Hoch-Integre Technische Systeme

Dieter Scheithauer

H·I·T·S Engineering, Dr.-Ing. Dieter Scheithauer
Breitensteinstraße 26, 83727 Schliersee, dieter.scheithauer@hitseng.eu

Zusammenfassung: Technische Systeme erfüllen immer einen Zweck, der über das technische System selbst hinausführt. Insofern sollen technische Systeme einen regulierenden Einfluss auf ihre Systemumgebung ausüben. Die Entwicklung technischer Systeme basiert auf Prognosen über die zukünftigen Auswirkungen des jeweiligen technischen Systems auf seine spezifische Systemumgebung wie auch auf die Umwelt im Allgemeinen. Mit zunehmender Systemkomplexität begrenzt die menschliche Vorstellungskraft valide Prognosen. Systematische Prognoseverfahren, wie zum Beispiel numerische Sicherheitsanalysen, können deshalb die Risiken aus der Produzentenverantwortung allein nicht absichern, da sie sich immer auf Modelle abstützen, also anstatt auf die Wirklichkeit selbst auf Approximationen dieser Wirklichkeit. Zur Überbrückung der Differenz wird gern von Systemintegrität gesprochen, ohne den Begriff überzeugend zu präzisieren. Dieser Aufsatz beginnt mit einer Begriffsdefinition hoch-integrer technischer Systeme und leitet Handlungsempfehlungen für die Entwicklung hoch-integrer technischer Systeme ab.

1 Einleitung

Etymologisch leitet sich der Begriff Technik vom griechischen τέχνη ab. Im heutigen Sprachgebrauch wird dem altgriechischen Begriff τέχνη eine Begriffsbedeutung zwischen Handwerk, Kunstfertigkeit und Kunst zugeordnet. Das Leben als Techniker und Ingenieur hat demnach auch eine wichtige kreative Seite. Der Begriff τέχνη ist grundsätzlich positiv besetzt: Τέχνη führt zur Verbesserung der menschlichen Lebensumstände. Zweckrationalität unterscheidet Technik von der reinen Kunst [JM96]. Die von Altphilologen geprägte Übersetzung fasst den Begriff der Technik umfassend, wie es auch der umgangssprachlichen Verwendung des Begriffes Technik entspricht, wenn zum Beispiel von der Technik des Klavierspielens gesprochen wird. In den heutigen Ingenieurwissenschaften existiert die Gewohnheit, Technik vor Allem mit der Nutzbarmachung von Naturgesetzen zur Gestaltung von Maschinen und deren Automatisierung zu assoziieren. Natürlich fehlte im Altertum der moderne mathematische Apparat zur Beschreibung der Naturgesetze, den wir heute in der Technik selbstverständlich einsetzen. Die Mathematik der alten Griechen war Geometrie [HW08]. Aus Naturbeobachtungen gewonnene Erkenntnisse waren allerdings auch damals bedeutsame Elemente von Technik. Nicht nur das Leben des Archimedes legt ein beredtes Zeugnis davon ab. Andererseits ist der Begriff τέχνη auch heute so weit gefasst, dass er bestens zum Systems Engineering passt und die Wirkung der Technik auf die menschliche Gesellschaft ins Zentrum stellt.

Ausgehend von diesem allgemeinen Technikbegriff wird im Folgenden eine Definition für technische Systeme entwickelt. Dabei wird zwischen einem Produkt als technischem System und seiner Systemumgebung klar differenziert. Auf jeder der beiden Betrachtungsebenen erfordert die Entwicklung spezifische Aktivitäten. Viele Standards und Lehrbücher zum System Engineering verzichten auf eine Abgrenzung und stellen alle Aktivitäten aus Sicht des Produktes selbst dar. Der nächste Schritt thematisiert zunehmende Komplexität sowohl im Systemdenken als auch bezogen auf die systemtechnischen Problemstellungen und deren technischen Lösungen. Aus der zunehmenden Komplexität resultiert ein Bedarf für einen Integritätsbegriff, der sich primär auf die Systemumgebung bezieht. Systemintegrität ergänzt somit am Produkt ausgerichtete Thematiken, die insbesondere durch die Betrachtung der funktionalen Sicherheit technischer Systeme dominiert werden. Auf der anderen Seite wirkt Systemintegrität aber auch als treibende Kraft von gesellschaftsbezogenen kontinuierlichen Verbesserungsprozessen.

