Qualität im System-Design

Dieter Scheithauer

Breitensteinstraße 26, 83727 Schliersee, dieter.scheithauer@hitseng.eu

Zusammenfassung: Maßgebende Texte und Standards über Systems Engineering beschreiben die Systems-Engineering-Methodik ohne speziell auf die besonderen Fragestellungen einzugehen, die sich für das Design eines einzelnen Systems oder Systemelements innerhalb einer mehrschichtigen Systemarchitektur ergeben. Entsprechend gibt es wenige Hinweise in der Literatur, wie weit Aufgaben und Verantwortung des Entwicklungsteams für ein spezifisches System oder Systemelement innerhalb der Systemarchitektur reichen. Dieser Frage wird im Folgenden nachgegangen. Es werden Qualitätsziele definiert. Deren Umsetzung im System-Design wird erläutert. Schließlich wird auf die zugehörige Nachweisführung eingegangen.

1 Einleitung

Diese Abhandlung widmet sich der Qualität im System-Design für ein einzelnes System beziehungsweise Systemelement in einer Systemarchitektur. Was an dieser Stelle unter Systemarchitektur verstanden wird, ist in Abbildung 1 dargestellt [SF13]. Über der Ebene des Gesamtsystems beschreibt die Systemumgebung den übergeordneten Kontext für den Systemlebenszyklus. Unterhalb des Gesamtsystems bildet sich die Systemarchitektur in den unterschiedlichen Zweigen aus. Sofern diese Systemelemente innerhalb der gleichen Organisation entwickelt werden, sind sie als abstrakte Systeme bezeichnet. Der Begriff des abstrakten Systems rechtfertigt sich aus der Überlegung, dass die Dekomposition der Systemarchitektur nach abstrakten Kriterien vorgenommen wird. Zu diesen Kriterien gehören neben einer sinnvollen funktionalen Strukturierung vielfältige Effizienzüberlegungen zu Kommunikation, Organisation und Wiederverwendbarkeit. Es sei angemerkt, dass abstrakte Systeme sehr wohl eine physikalisch abgrenzbare Einheit darstellen können, es aber nicht unbedingt müssen. Systemelemente, die von anderen Organisationen als standardisiertes oder nach spezifischen Anforderungen entwickeltes Produkt eingekauft werden, werden ungeachtet der dahinter liegenden, industriellen Differenzierungen global als Systemelemente auf der Implementierungsebene zusammengefasst.

Neben dem Streben nach Stakeholder-Zufriedenheit und wirtschaftlichem Erfolg leiten sich Qualitätskriterien für das Gesamtprodukt aus den Richtlinien der Europäischen Union über die Haftung für fehlerhafte Produkte [EC85] und über die allgemeine Produktsicherheit [EC01] ab. In Deutschland sind diese EU-Richtlinien durch das Produkthaftungsgesetz und das Produktsicherheitsgesetz in nationales Recht umgesetzt. Für eine adäquate Prozessqualität in der das jeweilige Produkt in Verkehr bringenden Organisation gibt die DIN EN ISO 9001 einen Rahmen [DIN09] für das Qualitätsmanagementsystem vor. Neben diesen allgemeingültigen Rahmenbedingungen

existieren weitere gesetzlichen Anforderungen und Regularien für spezifische Produktkategorien und Wirtschaftszweige. Für die Beschaffung der Systemelemente auf der Implementierungsebene sind diese Gesetze und Standards natürlich in entsprechender Weise auf die liefernden Unternehmen anwendbar. Zusätzlich zu den EU-Richtlinien ist die explizite Forderung nach einer Eingangsprüfung gemäß Paragraph 377 des Handelsgesetzbuches zu beachten.

