

Studienarbeit

„Die Rolle der Systemarchitektur in der Entwicklung“

für die Abschlussprüfung zum Certified Systems Engineers (GfSE)[®] Ebene A

Januar 2024

Vorgelegt von

Sascha Ackva
St.-Stephans-Weg 8
89340 Leipheim

Durchführungsort: Leipheim

Datum: 26.01.2024

Inhaltsverzeichnis

Einleitung.....	1
Aktuelle Beobachtungen.....	2
Die Architektur über den Produktlebenszyklus	3
(SysML) Modellierung und Systemarchitektur	6
Systemarchitektur und die technischen Prozesse	9
Systemarchitektur und die technischen Management Prozesse	12
Die Projektplanung und ihre Strukturpläne.....	12
Betrachtung weiterer technischer Management Prozesse	15
Teamwork Aspects	17
Zusammenfassung	18
Literaturverzeichnis	19

Einleitung

Die Systemarchitektur wird oft als rein technisches Arbeitsprodukt im Systems Engineering verstanden, analog zu dem Statement: „Der Prozess Architekturgestaltung dient der Erarbeitung und Erstellung alternativer Architekturen, [...] um die passenden technologischen oder technischen Systemelemente für das System bestimmen zu können.“¹ Es geht also augenscheinlich nur um die Bestimmung technischer Elemente ohne einen Hinweis auf eine koordinierende Rolle der Architektur. Aber wird ein solches Verständnis der Bedeutung der Systemarchitektur gerecht?

Der Titel der Arbeit bezieht sich auf die globale Betrachtung von Entwicklungsorganisationen innerhalb einer Firma und damit der Rolle der Architektur in einem breiteren Kontext. Ein Fokus ausschließlich auf die Entwicklungsphase eines Systems wäre hier nicht angebracht und wäre mitunter eine Ursache für die hier dargestellten Beobachtungen und Probleme. Die Betrachtung der Architektur in dieser Arbeit bezieht sich damit auf den gesamten Produktlebenszyklus.

¹ Walden et al., INCOSE Systems Engineering Handbook v.4, Deutsche Übersetzung, San Diego 2015, Seite 95

Aktuelle Beobachtungen

Im Projektalltag, so beobachtet im Bereich der Aerospace & Defence, wie auch der Automotive Industrie – wenn auch in unterschiedlichen Ausprägungen –, sind heute Arbeitsweisen beobachtbar, die der Systemarchitektur eine eher untergeordnete Bedeutung zukommen lassen. Diese sind u.a.

- die Systemarchitektur zu erstellen ist alleinige Aufgabe eines Systemarchitekten. Sie wird nicht als gemeinschaftliche Aufgabe eines Entwicklungsteams verstanden. Sie bildet keinen „Single-Point-of-Truth“ und wird als solche im Projektverlauf nicht weiter genutzt.
- Die Systemarchitektur wird teilweise aus dem Entwicklungsprozess „getailored“, da ja schon „alles“ bekannt ist. Dies gilt vor allem bei sogenannten Anpassungsprojekten bestehender Plattformen.
- Die Systemarchitektur braucht man (nur), um das Quality-Audit zu bestehen. Dies indiziert eine Prozesstreue in der Arbeit, ohne allerdings den Zweck der Aktivität verstanden zu haben oder davon zu profitieren.
- Systems Engineering und insbesondere die Beschreibung der Systemarchitektur wird nicht als Wertbeitrag im Projekt gesehen, sondern nur die Entstehung von Mechanik-, Elektronik- und Softwarekomponenten.
- Die Systemarchitektur wird in der Konzeptphase nur informell, z.B. in Form von Präsentationen dokumentiert und erst mit Beginn der Entwicklungsphase formal und strukturiert erfasst.

Die Architektur über den Produktlebenszyklus

Der generische Produktlebenszyklus eines Systems lässt sich wie unten abgebildet darstellen. Dabei wird hier Wert gelegt auf die prinzipielle logische Abfolge und auch der Parallelität der Lebenszyklusphasen. Explizit nicht dargestellt ist hier ein zeitlicher Maßstab, noch die Wiederholung von Phasen durch notwendige Anpassungsentwicklungen eines Systems über die Nutzungsdauer.

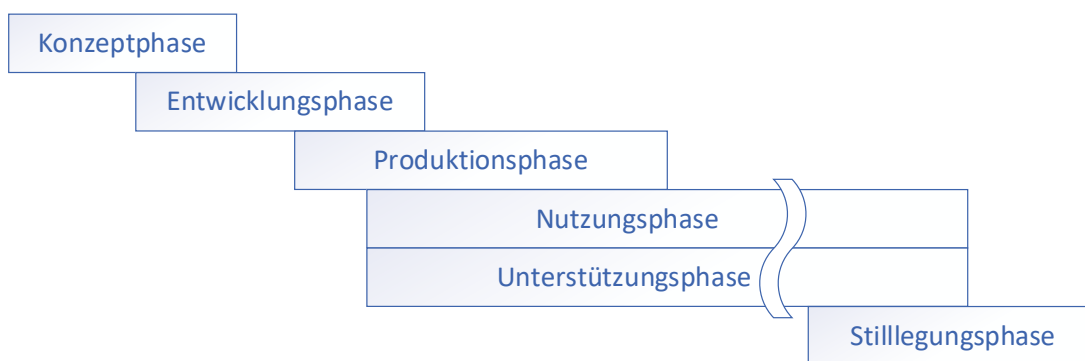


Abbildung 1: Generisches Produktlebenszyklusmodell gemäß ISO/IEC/IEEE 15288:2023²