Den Ausführungen im letzten Absatz gemäß ist Systemintegrität zwar auf der Ebene der Systemumgebung zu verorten, doch heißt dies nicht, dass Systemintegrität auf der Produktebene eine vernachlässigbare Rolle zugewiesen werden kann. Nur durch technische Systeme lassen sich übergeordnete System und deren Systemintegrität gezielt beeinflussen. Zu bedenken bleibt, dass nicht alle Systemelemente im übergeordneten System technische Systeme im oben genannten Sinne sind und sich somit nicht allein nach den Vorstellungen der Entwickler gestalten lassen. Die nicht-technischen Systemelemente müssen so hingenommen werden, wie sie sind. Es stellt sich also die Frage, was in der Produktentwicklung zur Erzielung einer hohen Systemintegrität berücksichtigt werden sollte. Entsprechende Maßnahmen betreffen sowohl die Wahl der Systemarchitektur für das übergeordnete System als auch den Systems-Engineering-Prozess.

2 Technische Systeme

Am Anfang jeder erfolgreichen Produktentwicklung steht immer die Beobachtung eines Sachverhaltes, siehe A in Abbildung 1. Dabei ist weniger die Tatsache an sich ausschlaggebend. Bedeutsam ist, dass diese Tatsache die Aufmerksamkeit einer Person auf sich zieht. Dieses Finden eines Problems ist ein initialer kreativer Akt, der konstitutiv für die effiziente Funktion unserer Gehirne ist. Daniel Kahnemann bezeichnet die Fähigkeit unserer Gehirne, auf komplexe, dynamische Szenarien in der Regel angemessen und zeitgerecht zu reagieren als schnelles Denken [DK11]. Wir erfassen eine Situation demnach ganzheitlich und nicht als Summe der Einzelreize. Aufmerksamkeitsgesteuert widmen wir uns dem Auffälligem, Besonderem oder Unerwartetem. Für einen überzeugten Systemingenieur ist dies wenig verwunderlich, passt es doch zum Postulat der Emergenz in Systemen als eine der Grundüberzeugungen im Systems Engineering: Das Ganze besitzt Eigenschaften und Funktionen, die aus den Teilen allein nicht ablesbar sind. Es ist gut zu wissen, dass Emergenz keine artifizielle Konstruktion ist, sondern bereits in unserer Wahrnehmung der Welt angelegt ist.

Die Bedeutung der Aufmerksamkeitssteuerung für die effiziente Arbeitsweise unserer Gehirne ist offensichtlich, doch sind die neurokognitiven Zusammenhänge bisher nicht

aufgeklärt. Allen Versuchen, dass Gehirn analog zu Digitalrechnern, die nach den von John von Neumann und Alan Turing aufgestellten Prinzipien konzipiert sind, zu erklären oder zu modellieren, sollte deshalb mit höchster Vorsicht und Zurückhaltung begegnet werden.

Dem schnellen Denken steht nach Kahnemann das langsame – meist bewusste und algorithmisch-logische – Denken gegenüber. Langsames Denken schafft Wissen durch das Erlernen von Distinktionen und Zusammenhängen. Es schafft wesentliche Grundlagen im semantischen Gedächtnis [BG15], um auf Umweltsituationen gezielt, flexibel und angemessen reagieren zu können. Kahnemann spricht in diesem Zusammenhang von einer Zehn-Jahres-Regel [DK11]: Zutrauen in die Intuition einer Person ist gerechtfertigt, wenn diese Person sich mehr als zehn Jahre intensiv mit dem betreffenden Sachgebiet auseinandergesetzt hat.

Man sollte aber die fortbestehende Dominanz radikal-positivistischer und reduktionistischer Überzeugungen in den Wissenschaften nicht verkennen. In den Naturwissenschaften und auch unter vielen Ingenieuren lebt der Glaube an eine Einheitswissenschaft mit einer Einheitssprache fort [RC32]. In der Psychologie sind die Nachwirkungen des Behaviorismus [BS76, WB05] noch wirksam, obwohl im Rahmen der kognitiven Wissenschaften immer mehr evident wird, dass von den drei Hauptströmungen der Psychologie in der ersten Hälfte des zwanzigsten Jahrhunderts nicht Behaviorismus oder Psychoanalyse die im Hinblick auf die Kognitionswissenschaften haltbareren wissenschaftlichen Erkenntnisse erzielt haben, sondern die ganzheitlich ausgerichteten Gestaltpsychologen um Max Wertheimer, Wolfgang Köhler und Kurt Koffka [ES15].