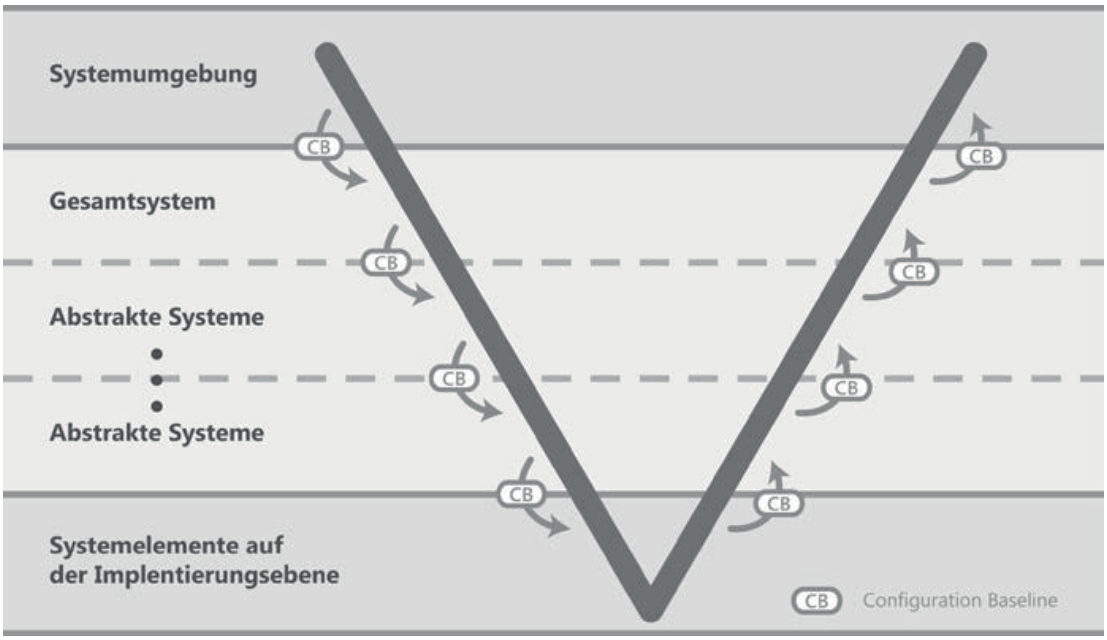

Bild1. Schematischer Aufbruch der Systemarchitektur

Für die Qualitätsanforderungen an abstrakte Systeme geben die referenzierten Gesetze und Normen zwar einen Rahmen vor. Ihre Anwendung auf abstrakte Systeme bedarf aber in der Regel einer weitergehenden Interpretation.

Nähert man sich den Qualitätsanforderungen an abstrakte Systeme nicht von der unternehmerischen, sondern aus der Systems-Engineering-Sicht, bleiben ebenfalls viele Fragen offen. Als repräsentatives Beispiel sei hier die ISO 15288 herangezogen [ISO08]. Hier werden Prozesse beschrieben, die in der vielfältigen Systems-Engineering-Literatur mit Methoden und Verfahren unterfüttert sind. Die gegebenen Anleitungen und Empfehlungen zielen auf eine globale Anwendbarkeit und geben Prinzipien vor, wie zum Beispiel vom Groben zum Feinen [HFV13]. Sie sind aber wenig konkret hinsichtlich der Frage, wie die Wertschöpfung im System-Design über die Systemarchitektur verteilt werden sollte, da die Prinzipien und ihre methodische Umsetzung diesen Aspekt allenfalls am Rande berühren.

Ein weiterer Ansatz zum Thema Qualität im System-Design ist im Systemdenken zu finden. Aus ethischen Überlegungen heraus lassen sich natürlich auch Antworten auf die Frage, was sollen wir tun, entwickeln [JS10]. Der Weg zur konkreten Umsetzung ist aber noch weiter als der von Systems-Engineering-Prinzipien ausgehende.

2 Qualitätsziele

Die Qualitätsziele für das System-Design abstrakter Systeme ergeben sich zunächst aus den Verpflichtungen des Entwicklungsteams des spezifischen abstrakten Systems gegenüber den anderen Entwicklungsteams, die an anderen Systemen und Systemelementen in der Systemarchitektur arbeiten. Primär sind dies das Entwicklungsteam des übergeordneten Systems auf der nächsthöheren Architekturebene und die Entwicklungsteams der Systemelemente des abstrakten Systems auf der direkt darunter liegenden Architekturebene. Die von der nächsthöheren Architekturebene allokierten Anforderungen müssen umgesetzt werden und führen schließlich zu allokierten Anforderungen für die Systemelemente auf der nächstfolgenden Architekturebene. Für die allokierten Anforderungen an die Systemelemente gilt, dass alle Systemelemente im vorgegebenen Zeit- und Kostenrahmen realisierbar sein müssen. Diese beiden Qualitätsziele beschreiben den Fluss freigegebener Informationen im System-Design innerhalb der Systemarchitektur in einem einfachen Modell [SF13]. Sie fokussieren die Verantwortlichkeit des für ein abstraktes System zuständigen Entwicklungsteams auf den Hauptinformationsfluss, aber beschreiben diese Verantwortlichkeit nicht vollständig. Insbesondere lässt dieses Modell die eigentliche Wertschöpfung, für die das Entwicklungsteam zuständig und verantwortlich ist, außer Acht.