Vergleicht man die Lebenszyklusphasen eines Systems in Bezug auf die Architektur, lässt sich folgendes feststellen:

Während der **Konzeptphase** liegt der Fokus auf dem Erfassen und der Analyse der Stakeholder-Bedarfe und vorliegenden Randbedingungen sowie dem abwägen möglicher Lösungsansätze. Diese reichen von operationellen Nutzungskonzepten, über dazu nötige Funktionsabläufe, die architekturellen Grundprinzipien bis hin zu detaillierten Leistungsabschätzungen in kritischen Teilbereichen. Hierbei ist zu beachten, dass als Ergebnis eine ausgewählte, vorläufige Lösungsarchitektur mit Schlüsselanforderungen an das System vorliegt, die geeignet ist erste Abschätzungen hinsichtlich Zeit und Kosten abzuschließen³, um z.B. einen Angebotsprozess abzuschließen.

² vgl. Walden et al., INCOSE Systems Engineering Handbook v.5, Abbildung 2.1, Seite 26

³ vgl. Walden et al., INCOSE Systems Engineering Handbook v.5, Seite 27f

Wie im Verlauf der Arbeit weiter herausgearbeitet wird, liefert hierbei die Systemarchitekturprozess mit der Festlegung der Grundprinzipien, funktionalen und physikalischen Strukturen, sowie Schlüsseleigenschaften des Zielsystems⁴ bereits einen wesentlichen Beitrag. Gerade weil die Entscheidungen, welche in der Konzeptphase getroffen werden, besonders kritisch hinsichtlich dem Einfluss auf die gesamten Produktlebenszykluskosten sind, sollte Wert auf eine sorgfältige und konsistente Dokumentation der (wenn auch vorläufigen) Architekturentscheidungen gelegt werden. Die Defense Acquisition University (DAU) hatte bereits 1993 festgestellt, dass mit Ende der Konzeptphase bereits 70% der auflaufenden Produktlebenszykluskosten festgelegt sind!⁵

Während der **Entwicklungsphase** werden die zuvor festgelegten Konzepte zu einer Entwicklungsreife gebracht, die die Herstellung, Nutzung und Unterstützung des Systems über die gewünschte Nutzungsdauer hinweg erlaubt. Hierbei ist klar zu unterscheiden zwischen der Systemarchitektur, welche wesentliche Konzepte, Eigenschaften, Strukturen, Verhaltensweisen und Funktionen, für die Systemlösung festlegt und dem Systemdesign, welches eine detaillierte Beschreibung der vorgeschlagenen Systemlösung bereitstellt, die ihre Implementierung ermöglicht. Wichtig hierbei ist, dass das Design nicht die Architekturbeschreibung als verfeinerte Variante ersetzt, sondern beide Sichten, abstrakt einerseits und detailliert andererseits, gleichberechtigt nebeneinander existieren und ihre Bestimmung haben.⁶

In der **Produktionsphase** wird das System aus einer Prototypenreife zu einer Serienproduktionsreife gebracht und letztendlich produziert. Dabei unterstützt die Architekturbeschreibung beim Systemzusammenbau, der Erstellung der Produktdatenstämme sowie weitergehender Dokumentation. Oft erfolgt hierbei ein Feedback an den Entwicklungsprozess und letztendlich eine Bestätigung der Produzierbarkeit und Freigabe des Systems.⁶

⁴ vgl. Walden et al., INCOSE Systems Engineering Handbook v.5, Seite 118

⁵ vgl. Walden et al., INCOSE Systems Engineering Handbook v.5, Seite 7

⁶ vgl. Walden et al., INCOSE Systems Engineering Handbook v.5, Seite 28

Während der folgenden Lebenszyklusphasen bezüglich **Nutzung** und **Instandhaltung** wird das System beim Endnutzer in die Anwendung gebracht, im Feld beobachtet, um Mängel und Fehlverhalten zu erkennen, und notwendige Anpassungen durchzuführen. Änderungen des Systems bedingen dabei das Durchlaufen der vorhergehenden Lebenszyklusphasen hinsichtlich Konzept, Entwicklung und Produktion. Um dies leisten zu können ist es unabdingbar, dass technische Dokumentationen wie u.a. die Systemarchitekturbeschreibung über den Produktlebenszyklus auf Stand gehalten werden.⁷

Mit der **Stilllegungsphase** wird das System außer Betrieb genommen. Diese wird bereits während der Konzept- und Entwicklungsphase des Systems berücksichtigt. Bestehende Dokumentation wird für zukünftige Entwicklungen archiviert.⁷

⁷ vgl. Walden et al., INCOSE Systems Engineering Handbook v.5, Seite 29

(SysML) Modellierung und Systemarchitektur

Bevor auf den Zweck der Systemarchitektur eingegangen werden kann, muss eine klare Unterscheidung zwischen Modellierungsaktivitäten, die mittels SysML durchgeführt werden, und der Systemarchitektur gemäß ISO15288:2023 getroffen werden.

