusive Mikrokontroller als Systemfunktion beschreiben und beschreibt die Software oder mehrere Softwarekomponenten und der Mikrokontroller als zwei oder mehrere Komponenten aus denen das System besteht.

Diese Festlegungen führen jedoch dazu, dass per Definition ausgeschlossenen technische Lösungen, wenn sie fälschlicherweise doch umgesetzt werden, zu neuen technischen Risiken führen. Verwendet man in einem Mikrokontroller keine Interrupts, so gibt es auch kein Risiko, dass diese Fehler verursachen können. In einem alten Käfer gab es keine Elektronik außer dem Transistorradio, also keine sicherheitstechnischen Risiken, deren Ursache Fehler in der Elektronik waren.

3.3.1 Stakeholder von Architekturen

Welchen Zweck erfüllt eine Architektur und welche Personen, Gruppen oder, um bei Prozessbeschreibungen zu bleiben, welche Rollen braucht eine Architektur?

Was ist der Unterschied zu Stakeholder eines Systems oder des Produktes?

Allgemein sollten hier Stakeholder des Produktes stehen, aber wir begrenzen uns hier auf die Architektur.

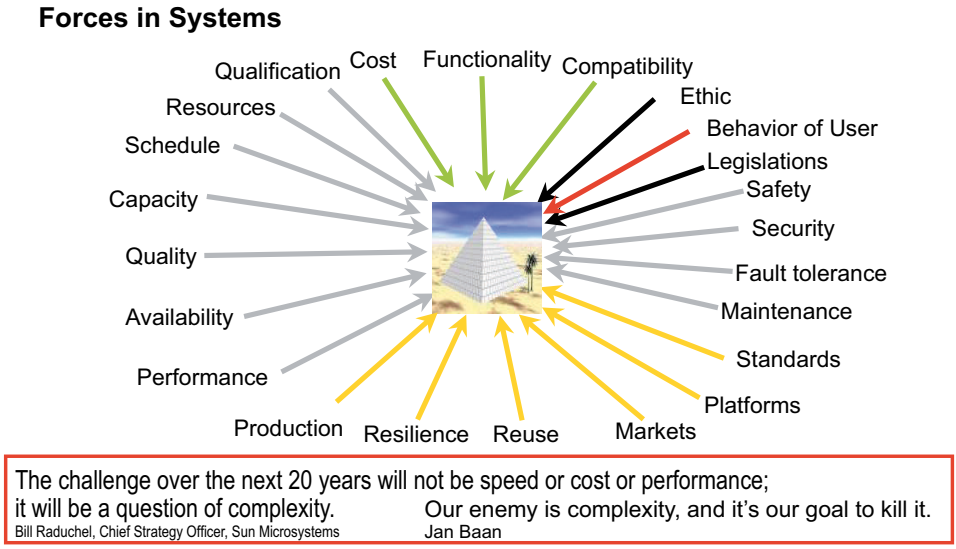

Bild 3.4 Treiber technischer Architekturen (Quelle: IBM)

Das Bild 3.4 zeigt die Treiber einer Architektur, das heißt, aus all diesen Aspekten können sich Anforderungen an ein Produkt ergeben. Hier gilt es natürlich in einer sehr frühen Phase in einer Produktentwicklung diese Treiber sowie im negativen Sinne die Risiken für ein Produkt zu identifizieren.

Bei den Kosten wird es sehr schnell transparent, aber wir werden sehen, dass fast alle Treiber in den verschiedenen Entwicklungen analoge Einflüsse und Risiken in sich bergen können.

Für eine Entwicklung benötigt man Geld, für Entwicklungsressourcen, Werkzeuge, Laboreinrichtung, Produktionsmittel und so weiter. Selbst die großen Erfinder im Automobilbereich, wie Benz, Diesel, Otto und so weiter, kannten das Problem. Das Geld sorgt dafür, dass bestimmte Eigenschaften knapp an der Auslegungsgrenze definiert werden, dass man nach günstigen und einfachen Lösungen sucht oder dass Entwicklungen gar eingestellt werden müssen, weil das Geld fehlt.

Dies führte dazu, dass man sehr sensible Kosten-Nutzen-Rechnungen erstellt, bevor jemand in eine Produktentwicklung einsteigt. Selbst einzelne Eigenschaften werden heute per Wertanalyse betrachtet und man untersucht, welche Eigenschaft für welche Zielgruppe ein Grundbedürfnis oder einen Begeisterungsfaktor darstellt.

Entsprechende Beispiele können nun für diese Treiber betrachtet werden, es würde jedoch wahrscheinlich nur zu folgendem Fazit führen: Je nach Markt und Anforderungen dieser Architekturtreiber wird es Produkteigenschaften geben, welche ein Grundbedürfnis für die Akzeptanz des Produktes darstellen, und andere, die sogar so viel Begeisterung hervorrufen, dass das Produkt auch ein Erfolg in verschiedenerlei Hinsicht werden kann. Im Umkehrschluss ist jede Nichterfüllung ein Risiko.

Somit wird man nie eine Architektur für eine Produktentwicklung rein nach Sicherheitsanforderungen entwickeln sondern man wird auch die anderen Architekturtreiber betrachten.

Das heißt, die Elemente, aus denen ein System zusammengesetzt wird, werden nicht nach reinen Sicherheitsaspekten definiert sein.

Doch klar, das Geld ist wichtig. Verfügbarkeit von Materialien (Stichwort seltene Erden), Produktionskapazitäten, Know-how, Erfahrung, Transportwege, Lieferkette und so weiter werden jedoch eine entscheidende Rolle spielen, wie die Elemente definiert und welche Position diese im System haben werden.

Die ISO 26262 adressiert natürlich nur elektrische und elektronische sowie Software-Elemente, jedoch wird ein Kondensator und ein Widerstand alleine noch keine Tiefpassfunktion realisieren können. Die in der Norm adressierten Elemente anderer Technologien spielen immer eine Rolle. Spätestens bei der Analyse der Fehlerabhängigkeit (Analysis of Dependent Failure oder der bekanntere Aspekt, die Common

Cause Analyse) wird man sehen, dass Stecker, Leiterplatten, Gehäuse einen großen Einfluss auf die Sicherheit haben können.

In der Automobilindustrie stellen Gehäuse immer noch eine Herausforderung an das Projektmanagement dar. Nicht nur, dass wir diese bereits zu Projektbeginn bestellen müssen, da die Gehäuse zu den sogenannten Langläufern gehören und damit den Bauraum für die Steuergräte festlegen, sondern auch die Anordnung von Platinen und Stecker muss dann bereits definiert sein. Was hat dies mit Sicherheit zu tun?

Die Metriken der ISO 26262 betrachten nur zufällige HW-Fehler, es gibt zwar Stimmen, dass Leiterbahnen, Lötungen, Kabel und Stecker auch zufällige HW-Fehler haben können, jedoch werden wir sehen, dass dies nicht das vordergründige Problem ist. Meist sind es die systematischen Fehler, sprich die potentiellen Auslegungsfehler, die uns hier vor besondere Aufgaben stellen. Stecker und Leiterbahnen müssen bestimmte Durchmesser haben, um bestimmte Ströme tragen zu können. Weiter sind die Abstände wichtig, besonders bei Spannungen über 60 Volt werden wir ganz neue Sicherheitsaspekte betrachten müssen. Die ISO 26262 verlangt zwar kein Derating (konservative Auslegung, sprich die Betriebskennwerte liegen deutlich unterhalb der Nominalwerte der Bauelemente) wie die IEC 61508 (70 %), jedoch müssen die Eigenschaften hinreichend robust über die gesamte Lebensdauer ausgelegt sein. Dadurch werden eventuell Steckerabstände, Pingröße, Leiterbahnenabstände, Dicke und so weiter durch die Auslegung der Sicherheitsmechanismen bestimmt. Dies kann im Einzelfall zu Raumknappheit im Gehäuse führen. Dadurch gibt es in der Praxis ein weiteres Problem, es fließt kein (außer Supraleitung) Strom ohne Verlustleistung. In jeder elektrischen Komponente entsteht Wärme, die nach außen abgeführt werden muss. Hier spielt die Wärmeleitfähigkeit des Gehäuses eine entscheidende Rolle. Überhitzung ist eine wesentliche Ursache für den Brand von Steuergeräten. Dies wird explizit von der ISO 26262 mit betrachtet, da es sich hier um eine Fehlfunktion der Elektronik handeln kann. Das Thema Wärme hat noch einen massiven Aspekt für einen typischen Projektlangläufer; den Mikrokontroller. Je höher man einen Mikrokontroller taktet, desto wärmer wird er. Die Anzahl der Operationen pro Zeiteinheit hat auch Einfluss auf die Erwärmung. Sprich, ein richtig an die Grenze dimensionierter Mikrokontroller wird sehr warm. Kann man die Wärme schnell abführen, so kann man die Grenzen besser ausreizen, sind aber auch andere Bauelemente knapp ausgelegt oder zu nah aneinander angeordnet, riskieren wir einen Wärmestau.

Das zeigt, dass viele Faktoren einen Einfluss auf die Eigenschaften des Produktes haben können. Neben der Komplexität der Anhängigkeit aus dem obigen Beispiel sehen wir, dass der Faktor Zeit einen gewichtigen Einfluss haben kann. Ein Gehäuse oder einen Mikrokontroller wechselt man nicht ohne gewichtigen Grund während der Entwicklung eines Produktes.

Das heißt aber, dass wir eine große Abhängigkeit zwischen Projektmanagement und Architektur haben.