Die Eigenverantwortung erstreckt sich vor Allem auf die sorgfältige und gewissenhafte Anwendung eigenen Wissens, eigener Erkenntnisse und eigener Erfahrungen auf das Design des spezifischen abstrakten Systems. Im Kontext einer hochentwickelten Industriekultur, die auf den Prinzipien Arbeitsteilung, Spezialisierung und Standardisierung beruht, muss darauf vertraut werden können, dass jedes Entwicklungsteam ein Maximum an Kompetenz auf den beteiligten Technikfeldern und deren Integration für das jeweilige abstrakte System besitzt und weiterentwickelt. Da im Allgemeinen niemand, auch innerhalb desselben Unternehmens, als kompetenter angesehen werden kann, kann diese Eigenverantwortung weder abgeschoben noch umgangen werden. Insofern wirkt der Tenor aus Produkthaftungsgesetz, Produktsicherheitsgesetz und der Pflicht zur Eingangsprüfung gemäß Handelsgesetzbuch auf jedes Entwicklungsteam eines abstrakten Systemelementes fort.

Daraus ergibt sich eine Reihe von Folgerungen. Erstens, jedes Entwicklungsteam ist den Unternehmenszielen auch direkt verpflichtet. Mit Bezug auf das Qualitätsziel Kunden und sonstige Stakeholder zufriedenzustellen, hat jedes Entwicklungsteam den Sinngehalt der auf das betreffende Systemelement allokierten Anforderungen hinsichtlich Kunden- und Stakeholder-Bedürfnissen zu erfassen. Manche spezifischen Stakeholder-Bedürfnisse mögen besser verstanden werden als von den Entwicklungsteams, die für übergeordnete Systeme in der Systemarchitektur zuständig sind, und von den Entwicklungsteams der Systemelemente, für die diese Bedürfnisse nicht mehr voll sichtbar sind. Zusätzlich mögen Design-Entscheidungen, zum Beispiel im Bereich der Mensch-Maschine-Schnittstellen, zu weiteren Stakeholder-Bedürfnissen führen.

Zweitens, jedes Entwicklungsteam hat eine wertschöpfende Aufgabe. Die Definition einer spezifischen Systemarchitekturebene oder eines spezifischen abstrakten Systems in

der Systemarchitektur macht nur Sinn, wenn auf dieser Ebene oder für dieses spezifische abstrakte System emergente Funktionen und Eigenschaften gestaltet werden, die sich aus den Funktionen und Eigenschaften der nachgeordneten Systemelemente nicht mehr automatisch ergeben. Dies reflektiert einen wesentlichen Grundsatz im Systemdenken. Ohne diese Wertschöpfung verkommt das Design eines abstrakten Systems zu einer rein administrativen Aufgabe, die die Sinnfälligkeit einer gewählten Systemarchitektur in Frage stellt. Es ist einleuchtend und zwingend, dass ein Entwicklungsteam eines abstrakten Systems Verantwortung für die eigene Wertschöpfung übernehmen muss.

Drittens, die wertschöpfende Natur des System-Designs führt zu Detaillierungen, die aus den erhaltenen allokierten Anforderungen nur eingeschränkt oder auch überhaupt nicht deduzierbar sind. Aus diesem Grund hat jedes Entwicklungsteam eine Verpflichtung auch gegenüber den Entwicklungsteams in anderen Zweigen der Systemarchitektur, zu denen inhaltliche Schnittstellen bestehen. Die inhaltlichen Schnittstellen sind gemeinsam auszugestalten und abzustimmen.

3 Vierstufenmodell des System-Designs

Die Vorgehensweise im System-Design kann anhand eines Vierstufenmodells erklärt werden, wie in Abbildung 2 dargestellt. In den nachfolgenden Unterkapiteln werden diese vier Stufen detaillierter beschrieben. Zuvor sei die Aufmerksamkeit aber auf die drei fundamentalen Sichten eines Systems gelenkt, die zusammen für eine komplementäre, konsistente und bei Abschluss des System-Designs auch vollständige Beschreibung des Systems erforderlich sind. Es sind dies die Systemanforderungen, die funktionale Beschreibung und die Architekturbeschreibung.