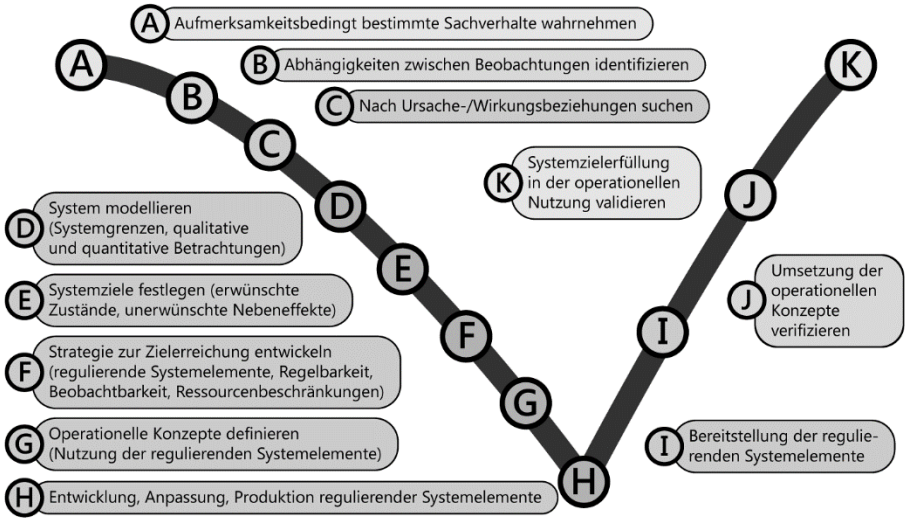

Bild 1. Systementwicklungsschritte

Erst nach Schritt A kann eine systematisch-logische Analyse des Problems einsetzen. Üblicherweise ergibt sich ein iterativer Prozess, in dem einzelne Stufen identifizierbar bleiben. Nachdem ein Sachverhalt Aufmerksamkeit auf sich gezogen hat (A), kann er in seinem Beobachtungskontext analysiert werden (B). Einzelne Ursache-/Wirkungsbeziehungen werden erkennbar (C). Es erfolgt nun der Übergang von der Problemanalyse zur Systemmodellierung. Zunächst sind Systemgrenzen zu ziehen und die qualitativen und quantitativen Abhängigkeiten in kohärente Zusammenhänge zu bringen (D). Um die ursprünglich identifizierten Probleme zu lösen, werden nun Systemziele hinsichtlich erwünschter Zustände und unerwünschter Nebeneffekte festgelegt (E). Ein wichtiger Schritt ist die Festlegung der Strategie zur Zielerreichung (F). Dies geht einher mit Überlegungen zur Systemarchitektur und der Definition der regulierenden Systemelemente, zu denen auch die neu zu entwickelnden oder anzupassenden technischen Systeme gehören. Hinsichtlich der regulierenden Systemelemente bleibt die Eignung zu untersuchen, ob die gewünschten Systemziele tatsächlich erreicht werden können. Sofern Beobachtbarkeit, Regelbarkeit und Ressourcenfragen abgeklärt sind, bleibt festzulegen, wie die technischen Systemelemente operationell betrieben werden sollen (G). Erst dann setzt die eigentliche Entwicklung der technischen Systeme ein (H).

In der klassischen, für die Anwendung expliziten Systems Engineerings charakteristischen Auftragsentwicklung findet zwischen den Stufen G und H eine Übergabe der Entwicklungsverantwortung für die technischen Systeme an Auftragnehmer statt. Erst, wenn die Auftragnehmer die technischen Systeme bereitstellen (I), können die entscheidenden Nachweise in Bezug auf das Gesamtsystem geführt werden. Zunächst muss in der Stufe J verifiziert werden, dass die technischen Systeme so genutzt werden können, wie in Stufe G definiert. Schließlich steht im tatsächlichen Einsatz die Bewährung aus, ob die Systemziele wirklich erreicht werden (K).

Die für die Auftragsentwicklung typische organisatorische Trennung zwischen Auftraggeber und Auftragnehmer impliziert starke betriebswirtschaftliche und vertraglich-juristische Einflüsse auf das Systems Engineering. Es ist deshalb nachvollziehbar, dass sich Standards und Lehrbücher zum Systems Engineering eher verhalten mit dieser Problematik auseinandersetzen. Leider wird häufig nicht deutlich, dass jede Produktentwicklung von Anfang an auf zwei gleichwertigen Architekturebenen gedacht werden muss. Zwar wird Vieles in den Überlegungen zu Systems-of-Systems angesprochen, doch vermischen Architecture Frameworks wie NAF [NA10] die Architekturebenen, da sie nicht auf die Einheit von Funktion, Struktur und Systemanforderungen für jedes System auf beiden Systemebenen als primäres Organisationsprinzip setzen [DS14-2], sondern vor Allem funktionale Schneisen durch die Systemarchitektur schlagen.