Wenn man sich heutige SysML Architekturmodelle anschaut, stellt man fest, dass diese für unterschiedlichste Zwecke erstellt werden. Dis sind u.a.:

- Erfassen operationeller Abläufe
- Bestimmung der Systemgrenze
- Erfassung externer Schnittstellen
- Beschreiben von Systemfunktionen und -verhalten
- Betriebszustände des Systems und deren Zustandsübergänge
- Interne Struktur des Systems
- Funktionsumfang und funktionales Verhalten von Systemelementen
- Zusammenspiel der einzelnen Systemelemente (statisch, dynamisch) und Bestimmung interner Schnittstellen
- Trade-Off Analysen von Architekturalternativen, soweit nicht separat im Systemanalyseprozess gehandhabt.

Aus der Auflistung ergibt sich, dass Modellierung mit SysML sich nicht auf die Systemarchitektur beschränkt, sondern einzelne Modellinhalte verschiedenen technischen Prozessen zuzuordnen sind. Natürlich kann eine Architekturbeschreibung auch in natürlich sprachlicher Beschreibung und Dokumentenbasiert erfolgen, was die prinzipielle Aussage jedoch nicht ändert.

Diese „Unschärfe“ der Systemmodellierung, d.h. sie ist nicht alleinig dem System Architecture Definition Prozess zuordenbar, spiegelt sich auch im System Architecture Framework (SAF)⁸ der GfSE wieder. Dieses Framework erfasst verschiedene Domains (vertikal: Operational bis Physical) und unterscheidet die abgebildeten Inhalte hinsichtlich des Aspekts (horizontal: Story & Context bis Mapping und Cross-Reference), wie im folgenden Diagramm abgebildet.

⁸ vgl. Ackva, Sascha, Leute, Michael et al., System Architecture Framework (SAF), <https://github.com/GfSE/SAF-Specification>, 06.01.2024 13:30 (Stand des letzten Zugriffs)

Studienarbeit: Die Rolle der Systemarchitektur in der Entwicklung

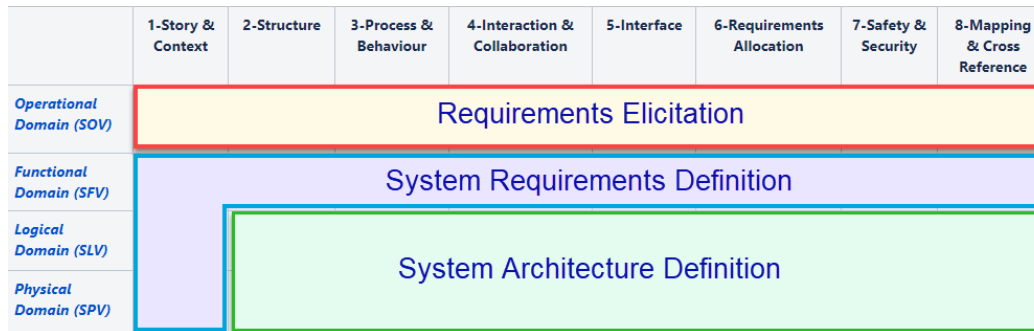


Abbildung 2: System Architecture Framework Matrix⁹

Die **Operational Domain** ist der Business or Mission Analysis und der Stakeholder Needs and Requirements Definition zuzuordnen. Der Fokus liegt hierbei auf der Identifikation von Stakeholdern, der Fähigkeitsanalyse und Beschreibung operationeller Handlungsabläufe. Diese Betrachtung ist essentiell, zur frühzeitigen Identifikation, Klärung und Konsolidierung der Stakeholder Erwartungen. Die damit erreichte Abgrenzung des Projektumfangs und Identifikation von „Features“ ermöglicht einen Einstieg in die Freigabeplanung und wann welches Release welche Fähigkeiten ermöglichen wird.

Die **Functional Domain** befasst sich mit den Systemfunktionen, dem Black-Box Verhalten des Systems, der Identifikation externer Schnittstellen und den externen Interaktionen, welche notwendig sind, die operationellen Fähigkeiten zu realisieren. Funktionalität und Interaktionen spiegeln Gesamtsystemeigenschaften wieder, die entweder mit Systemanforderungen korreliert werden oder ebendiese darstellen, je nach Prozessvorgabe. Damit ist die Functional Domain der System Requirements Definition zuzuordnen.

Die **Logical Domain** aggregiert identifizierte Systemfunktionen in logische Komponenten und definiert deren Verhalten und Kollaboration. Logische Komponenten erlauben eine weitgehend implementierungsfreie Betrachtung von Funktionsgruppen und den dazu nötigen Leistungsparametern für eine nachfolgende physikalische Auslegung des Systems.

⁹ vgl. Ackva, Sascha, Leute, Michael et al., System Architecture Framework (SAF), <https://github.com/GfSE/SAF-Specification>, 06.01.2024 13:30 (Stand des letzten Zugriffs)

Die **Physical Domain** bildet den Implementierungslevel mit konkreten physikalischen Elementen und deren tatsächlich verfügbaren Leistungsdaten ab. Die logischen Komponenten werden abschließend auf eine physikalische Architektur abgebildet, was einen Abgleich zwischen geforderter und gegebener Leistungsfähigkeit ermöglicht.

Logical Domain und Physical Domain sind größtenteils der System Architecture Definition zuzuordnen, jedoch sind die Kontextbetrachtungen (Abgrenzung nach außen, siehe ‚1-Story & Context‘) wie oben abgebildet der Systemanforderungsdefinition zuzuordnen.