In dem Buch „Engineering a Safer World“ von Nancy Leweson gibt es einen Vorschlag, der die Struktur für die verschiedenen Architektursichten und die Zuordnung der Anforderungen beschreibt. Die gesamten Zusammenhänge werden unter dem Begriff „Intent Specification“ dargestellt.

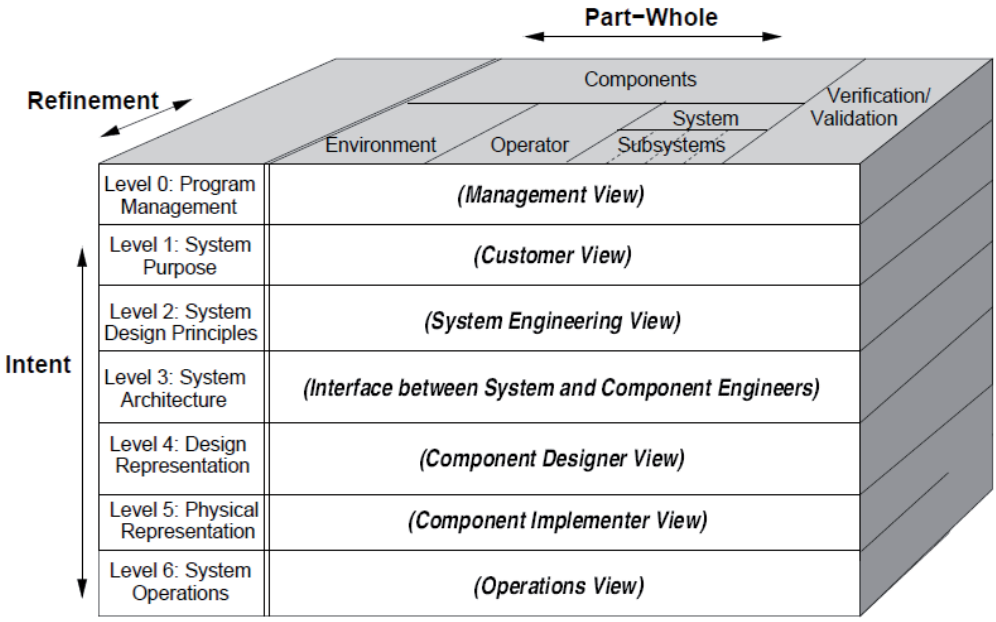

Bild 3.5 Struktur einer Spezifikation (Quelle: Figure:10.1, Nancy G. Leweson Engineering a Safer World)

Im Deutschen würde man wohl das Wort „intent“ mit Absicht oder Vorhaben übersetzen, somit würde die Würfelunterschrift „Struktur einer Vorhabensspezifikation“ heißen.

Die Kernaussage beruht darauf, dass die Produktstruktur, die Struktur der Organisation, die das Produkt erstellen soll, sowie die Managementstruktur aufeinander abgestimmt sein müssen. Damit die entsprechenden Organisationseinheiten miteinander arbeiten können, ist es notwendig, dass diese Struktur auch die Grundlage für die Spezifikationen darstellt.

Somit wird die Produktarchitektur in erster Linie die Grundlage für die Projektstruktur.

Als Konsequenz, wird es der erste Schritt einer Projektplanung sein, einen Projektstrukturbaum zu erstellen, der folgende Aspekte berücksichtigen muss:

- Produkt-, Organisations-, und Projektschnittstellen müssen miteinander harmonisiert werden. Je mehr Überschneidungen es zu den drei Schnittstellenklassen gibt, desto komplexer wird die Entwicklung des Produktes.
- Produkt-, Organisations- und Projektschnittstellen müssen definiert werden und durch Hierarchisierung gesteuert werden. Jede Schnittstelle muss in einer darüber liegenden Ebene definiert und gemanagt werden.
- Die Produktstruktur und deren horizontale und vertikale Schnittstellen bilden die Basis für die Spezifikation der Elemente der Architektur und deren Verhalten beziehungsweise die Abhängigkeit untereinander.

Die Würfelstruktur von Nancy G. Leveson wird hier in diesem Buch nicht weiterverfolgt, wobei auf Basis der Vorschläge eine Würfelsicht erzeugt werden könnte. Es werden nur die produkttechnischen Sichten weiter betrachtet, nicht die organisatorischen zu Kunden oder Zulieferern und so weiter. Diese Sichten müssen natürlich bei der Produktplanung, Projektplanung und Festlegung der Organisationsschnittstellen berücksichtigt werden. Diese Aspekte bilden jedoch die Grundlage für die Planung der Sicherheitsaktivitäten im Projektsicherheitsplan als Ableitung des Sicherheitslebenszyklus. (vergleiche ISO 26262, Teil 2, Kapitel 6.4.3, „Planning and coordination of the safety activities“).

3.3.2 Sichten einer Architektur

Da es unterschiedliche Stakeholder einer Architektur gibt, muss es auch unterschiedliche Abstrahierungen der Beschreibung für den jeweiligen Stakeholder geben. Optimal wäre, wenn man je nach Profil des Stakeholders bestimmte Informationen aus dem Gesamtbeschreibungsmodell entnehmen könnte. Um dies dann vollständig umsetzen zu können, müsste man neben einer Standardisierung der Stakeholder und den jeweiligen Interessen dieser oft sehr unterschiedlichen Personen auch ein Basisdatenmodell haben, welches in der Lage wäre, die ganze Welt mit ihren Zusammenhängen beinhalten zu können. So lange wird niemand warten können. Selbst solche genialen Datenmanagement- oder Informationssysteme wie Google, Wikipedia und so weiter würden hier an ihre Grenzen stoßen.

Jeder, der sich mit einem Hausbau beschäftigt hat, kennt eine Bauzeichnung. Das Ziel dieser Bauzeichnung ist natürlich dem späteren Eigentümer zu zeigen, wie das Haus fertig aussehen soll. Meist gibt es besonders bei Häusern, die durch einen

Bauträger gebaut werden, auch eine Baubeschreibung, aber die kann man ohne einen Anwalt nicht lesen.

Die Bauzeichnung zeigt meist eine Front-, Rück- und verschiedene Seitenansichten und Schnitte in der Vertikalen, damit man die Stockwerkaufteilung erkennt, sowie Schnitte in der Horizontalen, damit man zum Beispiel die Anordnung von Türen und so weiter sehen kann. Wir schauen aber immer auf dasselbe oder das gleiche Haus. Unsere Erwartung ist natürlich, dass die Sichten konsistent sind und wir eine Haustür in der Frontansicht an derselben Stelle im Haus sehen können, wie wir sie aus einem Horizontalschnitt erwarten würden.

Trotzdem müssen unterschiedliche Stakeholder einer Architektur identifiziert werden und diese wollen natürlich auch nur das in der Architektur sehen, was sie interessiert. Das heißt, wenn man einem Schreiner für die Innentüren die Bauzeichnung sendet, wird ihn die Höhe des Estrichs interessieren, jedoch nicht die Auslegung der Türstürze und wie viel Eisen für welche Traglast verwendet wurde. Grundsätzlich müsste man hier auch auf die Perspektive des Controllers und des Projektleiters eingehen, da die eingesetzten Ressourcen schon über die hinreichende Sicherheit entscheiden.

Phillipe Kruchten hat bereits Ende der 60er Jahre seine vier Sichten beschrieben, die (hier verglichen mit UML) zu folgenden 4+1 Sichten führten:

- Die logische Sicht („Logical View“) beschreibt die Funktionalität des Systems für den Endnutzer. Es werden logische Elemente genutzt, um unterschiedliche Abhängigkeiten der Elemente darzustellen. Als UML-Diagramme können Klassendiagramm, Kommunikationsdiagramm, Sequenzdiagramm verwendet werden.
- Die Entwicklungssicht oder Implementierungssicht („Development View“) beschreibt das System vom Standpunkt eines Entwicklers. Als UML-Diagramme können Komponentendiagramm oder Paketdiagramm verwendet werden.
- Prozesssicht (Verhalten oder funktionale Sicht, „Process View“) beschreibt die dynamischen Aspekte des Systems. Das Verhalten der Elemente an ihren Schnittstellen zueinander, in einer definierten Umgebung, wird hier beschrieben. Beziehungen können jegliche Art der Kommunikation (technisch, aber auch Mensch-Maschine etc.), zeitliches Verhalten sowie Allokations- und Strukturaspekte, wie Parallelität, Verteilung, Integration, Performanz und Skalierbarkeit sein. Als UML-Diagramme können Aktivitäten-, Sequenz- oder Timing-Diagramme verwendet werden.
- Die physikalische Sicht („Physical view“) oder Realisierungssicht („Deployment View“) beschreibt das System aus Sicht der Realisierung beziehungsweise des Planers der Realisierung. Hier soll man die Zuordnung der Komponenten, Module oder elektrischen Bauelemente und der Elemente, die zur Kommunikation (zum Beispiel Kabel, Bus, Stecker) untereinander realisiert oder beschafft wer-

den müssen, vorfinden. Als UML-Diagramme können Verteilungsdiagramme verwendet werden.-

- Die Szenariensicht („Scenario view“) beschreibt die geplanten Anwendungsfälle, mögliche Konfigurationen, auch Verhaltensvarianten. Dies kann die Grundlage für das geplante Verhalten der Elemente untereinander sein. Die Architekturverifikation bildet später die Grundlage für die Integrationstests. Als UML-Diagramme können Anwendungsfalldiagramme (Use-Case-Diagram) verwendet werden.

Im Rahmen des Förderprojektes „Safe“ wurden aus Definitionen des Projektes SPES2020 für die Automobilindustrie Sichten (siehe Bild 3.6) beziehungsweise Perspektiven abgeleitet.