Wenn Systems Engineering in einem rein marktwirtschaftlichen Kontext zur Anwendung kommt, ergibt sich leicht ein suboptimales Systems Engineering. Zum Beispiel wird häufig nicht zwischen Stakeholder-Anforderungen auf übergeordneter Systemebene und Stakeholder-Anforderungen an das technische System unterschieden. Nachdem sich die Systemingenieure vor allem für das technische System verantwortlich fühlen, ist es

wahrscheinlicher, dass die Stakeholder-Anforderungen an das übergeordnete System bei Bedarf passend hingebogen werden, als dass die spezifischen Stakeholder-Anforderungen an das technische System beeinträchtigt werden. Die im Systems Engineering angestrebte Stakeholder-Orientierung ist dann nur eingeschränkt umgesetzt, wobei die Defizite sich gerade bei den Stakeholder-Anforderungen auf Kundenseite in späteren Lebenszyklusphasen einstellen. Juristisch ist die Furcht vor Nichterfüllung von Verträgen offensichtlich in der Regel größer als vor einer Sanktionierung in Bezug auf Produkthaftungs- und Produktsicherheitsgesetz. Das Produktsicherheitsgesetz verlangt in §3 neben einer Berücksichtigung des bestimmungsgemäßen auch eine des vorhersehbaren Gebrauchs [PSG11]. Zugegebenermaßen profitiert die Industrie hier häufig von einer ihr gegenüber freundlich gesonnenen Rechtsprechung, die zum Beispiel Autoherstellern erlaubt, dem vorhersehbaren Gebrauch von kombinierter automatischer Abstandshaltung und Geschwindigkeitsanpassung bei Nebel durch Warnhinweise in der Betriebsanleitung zu begegnen und keineswegs eine systemtechnisch umgesetzte Geschwindigkeitsbegrenzung bei schlechten Sichtverhältnissen einfordert. Appelle an Nachhaltigkeit sowie allgemeine Verantwortung für Gesellschaft und Umwelt genießen dann, wenig überraschend, nur eine nachrangige Priorität.

3 Systemkomplexität und Systemintegrität

Der Informationsgehalt der Aussage, dass die Komplexität in unserer Welt zunimmt, ist gering, da sie wohl kaum noch jemanden wirklich überrascht. In der Tat erscheint dieses Komplexitätswachstum schicksalhaft und unabwendbar. Die Globalisierung hat zu einer weltweiten Vernetzung in Bezug auf Warenaustausch, Finanztransaktionen und Informationsfluss geführt. Die große Zahl der sich in kurzen Zeiträumen gegenseitig beeinflussenden Kulturen, Staaten, Organisationen und Individuen macht Zusammenhänge und Wirkungsmuster unübersichtlich. Neben dieser intendierten Globalisierung existiert ebenso die Konkurrenz um die Ausbeutung natürlicher Ressourcen und die gegenseitige Beeinflussung durch Freisetzung von festen, flüssigen und gasförmigen Schadstoffen in global umweltgefährdenden Mengen. Hinzu tritt auch noch der evolutionäre Extremfall, dass eine Art auf unserem Planeten so dominant geworden ist, dass sie viele andere Arten verdrängen und dabei letztlich auch die eigenen Lebensgrundlagen zerstören kann.

Klassische ingenieurmäßige Ansätze, die Auswirkungen eines Systems auf seine Umgebung als kleine Störungen einzustufen, das System frei zu schneiden und den verbleibenden Betrachtungsgegenstand als geschlossenes System zu manipulieren, greifen nicht mehr. Aufgrund des hohen Nutzungsgrades der Umwelt durch den Menschen, müssen mehr und mehr Auswirkungen auf die Umwelt und deren natürlichen Grenzen von Anfang an berücksichtigt werden. Es bleibt nur zu hoffen, dass die menschlichen Fähigkeiten, die uns zur dominanten Spezies gemacht haben, auch hinreichend sind, die Auswirkungen unseres Tuns in den Griff zu bekommen, bevor unsere natürlichen Lebensgrundlagen zerstört sind. Systems Engineering kann aufgrund der inhärenten Stakeholder-Orientierung und den nachhaltigen Erfahrungen mit multidisziplinärer Zusammenarbeit zur Lösung globaler Probleme beitragen.