Die in der Matrix aus Domains und Aspekten abgebildeten Sichten sind sehr eng miteinander verbunden und nicht immer trennbar. Als Beispiel wird hierfür die Systemgrenze benannt, welche einerseits eine notwendige Definition im Kontext des Entwicklungsumfangs ist und andererseits den Startpunkt für eine Systemdekomposition in der Systemarchitektur darstellt. Daher wird im Folgenden die ‚Systemarchitekturbeschreibung‘ als Begriff im Zusammenhang mit anderen Entwicklungsprozessen verwendet, welche die operationelle, funktionale, logische und physikalische Domain umfasst.

Im Bereich des Anforderungsmanagements und der Traceabilityanalyse spielt die Systemarchitekturbeschreibung eine wesentliche Rolle, sei es im Nachweis der vollständigen Berücksichtigung der Systemanforderungen oder auch der Zuordnung der verfeinerten Anforderungen auf die Systemelemente. Dies erfolgt heute oftmals als Verlinkung innerhalb des Anforderungsmanagement Tools (d.h. die Systemarchitekturbeschreibung steht daneben und gibt lediglich die Struktur vor), wird aber auch zunehmend über Verlinkung auf strukturelle und funktionale Elemente des SysML-Architekturmodells selbst realisiert.

Es wird davon ausgegangen, dass die detaillierte Auslegung von Schnittstellen und Algorithmen, Integrations- und Verifikationsszenarien, wie auch die detaillierte Systemanalyse von Alternativen, Machbarkeitsstudien und Leistungsabschätzungen sich zwar oft mit SysML-Werkzeugen abbilden lässt, zumeist aber spezielle Werkzeuge genutzt werden und eine Systemarchitekturbeschreibung deren Inhalte nur referenziert.

Systemarchitektur und die technischen Prozesse

Systems Engineering ist ein inkrementeller, iterativer und rekursiver Ansatz. Prozessgruppen und deren Prozesse laufen zumeist parallel über die Life-Cycle Stages hinweg in variierender Intensität und nicht rein sequenziell, wie unten dargestellt. Erkenntnisse in einem Prozess haben direkte Auswirkung auf andere, vermeintlich vorgelagerte Prozesse, wodurch die dort erarbeiteten Inhalte ergänzt oder überarbeitet werden müssen. Die Prozessaktivitäten existieren somit nicht nur parallel, sie interagieren und bedingen sich gegenseitig.

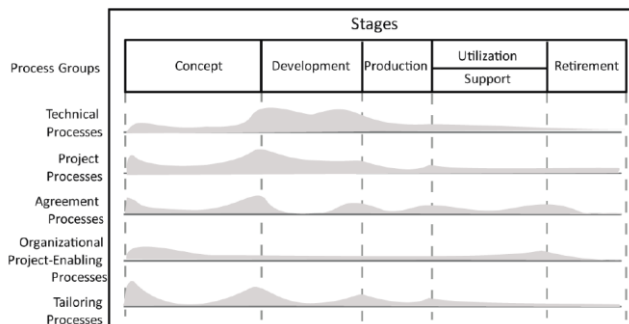


Abbildung 3: SE level effort across life-cycle stages¹⁰

Während auf die Beziehung der Systemarchitekturbeschreibung zu Business or Mission Analysis, der Stakeholder Needs and Requirements Definition und der System Requirements Definition eingegangen wurde, sollen hier nun weitere technische Prozesse betrachtet werden:

Die **System Design Definition** erweitert die Systemarchitekturbeschreibung dahingehend, dass der notwendige Detaillierungsgrad für eine nachfolgende Implementierung bereitgestellt wird. Auf Systemebene können diese Auslegungen von Algorithmen sein sowie detaillierte Schnittstellenbeschreibungen. Es ist jedoch nicht ungewöhnlich, dass die Systemarchitekturbeschreibung schrittweise erweitert wird. So identifiziert diese zu Beginn ggf. nur signifikante Systemelemente und Funktionen, während weitere (z.B. Monitorfunktionen) erst mit weiterführender Systemauslegung evident werden. Aus der Praxis heraus ist davon abzuraten, die Systemarchitekturbeschreibung übermäßig mit detaillierten Designinformationen anzureichern.

¹⁰ Haskins, Systems Engineering Handbook v.3.2.2, San Diego, 2011, Seite 24

Während man eine umfassende „Single-Source-of-Truth“ erhält, wird eine solche Beschreibung zunehmend umfangreich und damit schwerer verständlich. Folglich werden nur wenige Spezialisten sich mit der (zu detaillierten) Architekturbeschreibung befassen, während das Gros des Entwicklungsteams gemäß dem eigenen Verständnis der Lösungsarchitektur agiert. Dies führt in vielen Fällen zu Divergenz und folglich einem erhöhten Risiko in der Entwicklung. Damit verliert die Systemarchitekturbeschreibung ihre mögliche Koordinierungs- Harmonisierungs- und Steuerungsfunktion.

Für die **Integration** der Systemelemente sowie **Verification** und **Validierung** des Gesamtsystems spielt die Systemarchitekturbeschreibung eine wichtige, wenn auch oft wenig beachtete Rolle. Die Beschreibung der erwarteten Fähigkeiten, der bereitgestellten Funktionalität, der Betriebszustände und der dynamischen Interaktion über die internen und externen Schnittstellen stellt den Kontext und Zusammenhang zwischen meist sehr fokussierten (System- und Systemelement-) Anforderungen her. Hierbei können z.B. Ablaufdiagramme das Verständnis des erwarteten Verhaltens signifikant erhöhen und als Vorlage für die Testfalldefinition dienen.

