

Lebenszyklusphasenmodelle Heute

Dieter Scheithauer

CASSIDIAN, Rechliner Straße, 85077 Manching, dieter.scheithauer@cassidian.com

© Dieter Scheithauer, 2013.

Zusammenfassung: Bis in die achtziger Jahre des letzten Jahrhunderts standen Lebenszyklusphasenmodelle im Zentrum des Systems Engineering. Die Einteilung des Systemlebenszyklus in Phasen war ein wesentliches Gestaltungsmerkmal des Systems-Engineering-Prozesses. Heute werden Lebenszyklusphasenmodelle in der Systems-Engineering-Literatur zwar noch erwähnt und beschrieben, aber die intensiven Diskussionen über Sinn und Zweck, Vorteile und Nachteile einzelner Lebenszyklusphasenmodelle sind Vergangenheit. Sind Lebenszyklusphasenmodelle somit ausdiskutiert? Sind sie in Zeiten, in denen im Systems Engineering durchgängige Modellbildung, Wertschöpfungsketten und agile Vorgehensweisen im Zentrum der Debatten stehen, sogar obsolet geworden? Ganz im Gegenteil scheint eine Wiederaufnahme der Diskussion aus verschiedenen Gründen sinnvoll zu sein. Dieser Beitrag schlägt ein generalisiertes Lebenszyklusphasenmodell vor, das für marktgetriebene Systementwicklungen im organisatorischen Umfeld einer auf Spezialisierung und Arbeitsteilung beruhenden Industriekultur geeignet ist. Hier kommt es gerade auf die Verzahnung von sequentiellen Sichten auf den Systemlebenszyklus mit iterativen Entwicklungsansätzen an.

1 Einleitung

Systeme durchlaufen einen Lebenszyklus. Systems Engineering berücksichtigt diesen Lebenszyklus, um Kunden eine kosteneffiziente Lösung anzubieten und alle Stakeholder, die im Laufe des Lebenszyklus mit dem System befasst sind, zufrieden zu stellen. Zur groben Strukturierung des Lebenszyklus bietet sich eine Aufteilung in einzelne Phasen an. ISO/IEC 15288 definiert Lebenszyklusphasen als Zeitabschnitte des Gesamtlebenszyklus [II08-2]. Der Zuschnitt der einzelnen Lebenszyklusphasen, die in einzelnen Phasen durchzuführenden Aktivitäten und die Kriterien für den Übergang zwischen den Lebenszyklusphasen fungierten für lange Zeit als wesentliche Gestaltungsparameter für die Definition des Systems-Engineering-Prozesses.

Die Dominanz der Lebenszyklusphasenmodelle in Systems-Engineering-Debatten hat sich in den letzten zwei Jahrzehnten weitgehend verflüchtigt. Als Beispiel für den derzeitigen Umgang mit Lebenszyklusphasenmodellen lassen sich das INCOSE Systems Engineering Handbook [Ha11] und die ISO/IEC 15288 anführen: Es gibt sie, und gewisse Aktivitäten sollten in bestimmten Lebenszyklusphasen stattfinden. Eine Diskussion über die Vor- und Nachteile einzelner Lebenszyklusphasenmodelle findet nicht statt. Kriterien zur Auswahl eines spezifischen Lebenszyklusphasenmodells bleiben vage. Als beschreibende Handreichung zu ISO/IEC 12207 [II08-1] und ISO/IEC 15228 [II08-2] führt ISO/IEC TR 24748-1 [II10] noch an, dass ein Lebenszyklusphasenmodell am Geschäftsmodell einer Organisation ausgerichtet sein kann.

Für die abnehmende Relevanz, die Lebenszyklusphasenmodellen im Systems Engineering zugemessen wird, lassen sich in der historischen Entwicklung verschiedene Strömungen ausmachen.

Teils scheinen Lebenszyklusphasenmodelle ein Opfer ihres eigenen Erfolges geworden zu sein. Dennis Buede beschreibt in seinem Buch [Bu09], dass Arthur D. Hall, der 1962 mit dem Buch „A Methodology for Systems Engineering“ [Ha62] eines der ersten Bücher über Systems Engineering vorgelegt hat, mit einem Lebenszyklusphasenmodell mit fünf Phasen auskam. In den achtziger Jahren gab es dann Lebenszyklusphasenmodelle mit mehr als zehn Phasen, wie zum Beispiel für den Beschaffungsgang für Wehrmaterial im Geschäftsbereich des Bundesverteidigungsministeriums. Die Existenz von mehr Phasen in der Entwicklung erhöhte den bürokratischen Aufwand zur Administration. Ferner mussten kleinteiligere Phaseneinteilungen rein statistisch betrachtet dazu führen, dass es bei den Phasenübergängen häufiger Überhänge gab [Sc12]. Ein rein sequentielles Lebenszyklusphasenmodell stellt schließlich nur eine grobe Approximation des realen Entwicklungsgeschehens mit seinen Rückkopplungen und Iterationen dar.