Komplexität ist aber nicht nur Schicksal. Es gibt eine zweite Quelle, unser Leben zu verkomplizieren: Was möglich erscheint wird auch getan. So versprach die zivile Nutzung der Kernspaltung saubere und billige Energie. Aktuell en vogue ist ein Heilsversprechen eines immer sichereren Verkehrsgeschehens durch neuartige autonome Funktionalität in technischen Systemen. Diese sozialen Konstruktionen von Wirklichkeit neigen dazu, die sich neu ergebenden Problemklassen zu übersehen. Dazu gehören die Endlagerung nuklearer Abfälle mit Halbwertszeiten, die um Größenordnungen über der bisherigen Dauer menschlicher Zivilisationsgeschichte seit Beginn von Ackerbau und Viehzucht liegen. Auch der Glaube, dass immer leistungsfähigere Automatisierung bald die hochdynamisch adaptiven Fähigkeiten menschlicher Kooperation substituieren können, wird sich wohl als Irrglaube mit in der Konsequenz totalitärem Anspruch erweisen. Die geschichtlichen Zusammenhänge von Futurismus und Faschismus in Italien lassen grüßen.

Bisher haben sich alle menschlichen Versuche, umfassende Zukunftsprognosen abzugeben, als fehlerhaft erwiesen. Auch Systemingenieure sind nicht im Besitz des Heiligen Grals oder des Steins der Weisen. Es ist also nicht zu erwarten, dass Systems Engineering jemals letztgültige Lösungen hervorbringt. Systems Engineering kann aber dazu beitragen, dass technische Systeme nachhaltig wirken. Nachhaltigkeit sei hier als konstituierendes Element von Systemintegrität eingeführt. Zur Nachhaltigkeit gehört zunächst einmal, dass der Nutzen eines Systems über den gesamten Lebenszyklus hinweg größer ist als der Aufwand. Diese Definition für Systemlebenszykluseffizienz ist zwar leicht dahingesagt, aber nicht vollständig und eindeutig quantifizierbar. Als zweites Kriterium lässt sich Nachhaltigkeit so einführen, dass ein System die während Konzeption und Entwicklung prognostizierte Systemlebenszykluseffizienz auch erreicht.

Zur Illustration sei an das politische Versprechen eines blauen Himmels über der Ruhr in den frühen siebziger Jahren des letzten Jahrhunderts erinnert. Die Maßnahme, die Abgaskamine der Industrie zu erhöhen, zeigte sich zwar als effizient, soweit es die Verbesserung der Luftqualität im Ruhrgebiet betraf, doch waren zehn Jahre später die Auswirkungen nicht mehr zu übersehen. Die großflächige Verfrachtung schwefelhaltiger Gase bereicherte die deutsche Sprache um den Begriff Waldsterben. Erst die Rauchgasentschwefelung führte zu einer weiträumigen und nachhaltigen Verbesserung der Luftqualität. Dieses Beispiel verdeutlicht zwei Prinzipien. Die Bewertung der Systemlebenszykluseffizienz über der Zeit mag starken Schwankungen unterworfen sein, und die Nachhaltigkeit lässt sich in der Regel erst spät im Systemlebenszyklus endgültig bewerten. Aufgrund schwieriger Quantifizierbarkeit helfen Verfahren der Schätztheorie, die bei derartige Problemstellungen durchaus Hinweise geben können, nur bedingt.

Ein anderer, nicht mathematischer Ansatz stammt aus der Theorie der kontinuierlichen Verbesserung. Der auch als Shewhart oder Deming Circle bezeichnete Vierphasenzyklus PDCA (Plan, Do, Check, Act) [ED00] lässt sich auch im Fall der Systemintegrität anwenden, siehe Abbildung 2. Die Phase *Plan* umfasst Konzeption, Entwicklung und Implementierung eines technischen Systems. Die Phase *Do* beschreibt dessen Nutzung in der realen Welt. *Check* bewertet die Systemintegrität und in der Phase *Act* werden weitere Maßnahmen definiert, die dann wieder in eine Verbesserung oder eine Neuentwicklung von technischen Systemen münden.

Bei Anwendung dieses Modells bezieht sich Systemintegrität nie auf die technischen Systeme, sondern immer auf die Wirkungen im übergeordneten System. Damit ergibt sich auch eine klare Abgrenzung gegenüber einer auf ein technisches System ausgerichteten Sicherheitsanalyse. Sicherheit bemüht sich darum, dass ein technisches System keinen Schaden im übergeordneten System anrichtet. Demgegenüber ist Systemintegrität damit befasst, dass das übergeordnete System selbst nachhaltig wirkt und bei unerwünschten Konsequenzen gegengesteuert wird. Eine hohe Systemintegrität äußert sich demnach auch in einer geringen Notwendigkeit nachsteuern zu müssen, um sich einstellenden ungewollten und schädlichen Effekten entgegen zu wirken.