Im Rahmen der Validierung der angestrebten Lösung mittels Failure Mode Effect Analysis (FMEA), Fault Tree Analysis (FTA), Dependant Failure Analysis (DFA) und anderen Verfahren, spielt die Systemarchitekturbeschreibung eine wesentliche Rolle als „Prüfgegenstand“, zusammen mit dem abgeleiteten detaillierten Design.

Während der Nutzungs- und Unterstützungsphase des Produktlebenszyklus, mit den **Transition**, **Operation**, **Maintenance** und **Disposal** Prozessen, spielt die Systemarchitekturbeschreibung eine wichtige Rolle in der Analyse und Bewertung von beobachteten Fehlerverhalten oder dem Änderungsbedarf bei neuen oder geänderten Anforderungen an das System. Ohne eine Systemarchitekturbeschreibung und der Möglichkeit Abhängigkeiten und Zusammenhänge zu erkennen, ist es schier unmöglich, komplexe Fehlerursachen zu erkennen oder Systemanpassungen vorzunehmen, ohne ungewollte Emergenzen zu riskieren. Damit nimmt die Architekturbeschreibung zwar keinen direkten Einfluss, muss aber in dieser Lebenszyklusphase vorhanden sein.

Studienarbeit: Die Rolle der Systemarchitektur in der Entwicklung

Abschließend lässt sich sagen, dass die Systemarchitekturbeschreibung ein zentrales Element innerhalb der technischen Prozesse darstellt. Dies zeigt sich deutlich auch im N2 Diagramm des INCOSE SE Handbuchs. Der System Architecture Definition Prozess erhält Input von jedem anderen technischen Prozess, während es als Output in die meisten Prozesse in irgendeiner Form eingeht.

BMA	X	X	X	X	X	X	X	X	X	X	X	X	X
X	SNRD	X	X	X	X	X	X	X	X	X	X	X	X
		SRD	X	X	X	X	X	X	X	X	X	X	X
		X	SAD	X	X	X	X	X	X	X	X	X	X
		X	X	DD		X	X	X	X	X	X	X	X
X	X	X	X	X	SA								
X	X	X	X	X	X	IMPL	X		X				
X	X	X	X	X	X	X	INT	X	X	X			
X	X	X	X	X	X			VER	X	X			
X	X	X	X	X	X				TRAN	X	X	X	X
X	X	X	X	X	X				X	VAL			
X	X	X	X	X	X						OPER	X	X
X	X	X	X	X	X						X	MAINT	X
X	X	X	X	X	X								DISP

Abbildung 4: N2-Diagramm der technischen Prozesse gemäß INCOSE SE Handbuch v5¹¹

¹¹ vgl. Walden et al., INCOSE Systems Engineering Handbook v.5, Appendix D, Seite 319

Systemarchitektur und die technischen Management Prozesse

Die Projektplanung und ihre Strukturpläne

Die Darstellung von Produktstrukturen, logisch oder physikalisch, in der Systemarchitekturbeschreibung sind nicht die einzigen erfassten „Strukturen“ im Entwicklungsprozess. Im Rahmen der Projektplanungsaktivitäten gibt es eine ganze Reihe an „Strukturen“, die logisch wie auch inhaltlich aufeinander aufbauen bzw. sich beeinflussen. Nachfolgend eine nicht abschließende Auflistung:

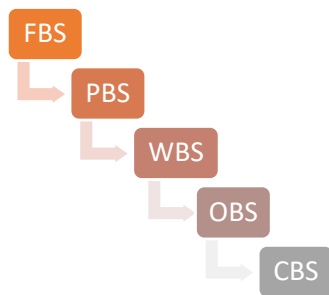


Abbildung 5: Übersicht verschiedener Strukturaufbrüche im Projekt

- FBS – Functional Breakdown Structure
- PBS – Product Breakdown Structure
- WBS – Work Breakdown Structure
- OBS – Operational Breakdown Structure
- CBS – Cost Breakdown Structure

Die **Functional Breakdown Structure (FBS)** ist ein hierarchischer, strukturierter modularer Aufbruch von Systemfunktionen, die notwendig sind die geforderten operationellen Fähigkeiten zur Verfügung zu stellen.¹² Mit der Allokation von (Teil-)Funktionen auf eine physikalische Architektur ermöglicht sie, unterschiedliche physikalische Architekturvorschläge hinsichtlich ihrer Effizienz und Effektivität der Funktionsrealisierung zu analysieren. Diese implementierungsfreie Analyse notwendiger Funktionalität ist sehr eng verknüpft, wenn

¹² vgl. DeHoff, Bryan et al., The Functional Breakdown Structure (FBS) and Its Relationship to Life Cycle Cost, Denver, 2009, <https://ntrs.nasa.gov/citations/20130012526>, 06.01.2024 13:30 (Stand des letzten Zugriffs)

Studienarbeit: Die Rolle der Systemarchitektur in der Entwicklung

nicht identisch, mit der Functional Domain und Logical Domain der zuvor vorgestellten Systemarchitekturbeschreibung gemäß SAF, wenn auch die Betrachtung der Kosteneffizienz über den Produktlebenszyklus typischerweise nicht Teil der Architekturbeschreibung selbst ist.