Die einzelnen Perspektiven können wie folgt beschrieben werden:

- Die Bedienerperspektive stellt die Verhaltensschnittstelle zwischen Menschen und technischen Systemen und deren Elemente dar.
- Die funktionale Perspektive stellt das beobachtbare technische Verhalten dar.
- Die Variantenperspektive beschreibt die Abhängigkeit oder Unterschiede von verschiedenen Ausprägungen oder Umsetzungen aus Sicht des jeweiligen Adressaten (Stakeholder) des Systems oder dessen Elemente. Wobei ein solcher Stakeholder auch ein System oder Element sein kann.
- Die logische Perspektive nutzt logische Elemente, um Schnittstellen darzustellen oder Verhalten an Schnittstellen zu beschreiben.
- Die technische Perspektive nutzt technische Elemente, um Strukturen und Schnittstellen darzustellen oder Verhalten an Schnittstellen zu beschreiben.
- Die geometrische Perspektive zeigt die Position des Systems oder dessen Elemente in einem bestimmten Kontext oder einer Umgebung.
- Die Sicherheitsperspektive zeigt die sicherheitsrelevanten Aspekte einer Architektur.

3.3.3 Horizontale Abstraktionsebene

Abstraktion wird meist umschrieben als Weglassen von Einzelheiten und das Überführen auf etwas Allgemeineres oder Einfacheres. Als horizontale Abstraktionsebene wird hier die Tiefe bezeichnet, in die ich praktisch in ein Auto hineinsehe. Die Floskel „man sieht vor lauter Bäumen den Wald nicht mehr“ wird bei der Entwicklung von Fahrzeugfunktionen zu einer treffenden Umschreibung der Herausforderung. Wenn man das Verhalten eines Fahrzeugs beschreiben will, wird es sicher Abhängigkei-

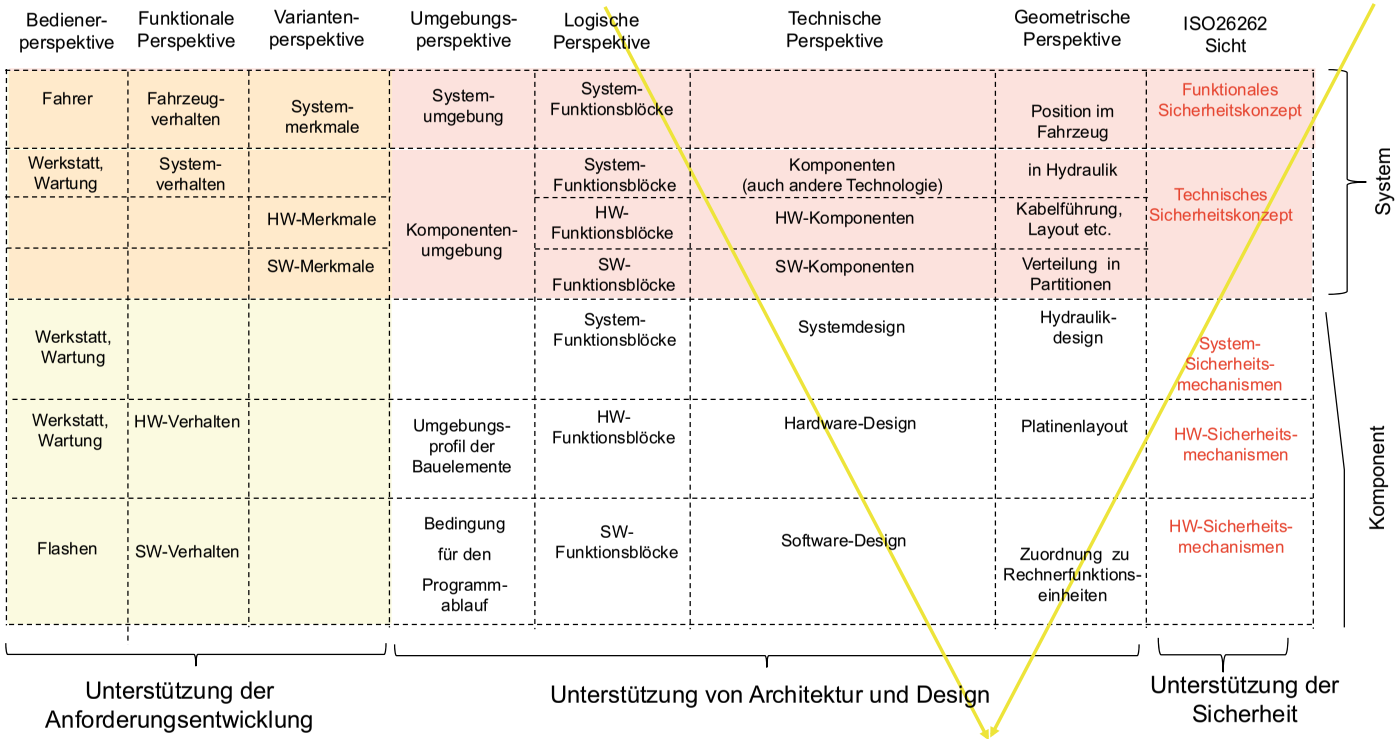

Bild 3.6 Perspektiven einer Architektur (Quelle: Förderprojekt „Safe“)

ten geben, die man bis in einzelne Zeilen des Softwarecodes, den Widerstand oder die Lötstellen der Bauelemente auf der Platine herunterbrechen kann. Sollte man diese Abhängigkeiten nicht kennen, wäre die Frage erlaubt, wofür braucht man die Elemente überhaupt.

Somit wird bereits in der Flugzeugtechnik von der Flugzeugebene, Systemebene und den Komponentenebenen gesprochen. Wäre dies auch in der Automobilbranche eine gewachsene Vorgabe für die Struktur, würde man sich bei der Entwicklung von Fahrzeugfunktionen sicher leichter tun. Offiziell wurden diese Ebenen auch nicht in der ISO 26262 eingeführt, doch die Norm lässt sich wesentlich besser verstehen, wenn man solche Ebenen in Erwägung ziehen würde.

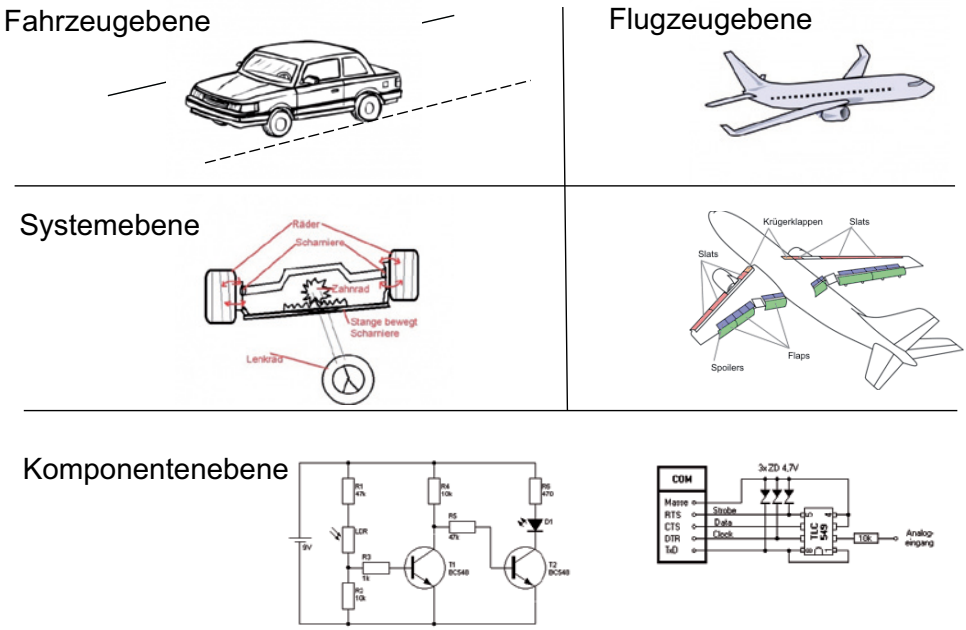

Bild 3.7 Vergleich Analogie Fahrzeugebene / Flugzeugebene sowie die System- und Komponentenebene

Hier sieht man, dass die Umgebung für ein Fahrzeug und ein Flugzeug einen wesentlichen Einfluss darauf hat, wie man ein System entwickelt. Alleine die Freiheitsgrade bei einem Fahrzeug sind für ein Lenksystem wesentlich geringer als für ein Flugzeug. Aber selbst für ein Fahrzeug sind die Freiheitsgrade sehr unterschiedlich. Ein Motorrad kippt um, ein PKW wird zunächst nicht umkippen, es sei denn, man denkt an den Elchtest. An diesem Vergleich sieht man direkt, dass bestimmte Events eine designbedingte Eintrittswahrscheinlichkeit haben. Auf der Systemebene, wo

das Zusammenspiel der Komponenten dargestellt wird, werden auch Mechanik-, Elektronik- und Softwarekomponenten eingesetzt, jedoch werden sie auf Basis anderer Anforderungen als auch anderer Umgebungsparameter zu einem sehr unterschiedlichen Systemdesign führen. Betrachtet man die Komponentenebene, wird sich für Elektronikhardware und Software wieder ein ähnliches Bild ergeben, hier werden die Unterschiede in erster Linie tatsächlich in der Architektur liegen. Es gibt Tendenzen, dass die Funktionen im Flugzeug sich mehr und mehr den Automobilarchitekturen annähern. Bisher wurden Vergleicher oder Voter (zum Beispiel ein 2-von-3-Auswahlsystem) meist auf der Systemebene definiert und durch zum Beispiel drei unabhängige Komponenten (auch Geräteredundanz genannt) realisiert, die dann über einen unabhängigen Mehrheitsentscheider oder durch eine passive Logik (Relais, Dioden, Schalter oder ähnlich) die Sicherheitsfunktionen umsetzen.