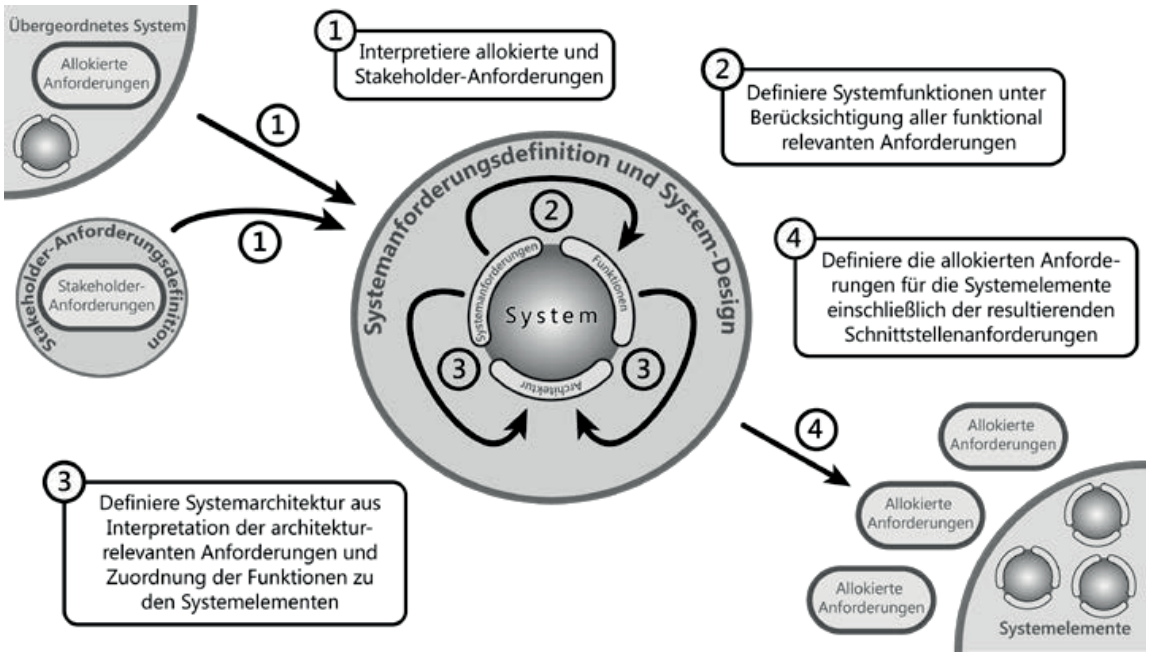

Bild 2. Vierstufenmodell des System-Designs

Die Bedeutung dieser Sichten beruht letztendlich auf der Arbeitsweise des menschlichen Gehirns [DK11; BG10]. Die funktionale Beschreibung bildet sich weitgehend auf den schnellen Denkkanal ab, der uns hilft das Leben in Echtzeit zu meistern. Das schnelle Denken bündelt unser Wissen und unsere Erfahrungen und erlaubt eine umgehende Reaktion auf Umweltreize. Schnelles Denken ist in hohem Maße assoziativ und entzieht sich in weiten Teilen einer bewussten Steuerung. Langsames Denken hingegen hilft uns der Welt eine Bedeutung zu geben, indem wir Differenzierungen und Attributierungen vornehmen, was zum Beispiel anhand der sprachlichen Entwicklung eines Kindes beobachtbar ist. Langsames Denken ist deshalb weitgehend immer bewusstes Denken. Langsames Denken ist für die Hervorbringung neuer Ideen und innovativer Lösungen unabdingbar, sofern man von einem reinen Versuch-und-Irrtum-Ansatz absieht. Die Architekturbetrachtung wurzelt letztlich im langsamen Denken. Aus diesen beiden Denkkanälen baut jeder Mensch für sich seine Sicht auf die Welt auf.

Kulturelle Prägungen sorgen für Ähnlichkeiten in den individuellen Weltsichten. Es wäre aber vermessen, hieraus zu schließen, dass zwei Menschen absolut identische Vorstellungen von der Welt entwickeln. Hier kommt den Systemanforderungen als dritte Sicht auf ein System eine wichtige Rolle für eine erfolgreiche zwischenmenschliche Kommunikation zu. Systemanforderungen geben Hinweise auf die essentiellen und wichtigen Funktionen und Eigenschaften eines Systems. Die Systemanforderungen drücken letztendlich aus, für welche Funktionen und Eigenschaften das Entwicklungsteam Verantwortung übernimmt.