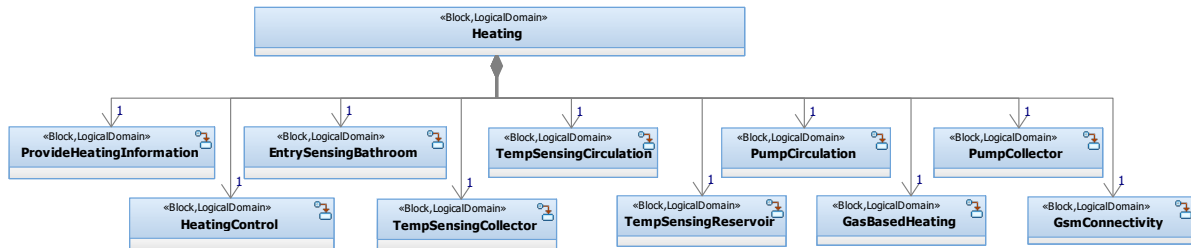


Abbildung 6: Beispiel Functional Breakdown Structure (FBS)

Die **Product Breakdown Structure (PBS)** beschreibt die physikalischen Komponenten eines Produkts oder Systems. Typischerweise wird die hierarchisch abgebildet. Die PBS erlaubt dem Projektleiter und dem Team ein klares Verständnis über die benötigten physikalischen Elemente zu bekommen.¹³ Es ist naheliegend, dass die PBS direkt mit der Physical Domain der Systemarchitekturbeschreibung korreliert, wenn sich auch die betrachtete Detailtiefe unterscheiden kann.

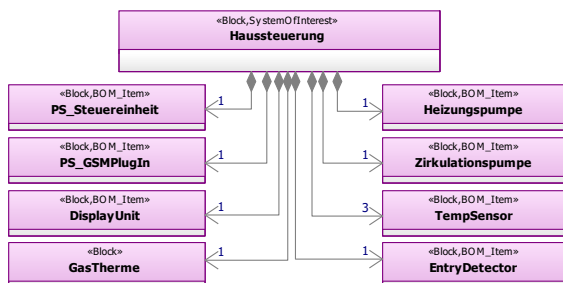


Abbildung 7: Beispiel Product Breakdown Structure (PBS)

Interessant in diesem Zusammenhang ist das NASA Systems Engineering Handbook. Dieses betrachtet die PBS nicht nur im Zusammenhang mit der Anforderungszuweisung auf Systemelemente sondern auch als zentrales Framework im Bereich des Technischen Assessments¹⁴.

¹³ vgl. Malsam, William, What Is a Product Breakdown Structure (PBS)?, 02.06.2022, <https://www.projectmanager.com/blog/product-breakdown-structure>, 06.01.2024 13:30 (Stand des letzten Zugriffs)

¹⁴ vgl. Hirshorn et al., NASA Systems Engineering Handbook Rev.2, 2017, Seite 249

Die **Work Breakdown Structure (WBS)** ist eine hierarchische Dekomposition eines Projektes in klar umrissene und handhabbare Arbeitspakete.¹⁵ Sie entspricht dem Projektstrukturplan (PSP) im deutschsprachigen Raum. Die Arbeitspakete orientieren sich in vielen Bereichen an der PBS, da sich für das System und jedes einzelne Systemelement nötige Arbeitspakete ableiten lassen bzw. Absicherungsmaßnahmen bezüglich technischer Risiken. Arbeitspakete für ein Systemelement können z.B. sein:

- Durchführen einer Funktionalen Analyse
- Erarbeiten der Systemspezifikation
- Erarbeiten der Systemarchitektur (logisch/physikalisch)
- Validierung der Systemarchitektur und Design mittels FTA/FMEA
- Spezifikation der identifizierten Schnittstelle X (...)
- Durchführen einer Machbarkeitsstudie bezüglich Y (...)
- Verifikation zum Nachweis der korrekten Implementierung

Diese Arbeitspakete werden hinsichtlich Umfang, notwendiger Ressourcen zur Durchführung und Abhängigkeiten (zeitlich, wie auch untereinander) betrachtet. Die WBS ist ein essentielles Tool mithilfe dessen Projektmanager und Teammitglieder den Projektumfang und Projektfortschritt verstehen und kontrollieren können.

Die **Operation Breakdown Structure (OBS)** zeigt die hierarchische Beziehungen zwischen im Projekt beteiligten Organisationseinheiten auf und wird genutzt, um Arbeitspakete aus der WBS auf Ressourcen (z.B. eine Entwicklungsabteilung) in einem Projekt zuzuweisen.¹⁶ Die OBS ist in vielen Fällen in der WBS integriert oder wird im Projektplan grafisch abgebildet.

Die **Cost Breakdown Structure (CBS)** ist eine hierarchische Darstellung von Kostenpunkten im Projekt. Sie lehnt sich stark an der WBS an. Die CBS stellt zu Beginn eine Auflistung der Soll-Kosten dar, wobei über die Zeit abgeschlos-

¹⁵ vgl. o.V., Work Breakdown Structure (WBS), <https://www.projectmanager.com/guides/work-breakdown-structure>, 06.01.2024 13:30 (Stand des letzten Zugriffs)

¹⁶ vgl. o.V., Organization Breakdown Structure (OBS), 02.12.2022, [https://cio-wiki.org/wiki/Organization_Breakdown_Structure_\(OBS\)](https://cio-wiki.org/wiki/Organization_Breakdown_Structure_(OBS)), 06.01.2024 13:30 (Stand des letzten Zugriffs)

sene Aktivitäten mit den tatsächlichen Ist-Kosten ergänzt werden. Somit lassen sich typische kalkulatorische Betrachtungen wie die Cost-to-Completion und Cost-at-Completion ableiten. In vielen Fällen wird die CBS in die WBS integriert oder wird in separaten Excel-Listen geführt.