In der Automobilindustrie gibt es seit annähernd 20 Jahren das EGAS-Prinzip. Hier werden redundante Software-Ebenen implementiert, die dann je nach Bedarf Sicherheitsfunktionen priorisieren, über Enable-Leitungen Signale in den sicheren Zustand überführen, oder ein intelligenter Watchdog schaltet den gesamten Rechner ab. Im Englischen, insbesondere im Flugzeugbau wird auch von einem Command/Monitoring-System gesprochen. Gemeinsam haben die Konzepte, dass die Zielfunktionalität (Sollfunktion) unabhängig von der Überwachungsfunktion umgesetzt wird. Das Designziel einer solchen Überwachungsfunktion ist es, sie selbst so realisieren zu können, dass bei deren Fehlverhalten oder Ausfall keine Gefährdung durch das Produkt entstehen kann. Solche Prinzipien der Redundanz haben sich weiterentwickelt und werden in der ISO 26262 unter anderem als ASIL-Dekomposition beschrieben. Hier kann man diese Vergleiche oder Votings dann als reine unabhängige oder hinreichend rückwirkungsfreie Software-Funktion realisieren. Um diese Unabhängigkeit und/oder Rückwirkungsfreiheit zu erzielen sind auch weiterhin System- oder Hardwaremaßnahmen notwendig, jedoch versucht man die Komponenten- oder Steuergeräteredundanz zu vermeiden. Diese Tendenz wird auch sehr stark von den Halbleiterherstellern unterstützt, die entsprechende Diagnose, Speicherseparierung oder Redundanzen (diversitäre I/O-Peripherie oder mehrere Rechnerkerne) auf einem Basischip anbieten.

Dieser Vergleich mit der Luftfahrtindustrie zeigt, dass es nicht eindeutig ist, wo die Grenzen liegen und anhand welcher Parameter die horizontalen Schnitte zu legen sind.

Die ISO 26262 geht bei der Systemintegration von drei (horizontale) Integrationsebenen aus. In Teil 4, Kapitel 8 legt man folgende Ziele fest:

Die Integrations- und Testphasen bestehen aus drei Phasen und zwei vorrangigen Zielen. Die erste Phase ist die Integration der Hardware und Software von jedem Element, aus denen das Fahrzeugsystem besteht. Die zweite Phase ist die Integration der Elemente eines vollständigen Systems, welche das Fahrzeugsystem bilden. Die dritte Phase ist die Integration des Fahrzeugsystems mit anderen Systemen des Fahrzeugs und deren Integration in das Fahrzeug selbst.

Daraus ergeben sich folgende drei horizontale Integrationsebenen:

- Integration des Fahrzeugsystems (Items) in das Fahrzeug
  Fahrzeugschnittstelle
- Integration der Komponenten zu einem definierten System
  Komponentenschnittstelle
- Integration von Elektronik-Hardware und der eingebetteten Software
  Hardware-Softwareschnittstelle

Da die Schnittstellen zu diesen Ebenen natürlich Einfluss auf die Architektur und die notwendigen Anforderungen für diese Ebenen haben, müssen diese Ebenen auch bei der Systemanforderungsentwicklung bereits berücksichtigt werden. Das heißt, die Architektur muss so geplant werden, dass es diese horizontalen Schnittstellen bereits gibt.

Bild 3.8 zeigt diese drei Systemebenen eingebettet zwischen der Komponentenebene und der Fahrzeugebene.

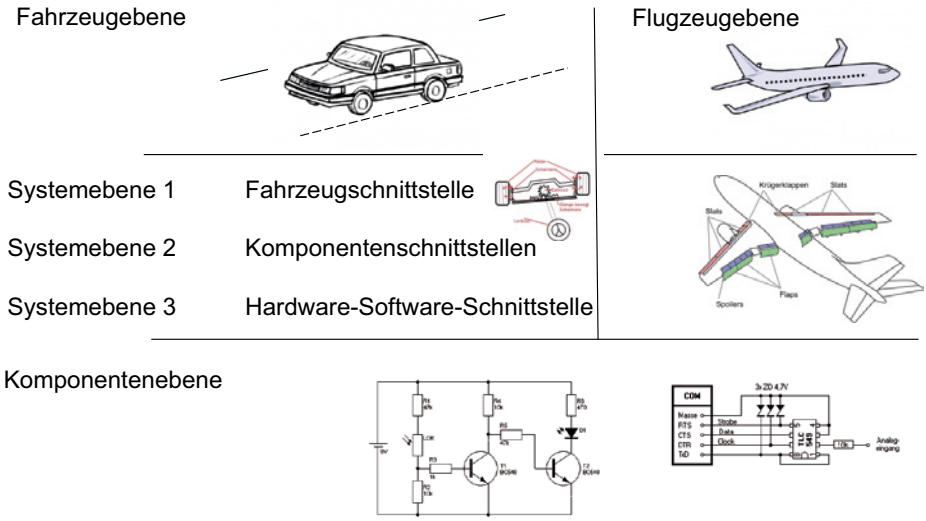

Bild 3.8 Systemebenen zwischen Komponenten- und Fahrzeugebene

Systemebene 1 orientiert sich an der Schnittstelle eines Items (Fahrzeugsystem), hier werden bereits sehr viele Entscheidungen und Definitionen getroffen, die für die spätere Komponentenrealisierung wesentlichen Einfluss haben können. Alle Anforderungen, die in der ISO 26262 Teil 3, Kapitel 4 „Item Definition“ adressiert werden, können für die Komponenten wichtig sein.

Gemäß dem Ford-FMEA-Handbuch gibt es vier Arten von Schnittstellen, die hier mit Beispielen beschrieben werden.

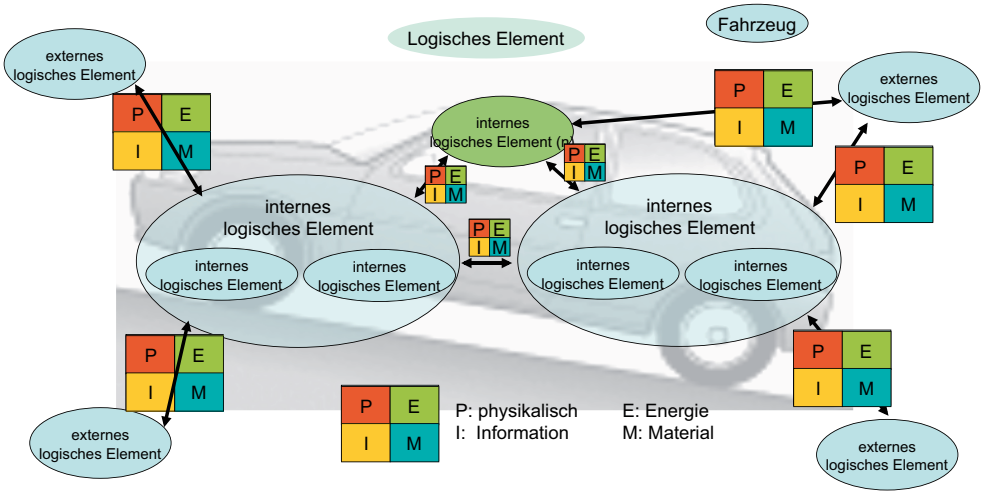

Bild 3.9 Schnittstellenanalyse (Quelle: abgeleitet von Ford-FMEA-Handbuch)

Folgende Faktoren können betrachtet werden:

Physikalische Schnittstellen

- geometrische Daten, die den Raum im Fahrzeug beschrieben, in den die Komponenten zu integrieren sind
- Umgebungsbedingen, wie Vibrationen, Temperaturen, Verschmutzung
- physikalische Größen oder Begrenzungen, wie Kraft, Momente, Drehzahlen, Stellwinkel, Übersetzungsverhältnisse, deren Toleranzen
- elektrische Größen, wie Spannungen, Ströme, EMV

Informationsschnittstellen

- Art der Information
- Datenformate, Dateninhalte, Signalpegel
- Datenschnittstellen, Bus- oder Kommunikationssysteme (CAN, Flexray, Ethernet)
- Netztopologie (Stern, Knoten, Gateways)

Energieschnittstellen

- Art der Energie, wie elektrische, kinetische Energie oder als Druck, Vakuum
- Energietransfer, wie Spannungsklassen, Kurzschlussströme, Sicherungsauslegung
- Energiemenge, wie Ladung von Batterie, Kondensatoren
- Art der Energiebereitstellung, wie über Kabel, Induktion

Materialtransfer (Schnittstellen)

- Kraftstoffförderung, Schmierstoffe
- Materialverträglichkeit, wie harte / weiche Materialien, Getriebe- oder Hydrauliköl
- Masseverschiebungen, Beladungszustände

Alle diese Schnittstellen können auch eine zeitliche Abhängigkeit aufweisen. Es ist wichtig, die Information zur Bremse zu leiten, dass das Fahrzeug bremsen soll, jedoch muss auch sichergestellt sein, dass dem Aktuator die Energie in hinreichender Form zur Verfügung gestellt wird. Dies ist bei einer hydraulischen Bremse weitgehend durch den Bremsdruck gewährleistet. Jedoch bei elektrischer Energie kann ein Bordnetz zum Beispiel bei einer bestimmten Last nicht mehr hinreichende Energie bereitstellen oder die Sicherung löst ab einer bestimmten Schwelle aus.