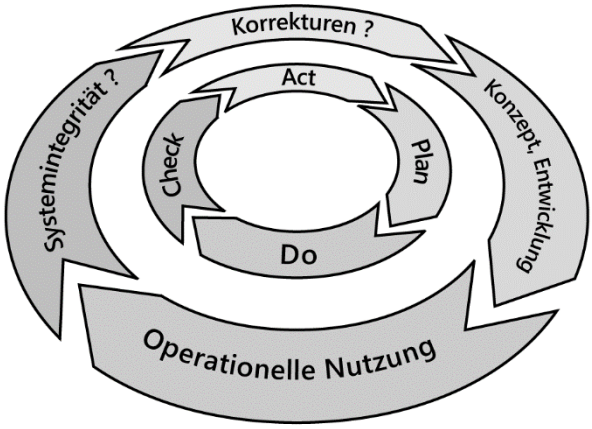

Bild 2. Systemlebenszyklus und kontinuierliche Verbesserung der Systemintegrität

4 Systemintegrität und Systemarchitektur

Über lange Phasen der technologischen Entwicklung konnten Sicherheit technischer Systeme und Systemintegrität im übergeordneten System weitgehend unabhängig voneinander betrachtet werden. Ausfallsichere Architekturen stehen dafür, dass Fehler in einem Systemelement erkannt und schädliche Auswirkungen auf das Gesamtsystem durch Selbstisolation verhindert werden. Ferner rekonfiguriert sich das Gesamtsystem, um weiterhin angemessene Funktionalität zur Verfügung zu stellen. Besondere Beachtung bedürfen die Transienten bis zur Fehlerisolation und während der Rekonfiguration des Gesamtsystems.

Zur Umsetzung dieser Prinzipien waren Flugregelungsanlagen in der Vergangenheit in einer Vorreiterrolle, die sich bis heute im Sprachgebrauch erhalten hat: Computer stürzen ab. Strenge Sicherheitsanforderungen, die Notwendigkeit einer multidisziplinären Entwicklung und die Unmöglichkeit eines vollständigen Sicherheitsnachweises im Rahmen der Entwicklung setzen hochfliegenden Visionen harte Grenzen. So lässt sich nachweisen, dass letztlich nur innovative technische Lösungen in Serie gegangen sind, die aufgrund operationeller Notwendigkeiten erforderlich waren [DS03].

Aufgrund der tolerierbaren katastrophalen Ausfallrate, werden elektronische Flugregelungsanlagen mehrkanalig ausgelegt. Die Mehrkanaligkeit unterstützt sowohl die Fehleridentifikation, die Begrenzung von Fehlertransienten als auch die Rekonfiguration. Um Fehlertransienten vor der Fehlerisolierung in ihrer Auswirkung auf das System zu begrenzen, wird in den Überwachungsstufen eine gewichtete oder ungewichtete Mittelwertbildung durchgeführt. Prinzipiell gilt, je mehr Kanäle an der Mittelwertbildung beteiligt sind, umso besser lassen sich die Auswirkungen von Fehlern begrenzen. Die Fehlerisolation muss einerseits Ausfälle eindeutig identifizieren. Auf der anderen Seite soll aber nicht jedes erwartbare Rauschen zu unnötigen Systemrekonfigurationen führen. Am besten gelingt der Kompromiss, wenn alle Kanäle gleichartig aufgebaut sind. Die Fehlererkennung erfolgt nach dem Mehrheitsprinzip. Dabei wird davon ausgegangen, dass in einem sehr kurzen Zeitintervall nur jeweils ein Fehler in einem Kanal auftritt. Die Redundanz ist erschöpft, wenn keine Mehrheitsbildung mehr möglich ist. Bei einer Pattsituation kann keine sichere Auswahl mehr getroffen werden. Folglich steht dieses System dann nicht mehr zur Verfügung. Das Mehrheitsprinzip arbeitet ohne eine Analyse der konkreten Ursache. Versuche, spezifische Fehler zu identifizieren und darauf angemessen zu reagieren, führen zu einem schnellen Komplexitätswachstum ohne Sicherheitsgewinn, da nie alle Fehlerszenarien a-priori identifiziert werden können. Ein ähnliches Schicksal war der Verwendung externer Daten zur Fehleridentifikation beschieden. Erstens, ergibt sich eine Frage nach der Integrität der externen Information. Zweitens, kann nicht ausgeschlossen werden, dass der Fehler im System die Verlässlichkeit der externen Information nicht unterminiert. Im Ergebnis nimmt die Sicherheit bei Verwendung externer Informationen in der Regel ab anstatt zu, da einer Ausweitung der Sicherheitsanalyse auf benachbarte Systeme in der Regel schon allein wirtschaftliche Grenzen gesetzt sind. Angesichts der begrenzten menschlichen Prognosefähigkeiten bei hochkomplexen Zusammenhängen sind zudem auch ein gewisses Maß an Bescheidenheit und Demut angebracht.