Im System-Design werden alle Sichten mehr oder weniger parallel entwickelt. Dies kann das Vierstufenmodell nicht abbilden. Es hilft im Gegenteil, bei allem kreativem Chaos Ordnung und Konsistenz zu wahren. Die Abfolge im Vierstufenmodell beschreibt die Reihenfolge, in der Entscheidungen festgeschrieben werden sollten, und die Freigabesequenz für die die jeweiligen Informationen enthaltenen Dokumente. Typischerweise wird das Vierstufenmodell mehrfach iterativ durchlaufen, um zu einem vollständigen und qualitativ hochwertigem System-Design zu kommen.

3.1 Systemanforderungsanalyse

Am Anfang der Systemanforderungsanalyse werden die allokierten Anforderungen und etwaigen zusätzlichen Stakeholder-Anforderungen zusammengetragen. Die Verständlichkeit der Anforderungen und ihre Sinnfälligkeit jeweils für sich genommen und im Kontext mit allen anderen Anforderungen sind zu analysieren. Hierzu gehört auch eine Untersuchung des Kontextes jeder einzelnen Anforderung. Zum einen sollte jede Anforderung bis zur ursprünglichen Stakeholder-Anforderung zurückverfolgt werden. Zum anderen ist es aber auch notwendig in anderen Zweigen der Systemarchitektur nach Auswirkungen der initialen Stakeholder-Anforderung zu suchen, um einen vollständigen Überblick über die Systemschnittstellen zu erhalten. Sofern Zweifel entstehen, ob die allokierten Anforderungen oder zusätzlichen Stakeholder-Anforderungen die dahinter liegenden Stakeholder-Bedürfnisse wirklich treffend beschreiben, ist diesen Zweifeln nachzugehen.

Soweit erforderlich, werden die allokierten Anforderungen und weiteren Stakeholder-Anforderungen interpretiert und verfeinert, um in der Fachsprache der im jeweiligen Systementwicklungsteam vertretenen Fachdisziplinen zu präzisen Aussagen zu gelangen. Widersprüchliche und im Wettbewerb zueinander stehende Systemanforderungen müssen identifiziert werden. Widersprüche sind aufzulösen. Für im Wettbewerb zueinander stehende Systemanforderungen sind die Auswirkungen auf das System-Design zu analysieren und bei unbefriedigenden Kompromissen ebenfalls Abhilfemaßnahmen einzuleiten.

In diesem Schritt kann nicht erwartet werden, dass die Systemanforderungen schon ihre letztgültige Form erreicht haben. Die funktionale Analyse und die Architekturdefinition sind mächtige Schritte, um ein tieferes Systemverständnis zu gewinnen, was sich auch in der weiteren Evolution der Systemanforderungssicht niederschlagen wird. Die in der Literatur häufig vorgeschlagenen Systemanforderungs-Reviews bergen die Gefahr, die Zahl der Systemanforderungen einseitig in die Höhe zu treiben. Insbesondere Reviewer, die nicht zum Entwicklungsteam gehören, verlangen häufig die explizite Niederlegung von Sachverhalten, die in späteren, nicht-verbalen Design-Modellen klarer und effizienter ausgedrückt werden können.

3.2 Funktionale Analyse

Die funktionale Analyse geht von den Systemanforderungen aus. Einzelne Systemanforderungen benennen erforderliche Funktionen explizit. Viele andere Systemanforderungen haben aber auch Implikationen auf das funktionale Design. Von einer simplen Unterscheidung funktionaler von nichtfunktionalen Anforderungen kann hier nur gewarnt werden, da dies in fast allen Fällen zu deutlich höheren Lebenszykluskosten führt. Es ist zielführender, für jede Funktion die für diese spezifische Funktion relevanten Systemanforderungen eindeutig zu identifizieren. Dies erlaubt zunächst eine objektive Überprüfung, ob wirklich alle relevanten Systemanforderungen berücksichtigt worden sind. Ferner ermöglicht die Verfolgbarkeit von Systemanforderungen zu Funktionen eine effiziente Nachweisführung. Tests werden immer funktionsorientiert geplant und durchgeführt. Andererseits besteht in vielen Anwendungsdomänen eine explizite Pflicht zur anforderungsbasierten Nachweisführung. Darüber hinaus ergibt sich für alle Anwendungsdomänen aus den gesetzlichen Rahmenbedingungen eine Notwendigkeit für jedes Unternehmen, hinreichend Evidenz für die eigene Sorgfalt zu generieren. Test-Design und der Nachweis der Anforderungserfüllung profitieren gleichermaßen, wenn alle funktionsrelevanten Systemanforderungen zu Funktionen und Subfunktionen eindeutig verfolgbar sind.