Auf der Systemebene 2 können die Beschreibungen der Schnittstelle anders sein, auch die zeitlichen Anforderungen werden hier meist detaillierter und damit oftmals kürzer sein.

Ein Beispiel aus der Lenkung zeigt diese hierarchische Kaskadierung. Ein Lenksystem kann einen Fehler ab einer bestimmten Impulslänge und bestimmter Energie circa 20 Millisekunden tolerieren. Steht der fehlerhafte Impuls länger an, wird der Fahrer das Fahrzeug nicht mehr beherrschen können und er fährt womöglich in den Gegenverkehr. Das heißt, die Sicherheitstoleranzzeit beträgt für ein solches System angenommene zwanzig Millisekunden. Heruntergebrochen auf das Steuergerät kann sich diese Zeit zum Beispiel auf unter fünf Millisekunden reduzieren. Sprich, von Steckerpin zu Steckerpin muss das Steuergerät unter fünf Millisekunden eine sicherheitstechnische korrekte Reaktion einleiten können. Um dies dann sogar auf der Systemebene 3 am Hardware-Software-Interface gewährleisten zu können, muss ein Mikrokontroller unter einer Millisekunde eine Softwarefunktion ausführen können, die am Pin des Mikrokontrollers eine adäquate Reaktion ausweisen kann.

Auf der Systemebene 2 könnten die Schnittstellen wie folgt detailliert werden:

Physikalische Schnittstellen

- geometrische Daten im Gehäuse, wie Anbau von Stecker, Platine.
- Umgebungsbedingen, wie Vibrationen, Temperaturen, Verschmutzung (diese Daten können variieren, da Sensoren, Steuergräte oder der Aktuator an verschiedenen Stellen verbaut werden oder die Gehäuse vor Schmutz / Feuchtigkeit schützen, Vibrationen reduzieren, Wärme ableiten)
- physikalische Größen oder Begrenzungen, wie Kraft, Momente, Drehzahlen, Stellwinkel, Übersetzungsverhältnisse sowie deren Toleranzen (diese Größen können wieder unterschiedlich auf die Elemente des Systems heruntergebrochen oder aufgeteilt sein)
- elektrische Größen, wie Spannungen, Ströme, EMV (siehe oben physikalische Größen)

Informationsschnittstellen

- Art der Information (hier wird die Information meist näher spezifiziert)
- Datenformate, Dateninhalte, Signalpegel (Detaillierung dieser Informationen)
- Datenschnittstellen, Bus- oder Kommunikationssysteme (CAN, Flexray, Ethernet), hier wird nun die physikalische Spezifikation der Kommunikationsschnittstellen notwendig sein, damit man mit internen und externen Kommunikationspartnern auch kommunizieren kann
- Netztopologie (Stern, Knoten, Gateways), hier werden diese Elemente detaillierter spezifiziert

Energieschnittstellen

- Art der Energie, wie elektrische, kinetische Energie oder als Druck, Vakuum
- Energietransfer, wie Spannungsklassen, Kurzschlussströme, Sicherungsauslegung
- Energiemenge, wie Ladung von Batterie, Kondensatoren
- Art der Energiebereitstellung, wie über Kabel, Induktion

Diese Schnittstellen werden nun auf die einzelnen externen wie internen Komponenten heruntergebrochen und je nach Bedarf entsprechend detailliert.

Materialtransfer (Schnittstellen)

- Kraftstoffförderung, Schmierstoffe
- Materialverträglichkeit, wie harte / weiche Materialien, Getriebe- oder Hydrauliköl
- Masseverschiebungen, Beladungszustände
- Auch diese Schnittstellen werden nun auf die einzelnen externen wie internen Komponenten heruntergebrochen und je nach Bedarf entsprechend detailliert.

In der Systemebene 3 werden sich oft neue Informationen ergeben, die sich aus Spezifikation des Mikrokontrollers ergeben. Aber auch hier werden wieder alle vier Schnittstellenkategorien mehr oder minder relevant sein. Materialtransfer wird für den Mikrokontroller weniger relevant sein, aber wenn man über Lebensdauersicherheit spricht, werden die Materialschnittstellen schon relevant. Hier gibt es Kontaktierungsprobleme wegen falscher Materialien bis hin zu Drift oder sporadische Effekte durch Korrosion.

Komponentenebenen

Reine Mechanikkomponenten in verschiedenen Abstraktionsebenen zu definieren, kann im Einzelfall sinnvoll sein, aber allgemein werden komplexere Mechanikkomponenten bereits auf der Systemebene beschrieben. Auch die Schnittstellen können jeder beliebigen Systemschnittstelle zugeordnet werden. Eine rein hydraulische Lenkung wird demnach mehr in der Systemebene 1 integriert, und Teile wie Platine, Stecker etc. womöglich auf der Komponentenebene.

Bei Software kommt es oft vor, dass es mehrere Softwarekomponenten gibt, die dann zur gesamten Embedded-Software in einem Mikrokontroller integriert werden. Formal kann man auch mehrere funktionale Gruppen, die in einem Mikrokontroller integriert sind, als Systemelemente integrieren. Da aber die Hardware-Software-Schnittstelle sehr viele Anforderungen an die zu realisierende Software stellt,

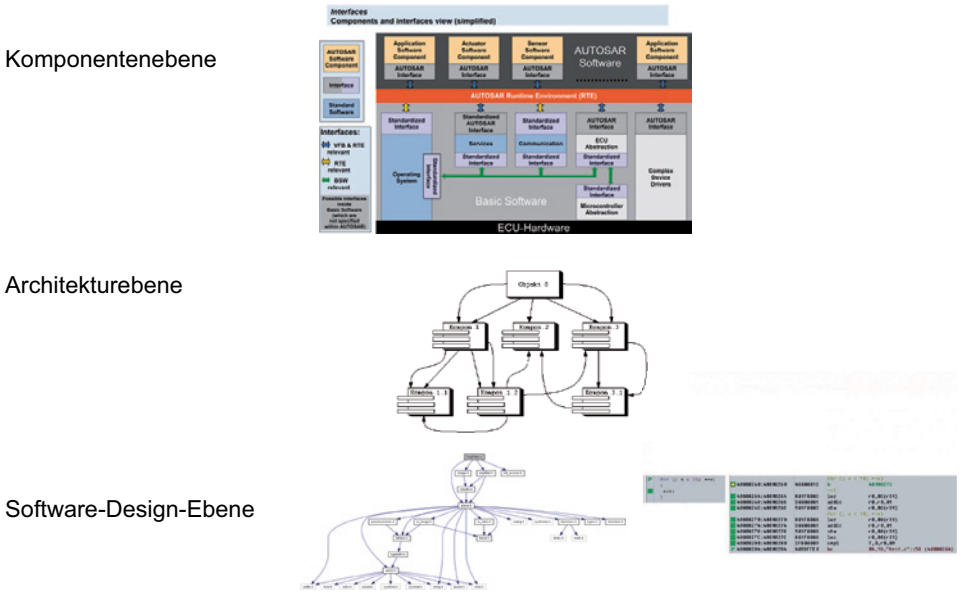

Bild 3.10 Software Abstraktionsebenen

wird die Schnittstelle sehr komplex werden. Zeitliche Analysen bei einer solchen Integration werden nur über alternative Sichten analysierbar sein, da bei jedem Softwareelement, die Laufzeit des Rechners betrachtet werden muss. Hier gibt es die sogenannte Architekturebene; sie bildet die Ebene zwischen dem Software-Design und der Software-Komponente. Wie im SPICE wird auch der Begriff Software Unit als die kleinste Einheit der Software betrachtet. Sprich: Anweisungen werden nicht mehr als eine Einheit angesehen. Bei der in der Automobilindustrie oft verwendeten C-Programmiersprache würde dann das C-File diese Software-Unit repräsentieren. In der Softwareentwicklung wird meist noch zwischen zwei Ebenen unterschieden, die der Basissoftware und der Applikationssoftware. Schnittstellen, die zur Laufzeit eine definierte Datenstruktur bereitstellen, eine sogenannte Laufzeitumgebung (Real-Time-Environment, RTE), wie man sie aus AutoSar kennt, bieten eine Separierung zwischen Basis- und Applikationssoftware.

Bei der Elektronik beginnt man meist doch wieder bei der Mechanik. Hier gibt es zuerst Gehäuse, Stecker, Halter, Platinen, Lüfter, Kühlereinrichtungen. Diese geben natürlich schon einige Parameter und Designbeschränkungen für die Realisierung vor. Da ein Gehäuse auch sehr früh in einem Projekt bestellt werden muss, wird die gesamte Elektronikauslegung vom Gehäusedesign abhängig sein. Formal sprechen wir hier über die Abstraktionsebene der Elektronikarchitektur. Daher sollte man diese Designabhängigkeiten kennen, aber nicht als eine eigene Abstraktionsebene definieren, sehr wohl können diese Mechanikkomponenten sinnvolle Trennungen für verschiedene Elektronikkomponenten sein: mehrere Elektronikkomponenten auf verschiedenen Leiterplatten, Trennung von Steuer- und Leistungselektronik, verschiedenen Spannungsebenen oder auch technische Trennung von sicherheitsrelevanter Elektronik und nichtsicherheitsrelevanter Elektronik. Bei Software trennt man auf jeden Fall sinnvoll, wenn verschiedene Software-Komponenten in verschiedene Mikrokontroller integriert werden. In der Elektronik wird man oft auch andere Kriterien finden müssen, wie man Komponenten, Funktionsgruppen und Bauelemente separiert. Daher würde man bei Elektronik die drei Abstraktionsebenen als Komponentenebene, Funktionsgruppenebene und Bauelementeebene bezeichnen können. Halbleiter, wie Mikrokontroller, ASICs, FPGAs oder andere Hybride, auch wenn sie als ein Bauelement gelten, werden oft als Funktionsgruppe integriert.