Eine besondere Herausforderung ergibt sich, wenn komplexe Funktionen in Software implementiert sind. Software fällt nicht aus, sondern allenfalls die Hardware, auf der die Software läuft. Software transformiert jede identische Zustandskombination immer zum gleichen Ergebnis. Schwierig wird es, wenn eine Zustandskombination zum ersten Mal im Betrieb auftritt und kein angemessenes Resultat erbringt. Eine Berücksichtigung derartiger Zustände ist in quantitativen Sicherheitsanalysen kaum möglich, da statistische Quantifizierungen weit weniger belastbar sind als bei Hardware-Ausfällen. Es ist in derartigen Situationen jedoch hilfreich, wenn der Pilot sich einstellende transiente Zustände beherrschen und das System manuell rekonfigurieren kann. Dies kann aber nur gelingen, wenn sich die Auswirkungen nach Amplitude und Frequenz so einstellen, dass der Mensch als übergeordnete Instanz überhaupt erfolgreich eingreifen kann. Sollte dies nicht möglich sein, bieten übergeordnete automatisierungstechnische Lösungen selten einen echten Sicherheitsgewinn.

Aus diesen spezifischen Erfahrungen lassen sich zwei allgemeine Lösungselemente für hoch-integre technische Systeme ableiten. Erstens sind ausfallsichere Systemarchitekturen zu bevorzugen, bei der der Ausfall eines technischen Systems durch Rekonfiguration im Gesamtsystem weitgehend aufgefangen werden kann. Zweitens, ist jedes technische System in Bezug auf die eigene Integrität als zellulärer Automat zu betrachten, der eigene

Fehler selbst identifiziert und isoliert. In der Regel ist dazu bei elektronischen Systemanteilen eine mehrkanalige Auslegung erforderlich. Alle anderen Maßnahmen können zwar in einzelnen Szenarien helfen, haben aber ein hohes Potential, diesen punktuellen Integritätsgewinn an anderer Stelle mehr als aufzuzehren.

5 Systemintegrität und der Systems-Engineering-Prozess

Die Schnittstelle zwischen der Integrität des übergeordneten Systems und der Sicherheit des technischen Systems manifestiert sich in den allokierten Anforderungen an das technische System und der Validierung des technischen Systems.

Im Rahmen einer Auftragsentwicklung werden die allokierten Anforderungen als Vertragsspezifikation verfügbar sein. In einem marktwirtschaftlichen Kontext mögen die allokierten Anforderungen eher aus einem Sammelsurium aus Stakeholder-Anforderungen sowohl an das übergeordnete als auch das technische System bestehen, die von Marketing- und Entwicklungsabteilungen zusammengetragen worden sind. Klarheit zwischen den Eigenschaften und Funktionen des übergeordneten Systems und des technischen Systems stellt sich so nicht ein. Dies belastet die Passung des technischen Systems in das übergeordnete System ebenso wie die Integrität des übergeordneten Systems.

Im Hinblick auf die Systemintegrität des übergeordneten Systems weisen allokierte Anforderungen an ein technisches System in der Regel Defizite auf. Allokierte Anforderungen konzentrieren sich auf das, was sein soll. Anforderungen zur Sicherheit werden pauschalisiert als Zahlenwerte angegeben, die von dem technischen System einzuhalten sind. Es wird jenseits katastrophaler Ereignisse zu wenig Gewicht auf eine spezifische Beschreibung der Dinge gelegt, die nicht sein sollen oder sein dürfen. Dies gilt im Übrigen gleichermaßen für die Erfassung der Stakeholder-Anforderungen und ist durch eine auch deshalb unvollständige Validierung nicht kompensierbar.