Die funktionale Modellierung schließt neben den Systemfunktionen auch eine Abbildung der Systemumgebung mit ein. Auch wenn die Modellierung zunächst dem System-Design dient, sollte die weitere Nutzung der Modelle zu Nachweiszwecken mit bedacht werden. Eine modell-basierte, virtuelle Systemintegration verspricht für die Zukunft in mehrfacher Hinsicht deutliche Effizienzsteigerungen. Sie erlaubt eine fortlaufende Systemvalidierung am virtuellen Produkt, die bereits vor der eigentlichen

Systemimplementierung einsetzen kann. Die fortlaufende Evolution der Design-Modelle führt zur qualitativ hochwertigen Darstellung von Systemumgebungen in der Systemintegration. Und mit validen Systemmodellen können schließlich risikoreiche Testfälle am realen Produkt substituiert werden.

In Bezug auf die ursprüngliche Aufgabenstellung, ergibt sich für abstrakte Systeme die Frage, wie weit die funktionale Modellierung ins Detail gehen soll? Als Leitlinie lässt sich postulieren, dass die funktionale Modellierung die Design-Entscheidungen auf der jeweiligen Architekturebene unterfüttern muss. Im Einzelfall kann dies zu sehr detaillierten Modellen führen, um die Auswirkungen einzelner Design-Entscheidungen in tieferen Schichten der Systemarchitektur untersuchen zu können. Schließlich steht das Entwicklungsteam in der Pflicht, die Implementierbarkeit der Systemelemente zu gewährleisten. Wenn jedoch sehr detaillierte Modelle den Systemelementen als verpflichtend aufoktroyiert werden, kann dies die Gestaltungsfreiheit der Entwicklungsteams untergeordneter Systemelemente unnötigerweise einschränken und zu ineffizienten Lösungen führen. Aus diesem Grund sollte genau unterschieden werden zwischen den Modellen, die der Absicherung der eigenen Design-Entscheidungen dienen, und solchen, die an die Entwicklungsteams der Systemelemente mit striktem Anforderungscharakter weitergereicht werden. Ungeachtet dessen sollten auch die detaillierteren Design-Modelle für die Entwicklungsteams der Systemelemente als weiterführende Information sichtbar sein.

Wie bereits oben erwähnt ergeben sich aus der funktionalen Analyse weitere Systemanforderungen. Diese gewährleisten die Komplementarität und Konsistenz von funktionaler Sicht und Systemanforderungssicht und vervollständigen die Qualitätskriterien für das System.

3.3 Architekturdefinition

Die Architekturdefinition führt zur Definition der Systemelemente und deren Schnittstellen. Zum einen ergibt sich die Architektur aus einer sinnvollen Zuordnung der Funktionen zu Systemelementen. Schnittstellen sollen minimiert und sinnvoll gewählt werden. Einzelne Systemelemente sollen gut implementierbar sein, um spätere Risiken zu vermeiden. Zum anderen sind aber auch einzelne Systemanforderungen für sich genommen architekturrelevant. Als Beispiele für architekturrelevante Systemanforderungen seien Anforderungen zur Segregation von sicherheitskritischen Systemanteilen von nichtsicherheitskritischen und die gewünschte Verwendung vorhandener oder standardisierter Systemelemente genannt. Alle architekturrelevanten Systemanforderungen sollten eindeutig identifiziert werden, um Komplementarität, Konsistenz und Vollständigkeit der drei Sichten auch zwischen Systemanforderungen und Architekturentscheidungen nachweisen zu können. Dies kann auch zur Definition weiterer Systemanforderungen führen.

Angesichts von domänenspezifischen oder firmeneigenen Standardarchitekturen werden die Gestaltungsmöglichkeiten in der Architekturdefinition häufig unterschätzt und nicht genutzt. Die Aufteilung der Funktionen auf Systemelemente und der Umgang mit architekturrelevanten Systemanforderungen bekommen dann einen mechanistischen und

vorwiegend administrativen Charakter. Traditionelle Systemarchitekturen werden jedoch beständig von zwei Seiten herausgefordert.

Auf der einen Seite nimmt die Integrationsdichte in technischen Produkten beständig zu, um die Gesamtsystemeffizienz zu optimieren. Zunehmende Integrationsdichte erhöht die Komplexität der Schnittstellen zwischen den Systemelementen in der Systemarchitektur. Demzufolge wird die Kommunikation zwischen den verschiedenen Entwicklungsteams schwieriger. Es besteht die Gefahr, dass die etablierten Kommunikationsmechanismen überlastet werden und Schnittstellenprobleme spät evident werden.