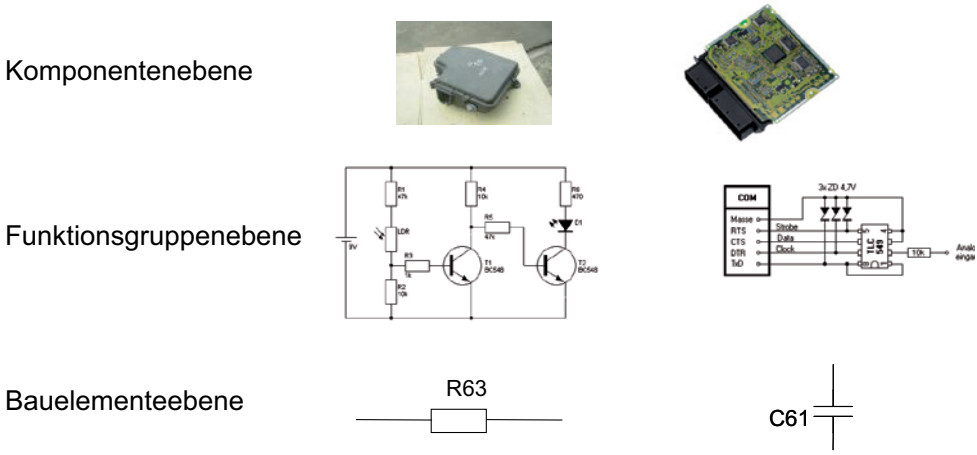

Bild 3.11 Abstraktionsebenen in der Elektronik

Besonderheit Software: Basic-Software - Applikationssoftware

Auch in der Basic-Software oder in AutoSar spricht man von Abstraktionsebene, jedoch sind hier nicht unbedingt horizontale Abstraktionsebenen gemeint, sondern *funktionale (perspektivische)* Abstraktionsebenen. Trotzdem spielt die Schnittstelle zwischen Applikationssoftware und Mikrokontroller eine besondere Rolle in der Definition der Abstraktionsebenen. Gemäß ISO 26262 wurde das Hardware-Software-Interface zuerst im Teil 5 (Produktentwicklung auf Elektronik Hardwareebene) und im Teil 6 (Produktentwicklung auf Softwareebene) und erst nach dem CD-Stand der Norm in Teil 4 (Produktentwicklung auf Systemebene) eingebunden, so dass das System die Schnittstelle zwischen Software und Hardware beschreibt. Die Besonderheit hier liegt in erster Linie darin, dass der Mikrokontroller als Hardwareelement, ähnlich wie das Gehäuse für die Elektronik, wesentliche Designmerkmale für die Software vorgibt. Damit diese beiden Komponenten richtig miteinander arbeiten können, müssen die Eigenschaften wie auch die potentiellen Fehler genauso betrachtet werden wie die Funktionen und möglichen Fehlfunktionen. Dies gilt natürlich für alle Komponentenschnittstellen, aber an der Hardware-Softwareschnittstelle handelt es sich um sehr viele relevante Schnittstellenparameter. Das heißt, es geht nicht nur um die korrekte Funktion der sogenannten Low-Level-Treiber, die die Informationen der Mikrokontroller der Software zur Verfügung stellen, das Betriebssystem, Peripherie, interne Kommunikation, Logikeinheit, Speicher oder Funktionsbibliotheken, die der Rechner zur Verfügung stellt, sondern auch um die systematische Absicherung vor möglichen Fehlern oder Fehlverhalten an dieser Schnittstelle.

3.4 Anforderungs- und Architekturentwicklung

Die Architektur sollte auch die Struktur der Anforderungen wiedergeben. Mit den beschriebenen horizontalen Abstraktionsebenen werden die oberen und unteren Schnittstellen für die Details vorgegeben, die die Architektur darstellen soll. Durch die Festlegung von logischen und technischen Elementen werden weitere Schnittstellen innerhalb einer horizontalen Abstraktionsebene dargestellt. In einem System werden logische und technische Elemente definiert, die die Aufgabe haben, die geforderte Funktion zu tragen beziehungsweise zu implementieren oder zu realisieren. Die logischen oder technischen Elemente müssen eindeutig spezifiziert werden, damit man für sicherheitsrelevante Systeme ein korrektes Verhalten erwarten kann.

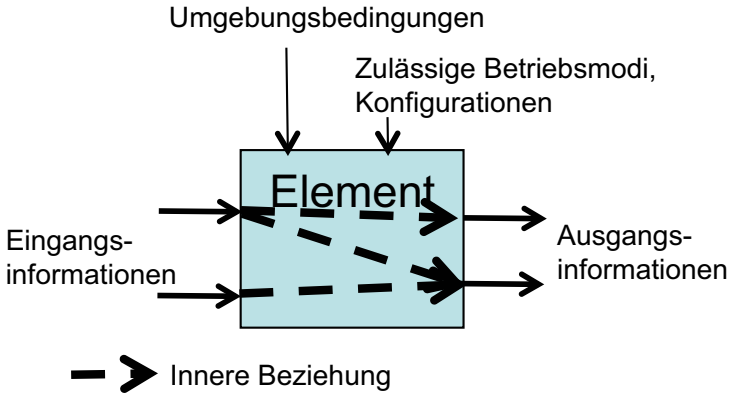

Bild 3.12 Spezifikation von Elementen für technischer Systeme

Die folgenden Eigenschaften (Merkmale oder Fähigkeiten) sollten als Anforderung betrachtet werden:

- Die Umgebung, in die das Element eingebettet werden soll, muss so spezifiziert werden, dass alle Einflussfaktoren, die das Verhalten des Elements beeinflussen können, definiert sind. Welche Faktoren zu betrachten sind, ist das Ergebnis einer Schnittstellenanalyse (im Sinne einer Boundary-Analyse (Begrenzungsanalyse)).
- Die zulässigen Einsatzarten, Betriebsmodi (zum Beispiel Initialisierung, Monitoring, On-Demand, Stand-by, Regelbetrieb) oder zulässigen Konfigurationen (zum Beispiel nur für Analogwertverarbeitung, Aufruf mit bestimmten Parametern (zum Beispiel bei Softwareelementen), getakteter oder getriggerter Betrieb) müssen spe-

zifiziert werden und ebenso die Art und Weise, wie die Information dem Element zugeordnet werden.

- Die Eingangsinformationen sollten so spezifiziert werden, dass definiert ist, wo diese generiert werden, in welchem Format diese übermittelt werden und in welchen Bereichen die Informationen gültig sind.
- Die Ausgangsinformationen sollten so spezifiziert werden, dass definiert ist, wohin diese adressiert werden, in welchem Format diese bereitgestellt werden und in welchen Bereichen die Informationen gültig sind.
- Die inneren Beziehungen sollen alle Ein- und Ausgabebedingungen in der spezifizierten Umgebungsbedingung unter den zulässigen Betriebsmodi oder Konfigurationen definieren. Sollte ein speicherndes Verhalten in den Elementen die inneren Beziehungen verändern können, so müssen diese ebenfalls durch die Spezifikation definiert werden. Speicherndes Verhalten innerhalb des Elementes führen zu geänderten Ein-/ Ausgabebeziehungen. Dies muss definiert werden.

Neben den funktionalen Eigenschaften werden bei den technischen Elementen folgende Eigenschaften einen Einfluss auf die elektrischen, elektronischen und mechanischen Hardwareelemente haben:

- Geometrie, Form, Volumen, Masse, Struktur, Oberfläche, Kennzeichnung, Farbe etc.
- Materialeigenschaften (Materialverträglichkeit, chemische Reaktionsfähigkeit)
- Verhalten und Reaktion bei physikalischer Beeinflussung, wie Temperatur, Strom, Spannung, Stressverhalten (Vibration, EMV, bestimmtes Verhalten gegenüber physikalischer Beanspruchung)
- Alterungseffekte (statistisches Alterungsverhalten (Weibull-, Binomial-, Chi-Verteilung))
- Wartungsanforderungen, Logistik
- zeitliche Aspekte

Aber auch technische Softwareelemente haben technische Eigenschaften, wie

- Größe des kompilierten Codes
- Verzweigungen, Speicherbedarf, Anzahl (Instruktionsaufrufe, Variablen, Adressen, Sprungbefehle, Interrupts) etc.
- realisierter Programmablauf, Task-Zugehörigkeit der Elemente

Kommerzielle, ideelle oder emotionale Aspekte sind hier nicht näher betrachtet, weil diese nicht in Zusammenhang mit Sicherheit betrachtet werden sollten. Ob all diese technischen Größen so oder anders definiert und spezifiziert werden müssen, ist allgemein wieder das Ergebnis einer Analyse. Man kann davon ausgehen, dass ein technisches Element bestimmte Eigenschaften hat, weil es sonst nicht die gedachte

Eigenschaft oder Funktion erfüllen kann. Somit ist naheliegend, dass wenn diese Eigenschaften nicht mehr gegeben oder konsistent sind, eine Funktionsbeeinträchtigung eintreten kann.