Betrachtet man die verschiedenen Strukturbäume, wie sie im Rahmen der Projektplanung erarbeitet und gepflegt werden, so erkennt man gerade im Bezug auf die WBS, FBS und PBS, eine enge Beziehung zur Systemarchitekturbeschreibung, vor Allem der Functional, Logical und Physical Domain. Hieraus ergibt sich der Mehrwert, welche eine – wenn auch zu Beginn abstrakte – Systemarchitekturbeschreibung in der Projektplanung bringen kann. Das INCOSE SE Handbook v5 benennt hierzu die Prozessaktivität für den Projektplanungsprozess: „Develop or select Breakdown Structures based on the evolving system architecture“¹⁷ und hebt die gemeinsame Verantwortung von SE und PM für die „Breakdown Structures“ gerade zu Beginn des Projektes hervor. Darüber hinaus wird ein erhebliches Projektrisiko gesehen, wenn die verschiedenen Strukturen unkoordiniert gehandhabt werden: “If the different structures are managed separately by the respective teams without coordination, problems may arise in the project.“¹⁸

Betrachtung weiterer technischer Management Prozesse

Ohne im Einzelnen auf jeden Prozess innerhalb der technischen Management Prozesse einzugehen, lassen sich aus der vorangegangenen Betrachtung des Zusammenspiels zwischen Systemarchitekturbeschreibung und FBS, WBS und PBS weitere Analogschlüsse zur unterstützenden Bedeutung der Systemarchitektur schließen.

Durch die hierarchische Identifikation der Systemelemente, der allokierten Funktionen und deren geforderten Qualitätsanforderungen sowie der verschiedenen internen und externen Schnittstellen, lassen sich analog zu den Arbeitspaketen der WBS auch notwendige Arbeitsprodukte als Configuration Items im Bereich des Konfigurationsmanagements ableiten. Für das Gesamtsystem

¹⁷ Walden et al., INCOSE Systems Engineering Handbook v5, San Diego, 2023, Seite 72

¹⁸ Walden et al., INCOSE Systems Engineering Handbook v5, San Diego, 2023, Seite 269

Studienarbeit: Die Rolle der Systemarchitektur in der Entwicklung

und seine Systemelemente gemäß PBS lassen sich Konfigurationen definieren, d.h. welche Information Items für einen bestimmten Zweck das Element vollständig beschreiben. Damit beeinflusst die Architektur auch die Dokumentenablagestruktur wesentlich mit.

Eine strukturierte und systematische Analyse der identifizierten Funktionen und Schnittstellen im Sinne der technischen Risikoanalyse kann frühzeitig auf Unwägbarkeiten hindeuten, die wiederum mit vorbeugenden Maßnahmen (wie Machbarkeits- und Performanceabschätzungen oder gezielter prototypischer Entwicklung) in der Projektplanung berücksichtigt werden kann.

Die Systemarchitekturbeschreibung als ist ein wesentlicher Input in die Projektplanung, Teil der technischen Managementprozesse. So kann dies auch aus dem N2 Diagramm des INCOSE SE Handbuchs, wie unten verkürzt dargestellt, herausgelesen werden. Darüber hinaus wurde hier herausgearbeitet, dass die hierarchische Aufgliederung des Systems in seine Elemente (physikalisch oder funktional) weitere Prozesse unterstützt.

PP	X							
X	PAC			X	X	X		
X	X	DM						X
X	X		RM					
X	X			CM	X			
X	X			X	INFOM			
X	X					MEAS		
X	X						QA	
X	X	X						SA

Abbildung 8 : N2-Diagramm der Systemarchitekturentwicklung und der technischen Managementprozesse gemäß INCOSE SE Handbuch v5¹⁹

¹⁹ vgl. Walden et al., INCOSE Systems Engineering Handbook v.5, Appendix D, Seite 319

Teamwork Aspects

In der Systemarchitektur wird eine ganze Reihe an zentralen Prinzipien für spätere Produktlösungen entwickelt, dargestellt und analysiert. Die getroffenen Entscheidungen haben oft vielschichtige Aspekte und Konsequenzen, die sich in unterschiedlichen Arbeitsprodukten der verschiedenen technischen Entwicklungsprozesse abbilden. Diese Vielschichtigkeit ist durchaus als komplex zu betrachten, da sie viele fachliche Disziplinen betrifft, die in ihren Abhängigkeiten oft schwer zu durchschauen sind. So kann es sein, dass die Allokation einer Funktion von einem Prozessor auf einen anderen neben der Änderung der Architekturbeschreibung selbst Änderungen in Spezifikationen, Schnittstellenbeschreibungen und Integrationsplänen führt. Die verschobene Prozessorlastverteilung könnte weiterhin zu einer erhöhten Lastaufnahme führen, was ein thermisches Re-Design von Elektronik und Gehäusebauteilen bedingt und durch die strukturelle Änderung eine Neubewertung des Sicherheitskonzepts notwendig macht (usw.).

Diese komplexe Vielschichtigkeit ist ein klarer Hinweis darauf, dass die Systemarchitekturbeschreibung keinen Selbstzweck erfüllt und nicht die Arbeit „eines Architekten“ sein kann, sondern die gemeinsame Analyse von Lösungsalternativen, Abstimmung der gewählten Lösung und Verständigung eines Cross-disziplinären Entwicklungsteams ist.