Auf der anderen Seite führen technische Innovationen zur Minimierung, so dass der Aufwand für die Implementierung einzelner Funktionen schwindet. Die Existenz einzelner Systemelemente als separate Entwicklungsgegenstände kann dann infolge der geringeren Wertschöpfung nicht mehr gerechtfertigt sein.

Beide Entwicklungen zusammen erhöhen die Gefahr, von weniger traditionell denkenden und handelnden Wettbewerbern überholt zu werden, wenn die Chancen der Architekturgestaltung im Zusammenhang mit technischen Fortschritten vernachlässigt werden. Technologische Durchbrüche mit Markterfolg sind in der Regel mit der Änderung von Systemarchitekturen verbunden.

3.4 Zuweisung von Anforderungen zu den Systemelementen

Die Allokation von Anforderungen zu den Systemelementen ist im Vierstufenmodell klar gegenüber der Architekturdefinition als separate Aktivität abgegrenzt, da sich die Perspektive für das Entwicklungsteam ändert. In der Architekturdefinition sind die Entscheidungen weiterhin auf die Funktionen und Eigenschaften des abstrakten Systems fokussiert. Die Zuweisung von Anforderungen zu den Systemelementen verlangt demgegenüber ein Eingehen auf die Entwicklungsteams der Systemelemente. Während Systemanforderungen die Übernahme von Verantwortung kennzeichnen und die Basis für die Verifikation bilden, dienen die allokierten Anforderungen der Kommunikation. Entsprechend müssen die allokierten Anforderungen so formuliert sein, dass sie für die Entwicklungsteams der Systemelemente sinngebend sind.

Die Konsistenz zwischen Systemanforderungen, funktionaler Beschreibung und Architekturbeschreibung muss sich auch in den Zuordnungen der Systemanforderungen zu den allokierten Anforderungen der Systemelemente widerspiegeln. Funktionsrelevante Systemanforderungen sind auf die Systemelemente herunter zu brechen, denen die entsprechenden Funktionen zugeordnet werden. Architekturrelevante Systemanforderungen sind je nach ihrer Anwendbarkeit an die entsprechenden Systemelemente weiterzureichen. Einzelne architekturrelevante Systemanforderungen mögen für die Systemelemente nicht mehr anwendbar sein. Wenn zum Beispiel alle sicherheitskritischen Funktionen von den nichtsicherheitskritischen bei der Zuordnung auf die Systemelemente vollständig separiert sind, kann eine entsprechende Systemanforderung als konsumiert gelten. Unter den Systemanforderungen mag es Anforderungen geben, die weder als funktionsrelevant noch als architekturrelevant eingestuft wurden. Dies bedeutet nur, dass sie für das Design des abstrakten Systems

selbst keine Rolle gespielt haben. Derartige Anforderungen sind den Systemelementen ebenfalls je nach Anwendbarkeit zuzuordnen.

Wenn alle vier Schritte des Vierstufenmodells durchlaufen sind, sollte ein Design-Review durchgeführt werden, bevor die allokierten Anforderungen für die Systemelemente an die entsprechenden Entwicklungsteams als verbindliche Vorgabe weitergegeben werden. Ziel des Design-Reviews ist die Überprüfung von Komplementarität, Konsistenz und Vollständigkeit der Systemanforderungssicht, der funktionalen Sicht und der Architektursicht. Neben dem Systementwicklungsteam selbst, das seine Systemlösung vorstellen und verteidigen muss, sind aus technischer Sicht mindestens die Entwicklungsteams des übergeordneten Systems und aller direkten Systemelemente zu beteiligen. Das Entwicklungsteam des übergeordneten Systems bewertet in erster Linie, ob die eigenen Vorgaben und Intentionen in der Systemlösung entsprechend berücksichtigt sind. Die Entwicklungsteams der Systemelemente lernen den Systemkontext ihrer Systemelemente kennen und müssen klären, ob die ihnen vorgegebenen allokierten Anforderungen aus ihrer Sicht hinreichend und unter den weitergehenden Rahmenbedingungen hinsichtlich Zeit und Kosten erfüllbar erscheinen.

4 Nachweisführung

Ein qualitätsvolles System-Design ergibt sich in erster Linie aus den Design-Aktivitäten selbst. Dedizierte Nachweisaktivitäten sind dennoch für eine umfassende Nachweisführung unerlässlich. Zum einen wird so eine unabhängige Prüfung gewährleistet. Zum anderen bietet sich erst nach Abschluss der Design-Aktivitäten die Möglichkeit, eine Gesamtbewertung des Designs vorzunehmen, um Vollständigkeit, Ausgewogenheit und Konsistenz des Designs zu bewerten.