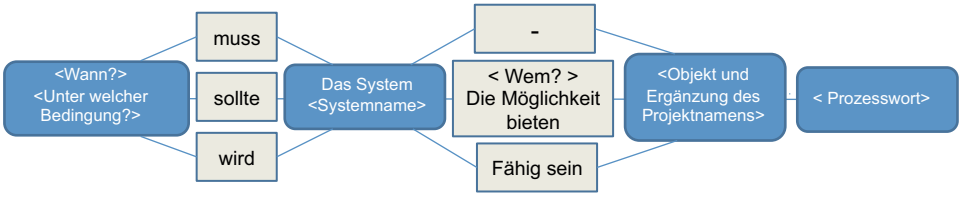

Bild 3.13 Anforderungsschablone (Quelle: in Anlehnung an Chris Rupp, Requirements-Engineering und -Management)

Für alle diese Anforderungsaspekte sollte man Schablonen entwerfen, so dass die Anforderungen in einem klaren Format vorliegen, somit wird man falsche Interpretationen vermeiden und die Konsistenz mit der Architektur gewährleisten können. Spezifikation in natürlicher Sprache heißt nicht, dass alle Listen von Eigenschaften in Prosa formuliert sein müssen. Besonders beim technischen Verhalten eignen sich semi-formale Methoden wesentlich besser und sind eindeutiger und damit unmissverständlicher als wohl formulierte Sätze. Bei einer gut strukturierten Architektur werden durch solche Templates oder Schablonen viele Anforderungen auch automatisiert aus der Architektur ableitbar sein. Oder die wesentlichen Inhalte lassen sich durch die Definition von Key-Worten entsprechend vorformulieren, dass man nur noch Parameter oder bestimmte Eigenschaften ergänzen muss. Sämtliche Signalfluss- oder Datenflussaspekte müssen konsistent zur Architektur sein, das heißt, wenn diese Anforderungen systematisch aus der Architektur abgeleitet werden, kann man eine gute Konsistenz erwarten.

3.5 Anforderungs- und Designspezifikation

Sämtliche Anforderungen zum Anforderungsmanagement beziehen sich auf die Anforderungsspezifikation in der ISO 26262. Anforderungen zu einer Designspezifikation finden wir sehr selten in Normen und Standards. Hier gilt wohl nur die Anforderung, dass der Inhalt verstanden werden muss.

Die Kunst liegt darin, eine gesunde Mischung zwischen Anforderung und Design zu finden und dies hinreichend und korrekt zu spezifizieren.

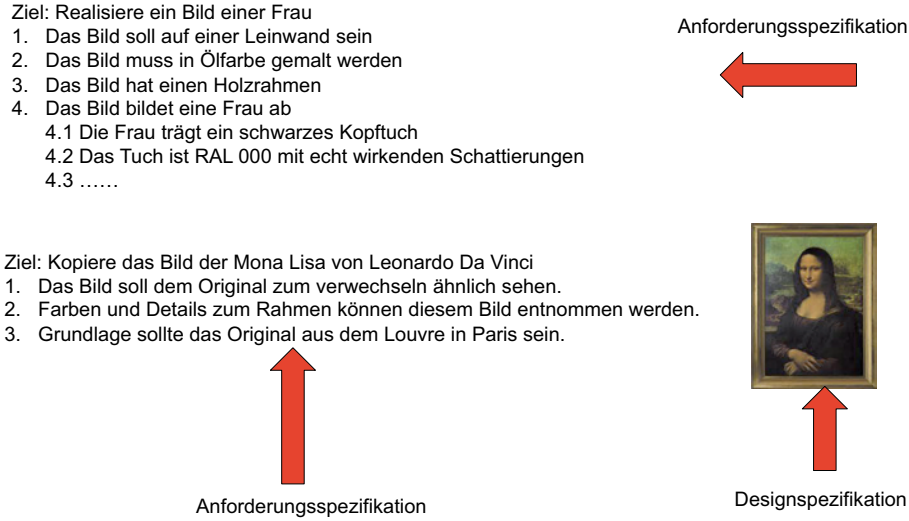

Bild 3.14 Anforderungs- und Designspezifikation

Dieses Beispiel mit dem Bild der Mona Lisa zeigt, eine reine Anforderungsspezifikation sehr umfangreich werden kann. Der Adressat der Anforderungsspezifikation wird es aber schwer haben aus dem Text das geforderte Bild umsetzen zu können. Eine gute Mischung aus Anforderungen und klar geforderten Designeigenschaften, die auch entsprechend illustriert werden, kann zielführender sein.

Für eine Mechanikkonstruktion würde niemand vorschlagen eine M6-Schraube mit einer Anforderungsspezifikation zu spezifizieren. Weiter würde auch niemand freiwillig eine Anforderungsspezifikation für einen Widerstand mit 100 Ohm und einer Toleranz von 1 % schreiben wollen. Nun stellt sich die Frage, wenn dies für Elektronik und Mechanik so eindeutig ist, wie definiert man die Grenze beim System oder bei der Software? Das Bild mit der Mona Lisa zeigt eindeutig, dass reine Anforderungsspezifikationen nicht unbedingt zielführend sind. Hier stellt sich allgemein wieder die Frage: Wie gliedert man Spezifikationen und an wen wird die Spezifikation gerichtet? Die Spezifikation eines Fahrzeugsystems ist doch im Allgemeinen nicht an einen Autofahrer gerichtet, sondern sollte doch an einen Fachmann gerichtet sein. Das heißt, für einen Systementwickler sollten Timing-Diagramme, Tabellen, Sequenzdiagramme und so weiter aussagekräftige Informationen sein. Diese in Form von Anforderungen nochmals in natürlicher Sprache zu spezifizieren, ist nicht unbedingt ein effizienter Mehrwert. Das eindeutige Beschreiben vom technischen Verhalten ist auch über Modelle oft einfacher erklärbar. Im Grunde genommen ist das Bild der Mona Lisa nichts anderes als ein Modell, welches die Anforderungen spezifisch ergänzt. Die Designkapitel im System Teil 4 Kapitel 7 fordert eine System-

designspezifikation und in der Software Kapitel 8 eine Softwaredesignspezifikation, aber keine Anforderungsspezifikation.

Teil 8 der ISO 26262 Kapitel 6 beschreibt die Anforderungen und wie sie gemäß ISO 26262 gemanagt werden sollen. Der Absatz „General“ definiert, dass während des Sicherheitslebenszyklusses Sicherheitsanforderungen spezifiziert werden müssen und hierarchisch detailliert werden sollen. Die Struktur und Anhängigkeiten werden durch das folgende Bild dargestellt.

Das folgende Bild (3.15) ist ein Ausschnitt aus dem Sicherheitslebenszyklus und es zeigt wie die Aktivitäten, Anforderungen und Arbeitsergebnisse von der Konzeptionierung in die Entwicklung einfließen.

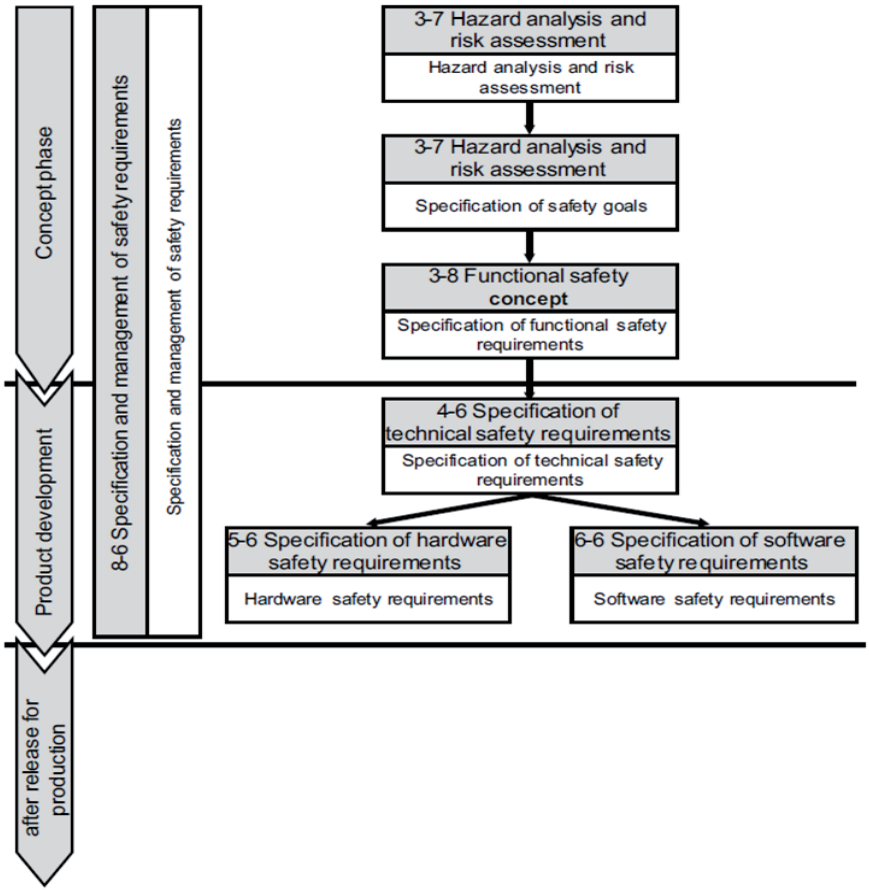

Bild 3.15 Struktur von Anforderungen (Quelle: ISO 26262, Teil 8, Kapitel 2)

Die Sicherheitsanforderungen werden den Elementen (der Architektur) zugeordnet oder verteilt.

Das nachfolgende Bild 3.16 verdeutlicht die Anforderungen an Anforderungen und die Art und Weise wie ein Ingenieur (Requirement Engineering) damit umgehen sollte beziehungsweise sie gemanagt werden sollten.