Das INCOSE Handbuch v5 sagt hierzu: “This process is iterative and requires participation of architects, SE practitioners, and specialists in relevant domains, subject matter experts and other stakeholders.”²⁰

Die Systemarchitekturbeschreibung sollte im Projekt als Single-Source-of-Truth angesehen werden. Dazu gehört, dass die Architekturbeschreibung über die Zeit inhaltlich aufwächst, d.h. jederzeit relevante Informationen bereitstellt. Dies sind zu Projektbeginn abstrakte Darstellungen und später eher konkrete Detailbeschreibungen. Daraus ergibt sich, dass man weder die Architektur erst „fertigstellen“ kann, noch dass man eine abstrakte Architektur erstellt nicht weiter pflegt, um kontinuierlich von Wert zu sein.

²⁰ Walden et al., Systems Engineering Handbook v. 5, San Diego, 2011, Seite 118

Zusammenfassung

Eingangs wurde die These aufgestellt, dass die Systemarchitekturbeschreibung heute oftmals

- als separates Arbeitsprodukt gehandhabt wird
- nur wenige die Architektur erarbeiten oder nutzen
- nur für Quality Assessments als relevant angesehen wird

Im Laufe der Studienarbeit wurde herausgearbeitet, dass die Systemarchitekturbeschreibung ein wesentlicher Informationsbaustein über den gesamten Produktlebenszyklus darstellt.

So beschreibt sie die Abgrenzung des Systems nach außen, sowie die innere Struktur des Systems. In einer abstrakten Form beeinflusst dies wesentlich die Projektplanung, Entwicklungsaktivitäten, Arbeitsprodukte und Datenabgestruktur. Architekturfestlegung und Projektplanung müssen eng abgestimmt miteinander laufen, ohne das Entwicklungsrisiko unnötig zu erhöhen.

Die Systemarchitektur legt wichtige Prinzipien in der Lösungsbeschreibung fest. Sie ist damit ein Kernelement in den technischen Prozessen und ist mit nahezu jedem andern technischen Prozess eng verbunden. Dies und auch die Komplexität bei der Bewertung von Auslegungsalternativen machen die Systemarchitektur zu einem Thema, dass nur in einem Cross-disziplinarem Team gehandhabt werden kann und muss.

Die Systemarchitektur beschreibt die angestrebte Lösung, d.h. das Produkt. Das Versäumnis diese zu erfassen, führt zur Divergenz der Entwicklungsaktivitäten und erhöhtem Projektrisiko.

Fazit: Die Systemarchitektur ist das zentrale Dokument der Lösungsbeschreibung und muss damit auch mehr im Zentrum der Entwicklungsaktivitäten stehen als dies heute oftmals der Fall ist.

Literaturverzeichnis

Bücher und Beiträge aus Sammelwerken

- 1) Haskins, Cecilia, INCOSE Systems Engineering Handbook v3.2.2, San Diego, 2011
- 2) Walden, David D. et al., INCOSE Systems Engineering Handbook v4, San Diego, 2015
- 3) Walden David D. et al., INCOSE Systems Engineering Handbook, v4, deutsche Übersetzung, San Diego, 2015
- 4) Walden David D. et al., INCOSE Systems Engineering Handbook v5, San Diego, 2023

Internetquellen

- 5) Ackva Sascha, Leute Michael et al., System Architecture Framework (SAF), <https://github.com/GfSE/SAF-Specification>, 06.01.2024 13:30 (Stand des letzten Zugriffs)
- 6) Alby, Tom, Projekt Management Wissen, <https://project-management-knowledge.com/de/definitionen/o/organisational-breakdown-struktur-obs/>, 06.01.2024 13:30 (Stand des letzten Zugriffs)
- 7) DeHoff, Bryan et al., The Functional Breakdown Structure (FBS) and Its Relationship to Life Cycle Cost, Denver, 2009, <https://ntrs.nasa.gov/citations/20130012526>, 06.01.2024 13:30 (Stand des letzten Zugriffs)
- 8) Hirshorn, Steven R. et al., NASA Systems Engineering Handbook Rev.2, 2017, <https://ntrs.nasa.gov/citations/20170001761>, 06.01.2024 13:30 (Stand des letzten Zugriffs)
- 9) Malsam, William, What Is a Cost Breakdown Structure (CBS) In Project Management?, 07.06.2023, <https://www.projectmanager.com/blog/cost-breakdown-structure>, 06.01.2024 13:30 (Stand des letzten Zugriffs)
- 10) Malsam, William, What Is a Product Breakdown Structure (PBS)?, 02.06.2022, <https://www.projectmanager.com/blog/product-breakdown-structure>, 06.01.2024 13:30 (Stand des letzten Zugriffs)
- 11) o.V., Organization Breakdown Structure (OBS), 02.12.2022, [https://cio-wiki.org/wiki/Organization_Breakdown_Structure_\(OBS\)](https://cio-wiki.org/wiki/Organization_Breakdown_Structure_(OBS)), 06.01.2024 13:30 (Stand des letzten Zugriffs)
- 12) o.V., Work Breakdown Structure (WBS), <https://www.projectmanager.com/guides/work-breakdown-structure>, 06.01.2024 13:30 (Stand des letzten Zugriffs)