Die design-begleitenden Nachweisaktivitäten konzentrieren sich auf eine Überprüfung der Sinnfälligkeit des System-Designs. Dementsprechend überwiegen die Validierungstätigkeiten, um zu überprüfen, inwieweit das System-Design Stakeholder-Bedürfnisse befriedigt. In einem früheren Aufsatz wurde eine defensive, gestufte Validierung empfohlen [SF13]. Im Rahmen des System-Designs spielen dabei eine Validierung der Stakeholder-Anforderungen sowie eine Validierung der vorgegebenen allokierten Anforderungen eine Rolle. Beide Validierungsstufen dienen der Überprüfung des Kommunikationserfolges zwischen unterschiedlichen Personengruppen.

Mit Abschluss der Design-Aktivitäten lassen sich alle drei Sichten zur Validierung heranziehen, um die Sinnfälligkeit des Designs zu bewerten sowie die Abwesenheit unerwünschter Funktionen und Eigenschaften zu untersuchen. Sobald Design-Ergebnisse für die Systemelemente verfügbar werden, können im Rahmen einer virtuellen Integration die Implikationen berücksichtigt werden, die aus dem Design der tieferliegenden Systemelemente resultieren.

Die im Rahmen der Systemintegration durchgeführte Verifikation erbringt schließlich den Nachweis, dass die Systemanforderungen korrekt und vollständig implementiert sind. Verifikation ist insbesondere für den Nachweis der Vertragserfüllung bedeutsam.

Verifikation allein liefert aber keinen hinreichenden Nachweis im Sinne der gesetzlichen Anforderungen zu Produkthaftung und Produktsicherheit, da Verifikation nicht auf die Falsifizierung des Systems ausgerichtet ist. Die Untersuchung auf die Abwesenheit ungewollter Systemfunktionen und Systemeigenschaften liegt allein im Zuständigkeitsbereich der Validierung.

5 Zusammenfassung

Die Verantwortlichkeit des Entwicklungsteams eines Systems oder Systemelementes in der Systemarchitektur reicht über die pure Umsetzung vertraglicher Anforderungen hinaus. Folglich befreit ein Verweis auf die Defizite vorgegebener allokierter Anforderungen nicht von der Eigenverantwortung eines Entwicklungsteams für Mängel der eigenen Systemlösung. Methodische Grundlage zur Wahrnehmung der Eigenverantwortung ist die Erzeugung komplementärer und konsistenter Sichten hinsichtlich Systemanforderungen, Funktionen und Architektur. Letztendlich müssen diese Sichten zusammen zu einer kompletten Beschreibung der auf der jeweiligen Architekturebene evidenten Systemfunktionen und Systemeigenschaften führen. Die Systemanforderungen übernehmen dabei die Rolle, alle wichtigen Systemfunktionen und Systemeigenschaften zu kommunizieren, für die das Entwicklungsteam verantwortlich zeichnet, bis hin zur Justiziabilität.

Literaturverzeichnis

[BG10] Goldstein, B.: Sensation and Perception. Eighth Edition. Wadsworth, Cengage Learning, Belmont, CA, 2010.

[DIN01] DIN EN ISO 9001:2008-12: Qualitätsmanagementsysteme – Anforderungen. Beuth Verlag, Berlin, 2008.

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

[EC01] 2001/95/EG: Richtlinie des Europäischen Parlaments und des Rates vom 3. Dezember 2001 über die allgemeine Produktsicherheit.

[EC85] 85/374/EWG: Richtlinie des Rates vom 25. Juli 1985 zur Angleichung der Rechts- und Verwaltungsvorschriften der Mitgliedstaaten über die Haftung für fehlerhafte Produkte.

[HFV13] Haberfellner, R, de Weck, O., Fricke, E., Vössner, S.: Systems Engineering – Grundlagen und Anwendung. 12. völlig neubearbeitete und erweiterte Auflage. Orell Füssli Verlag, Zürich, 2012.

[ISO08] ISO/IEC 15288-2008: Systems and Software Engineering – System Life Cycle Processes. 2008.

[JS10] Jahns, V.; Siebert, H.-G.: Ethik als Grundlage Sozialer Kompetenz im Systems Engineering. Tag des Systems Engineering, 2010.

[SF13] Scheithauer, D.; Forsberg, K.: V-Model Views. Proc. 23rd INCOSE International Symposium, Philadelphia, PA, 2013.