Die Qualität der auf das technische System allokierten Anforderungen und der zugehörigen Validierungsergebnisse sind für die Bewertung der Systemintegrität des übergeordneten Systems nur dann belastbar, wenn auch die Prozessqualität bei der Entwicklung des technischen Systems hohe Standards erfüllt. Bei schlechter Prozessqualität können Informationsverluste und Fehlinterpretationen an vielen Stellen im technischen System zu Problemen führen. Im übertragenen Sinn gibt es in so einem Heuhaufen viele Nadeln. Eine gute Prozessqualität wird niemals alle Irrtümer ausschließen können, aber sie kann dafür sorgen, dass die Zahl der Nadeln im Heuhaufen möglichst klein bleibt. Um eine gute Prozessqualität zu erreichen empfiehlt der Autor einen streng wertschöpfungskettenorientierten Entwicklungsprozess, der im Detail geplant und gesteuert wird und dessen Ergebnisse und Abhängigkeiten im Konfigurationsmanagement ebenso detailliert erfasst werden [DS14-1].

6 Schlussfolgerungen

Technische Systeme haben eine regulierende Wirkung in einem übergeordneten System, um definierte Ziele zu erreichen. Systemintegrität ist primär an das dem jeweiligen technischen System übergeordnete System gekoppelt. Systemintegrität ist ein Maß für Nachhaltigkeit im Sinne der Erfüllung von Erwartungen. Sie korreliert positiv mit der Systemlebenszykluseffizienz und negativ mit der Notwendigkeit von Korrekturen während des intendierten Systemlebenszyklus. Technische Systeme haben in der Regel maßgeblichen Einfluss auf die Systemintegrität des übergeordneten Systems. Insbesondere dürfen sie die Systemintegrität nicht gefährden, indem sie die eigene Funktionsfähigkeit überwachen und sich bei Teil- oder Totalausfällen selbst isolieren, ohne kritische Zustände im übergeordneten System auszulösen. Schließlich ist für die Entwicklung hoch-integrer technischer Systeme eine hohe Prozessqualität erforderlich.

Literaturverzeichnis

[BG15] Goldstein, E. B.: Cognitive Psychology: Connecting Mind, Research, and Everyday Experience. 4th Edition. Cengage Learning, Stamford, CT, 2015.

[BS76] Skinner, B. F.: About Behaviorism. Vintage Books, New York, NY, 1976.

[DK11] Kahnemann, D.: Thinking, Fast and Slow. Farrar, Straus and Giroux, New York, NY, 2011.

[DS03] Scheithauer, D.: Fly-By-Wire: Einsatzbereiche und Randbedingungen. Kooperationsforum Mechatronik für den Automobilbau, Nürnberg, 2003. (https://www.hitseng.eu/knowledge/pubs/downloads/fbw.pdf)

[DS14] Scheithauer, D.: Systems Engineering Value Stream Modelling. Proc. EMEA Systems Engineering Conference, Kapstadt, 2014.

[DS14] Scheithauer, D.: Qualität im System Design. In (Maurer, M.; Schulze, S.-O., Hrsg.): Tag des Systems Engineering, Carl Hanser Verlag, München, 2014.

[ED00] Deming, W. E.: Out of the Crisis. MIT Press, Cambridge, MA, 2000.

[ES15] Shiraev, E.: A History of Psychology: A Global Perspective. Sage Publications, Thousand Oaks, CA, 2015.

[HW08] Wußing, H.: 6000 Jahre Mathematik: Eine kulturgeschichtliche Zeitreise. Bd. 1. Springer-Verlag, Berlin Heidelberg, 2008.

[JM96] Mittelstraß, J. (Hrsg.): Enzyklopädie Philosophie und Wissenschaftstheorie. Verlag J. B. Metzler, Stuttgart Weimar, 1996.

[NA10] NATO (North Atlantic Treaty Organisation): NATO Architecture Framework. Version 3.1, 2010.

[PSG11] Gesetz über die Bereitstellung von Produkten auf dem Markt (Produktsicherheitsgesetz - ProdSG). Bundesrepublik Deutschland, 2011.

[RC32] Carnap, R.: Die physikalische Sprache als Universalsprache der Wissenschaft. *Erkenntnis* (2), S. 432-465, 1932. In (Stöltzner, M.; Uebel, T., Hrsg.): Wiener Kreis: Texte zur wissenschaftlichen Weltauffassung. Felix Meiner Verlag, Hamburg, 2006.

[WB05] Baum, W. M.: Understanding Behaviorism: Behavior, Culture, and Evolution. Blackwell Publishing, Malden, MA, 2005.