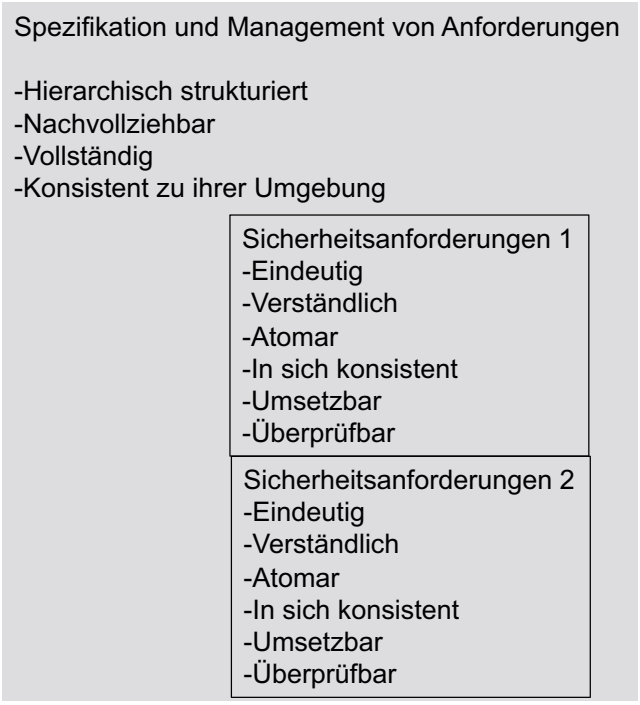

Bild 3.16 Beziehung zwischen Management von Sicherheitsanforderungen und den Anforderungen (Quelle: in Anlehnung an ISO 26262, Teil 8, Bild 3)

Es gibt keine Anforderungen in den gängigen Sicherheits- oder Entwicklungsstandards, dass alle Eigenschaften von Designelementen in Form von Anforderungen spezifiziert werden müssen. In allen Architekturkapiteln spielen Anforderungen aber wieder eine wichtige und zentrale Rolle, Anforderungen werden an die Architektur allokiert. Eine Software-Unit und ein Elektronikbauelement muss demnach nicht vollständig mit Anforderungen spezifiziert werden, aber worauf bezieht sich dann die Vollständigkeit? Die Kunst besteht jetzt darin, die Ebene zu finden, die ausreicht, um alle sicherheitsrelevanten Kenngrößen und das sicherheitstechnische

Verhalten eindeutig und hinreichend zu definieren. Diese Ebenen müssen geplant werden und in einer Anforderungs- und Architekturstrategie eindeutig beschrieben werden. Technische Produkte oder Teile davon sowie ihre Eigenschaften, Einschränkungen, Anwendungsbereich, ihr Verhalten und so weiter sollten immer über eine sinnvolle Mischung aus Anforderungen, Architektur und Designvorgaben spezifiziert werden. Produkte werden nie durch ihre Fehler beschrieben oder durch Risiken gekennzeichnet, trotzdem kann es notwendig sein, diese auch für den Anwender zu dokumentieren (Beipackzettel, Handbücher etc.).

Funktionale Architektur und Verifikation

Eine Funktion ist allgemein ein mathematischer Ausdruck oder ein Zusammenhang.

$$f(x) := ay+bx$$

Dies ist eine typische mathematische Funktion. Bei systemischen Funktionen gibt es folgende Darstellungen für folgende Funktion:

Funktion1 := Funktion1.1 v Funktion1.2 v Funktion1.3

Bild 3.17 Funktionsdekomposition als Baum oder Liniendiagramm

Alle 3 Darstellungsarten repräsentieren den selben Zusammenhang, es handelt sich nur um eine andere Darstellung der Information, dass Funktion 1 sich aus den drei Teilfunktionen zusammensetzt. Funktion 1 repräsentiert die sequentielle Kette der drei Teilfunktionen. Diese rein funktionalen Perspektiven ermöglichen keine Identifikation der Schnittstellen, der System- oder Elementgrenzen. Es ist nur eine eingeschränkte Beschreibung des Verhaltens eines technischen Systems möglich.

Auch die mathematische Übertragungsfunktion beschreibt ein erwartetes Ausgangsergebnis auf Basis von definierten Eingängen.

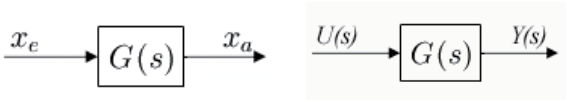

Bild 3.18 mathematische Übertragungsfunktion

Die mathematische Übertragungsfunktion betrachtet bereits Ein- und Ausgänge, damit sind hier bereits Schnittstellen verfügbar. Auch bei einem Matlab-Simulink-Modell sind die Eingangs-Ausgangsbeziehungen die Grundlagen für die Schnittstellen in der Architektur.

Werden Anforderungen von einem Element auf eine innere Struktur abgeleitet, so entstehen durch die innere Struktur neue Vorgaben für Schnittstellen.

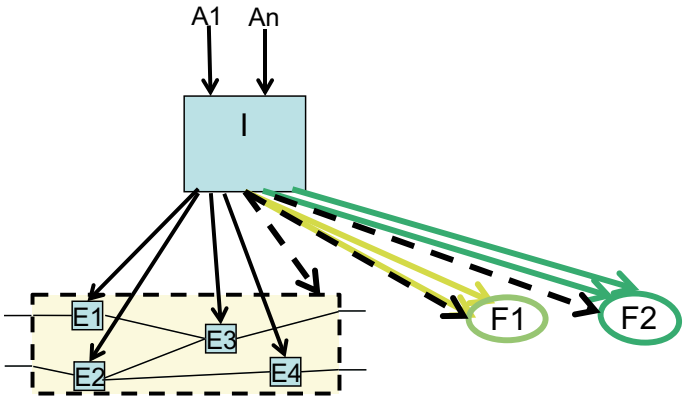

Bild 3.19 Allokation von Funktionen auf logische Elemente

Allokation von Funktionen, Teilfunktionen und ihren Anforderungen (sog. funktionale Anforderungen) zu einem logischen Element ist die Hauptaktivität bei der Entwicklung des funktionalen Sicherheitskonzepts neben der Verifikation dieser Anforderungen. Ohne eine solche Allokation ist eine Verifikation nicht möglich. Die logischen Elemente E1 bis E4 sollen Funktion 1 und Funktion 2 umsetzen. Die Allokation könnte zu folgendem Resultat führen:

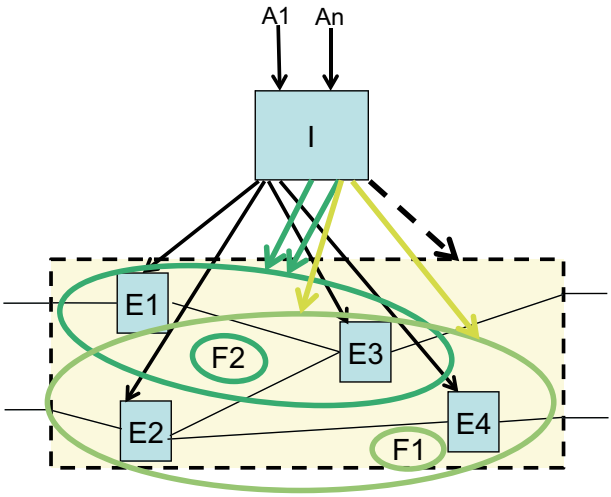

Bild 3.20 Allokation von Anforderungen auf logische Elemente

Logische Elemente haben eine Begrenzung und identifizierbare Schnittstellen, aber auch die Funktionen erhalten Schnittstellen und Begrenzungen durch die Zuordnung zu den logischen Elementen.

In dieser Struktur und mit den hier vorliegenden Informationen und Zusammenhängen können die Anforderungen verifiziert werden. Folgende Verifikationen der Anforderungen wären möglich:

Wurden alle Anforderungen abgeleitet?

Wurden die Anforderungen so klassifiziert, dass es eindeutig ist, ob es sich um Anforderungen an ein Eingangssignal, Ausgangssignal, eine Beziehung innerhalb eines Elementes, eine Beziehung zwischen zwei Elementen, an eine Funktion zwischen zwei Elementen, an die Umgebung der Elemente oder um Designannahmen oder Limitierungen handelt?

Leiten sich wie im Bild 3.20 die Anforderungen nur aus einem höheren Element ab oder gibt es weitere höhere Elemente zum Beispiel außerhalb der Systemgrenze, die die Anforderungen beeinflussen können?

Ist die interne Struktur der abgeleiteten Elemente hinreichend beschrieben?

Diese Frage bildet die Grundlage zur Verifikation des funktionalen Sicherheitskonzepts. In jeder anderen Ebene, in der Anforderungen verifiziert werden, kann eine ähnliche Vorgehensweise für die Anforderungsverifikation angewendet werden. Das obige Bild zeigt, dass wenn Funktionen und die logischen, technischen oder die Elemente, die diese Funktion realisieren sollen, keine gemeinsame Schnittstellen haben, dann explodiert die Zahl der Schnittstellen exponentiell. Würde man auf

Basis dieser heterogenen positiven Beschreibung noch eine situationsbedingte Fehleranalyse umsetzen müssen und wohl möglich noch über mehrere horizontale Abstraktionsebenen diese Zusammenhänge beschreiben müssen, so wäre Vollständigkeit, Nachvollziehbarkeit, Konsistenz und Korrektheit nicht mehr argumentierbar. Eine solche Vielzahl von Schnittstellen ist nicht mehr analysierbar und damit nicht mehr beherrschbar.