Teils boten andere Ansätze, deren Ursprung häufig im Software Engineering lag, eine detailliertere Prozessmodellierung. Einen Überblick hierzu findet man bei Larman und Basili [LB03]. Ähnelte das 1970 von William Royce vorgeschlagene Wasserfallmodell noch stark einem sequentiellen Lebenszyklusphasenmodell, so arbeitet das Spiralmodell von Barry Boehm Mitte der achtziger Jahre eher den evolutionären Aspekt jeder Entwicklung heraus. Stilbildend im Systems Engineering war aber besonders das V-Modell [FM91, FMC05]. Diese Modelle waren den Lebenszyklusphasenmodellen darin überlegen, die Abhängigkeiten und Flüsse von Information besser darzustellen, als dies mit Lebenszyklusphasenmodellen möglich war. Versteht man diese Modelle als rein logische Sequenzen und löst sie vom kontinuierlich fortschreitenden Zeitstrahl, ergeben sich Interpretationen als Wertschöpfungsketten im Sinne eines Lean Systems Engineering, die für agile Managementverfahren offen sind [SF13, Sc12].

Im Systems Engineering scheint es bezüglich des Werts von Lebenszyklusphasenmodellen eine gewisse Verunsicherung zu geben. Nahrung erhält dieser Verdacht durch weitere Ausführungen in ISO/IEC TR 24748-1 [II10]. Dort wird gesagt, dass Lebenszyklusphasen nicht als diskrete Abfolge einzelner Phasen zu verstehen sind, sondern sich in der Praxis gegenseitig bedingen und überlappen. Ferner wird gewarnt, dass Lebenszyklusphasen nicht notwendigerweise zeitlich sequentiell aufeinander folgen. Wie diese Aussagen noch mit der oben zitierten Definition von Lebenszyklusphasen im gleichen Dokument zur Deckung gebracht werden können, bleibt rätselhaft.

Es ist richtig, dass Wertschöpfungskettendarstellungen besser geeignet sind, die Systems-Engineering-Aktivitäten abzubilden. Ein detailliertes, qualitätsorientiertes Systems Engineering Management kann darauf nicht verzichten. Es bleibt aber genauso festzuhalten: Es gibt einen Bedarf für eine zeitlich sequentielle Betrachtung von Systems-Engineering-Projekten. Dieser Bedarf liegt weniger bei den projektdurchführenden Systemingenieuren, sondern beim Management der ausführenden Organisation sowie auf Seiten von Kunden und gegebenenfalls von Zulassungsbehörden.

Es ist deshalb nicht sinnvoll, Lebenszyklusphasenmodelle als Wertschöpfungskettendarstellung oder Wertschöpfungsdarstellungen als Lebenszyklusphasenmodelle umzuinterpretieren. Beide Sichten haben unabhängig voneinander ihre Daseinsberechtigung: Wertschöpfungsketten sind projektintern von höherer Bedeutung, während Lebenszyklusphasenmodelle für die Sicht von außen auf ein Systems-Engineering-Projekt nützlich sind.

Nach diesen Klärungen geht es im Weiteren darum, ein generalisiertes Lebenszyklusphasenmodell zu definieren und zu begründen. Dabei sind die Ausführungen auf die Lebenszyklusphasen beschränkt, die die Systems-Engineering-Aktivitäten im V widerspiegeln. In einem ersten Schritt werden die Lebenszyklusphasenmodelle in der gängigen Literatur zum Systems Engineering analysiert. Im zweiten Schritt wird ein generalisiertes Lebenszyklusphasenmodell entwickelt und begründet. In einem abschließenden dritten Schritt werden die einzelnen Phasen beschrieben.

2 Stand der Lebenszyklusphasenmodellierung

Diesem Abschnitt liegt die Auswertung von heute gebräuchlichen Lehrbüchern zum Systems Engineering und verschiedenen Systems-Engineering-Standards zugrunde.

Zwei Bücher thematisieren Lebenszyklusphasenmodelle nur am Rande. Hybertsen argumentiert auf einer Abstraktionsebene, die für das weitere Vorgehen unergiebig ist [Hy09]. Haberfellner und Koautoren legen den Schwerpunkt auf den Projektlebenszyklus und weniger auf den Systemlebenszyklus [HNB02].

ISO/IEC 15288 beschränkt sich auf die Konzeptphase und Entwicklungsphase [II08-2]. Forsberg, Mooz und Cotterman definieren Ähnliches als „Study Period“ und „Implementation Period“ [FMC05]. Kossiakoff und Koautoren sprechen von „Concept Development“ und „Engineering Development“ [KSS11]. Blanchard spaltet die Entwicklungsphase auf in „Preliminary System Design“ und „Detail Design and Development“ [Bl04]. Im gleichen Schema bleibt er auch zusammen mit Fabrycky [BF06]. Driscoll und Kucik ergänzen als dritte Phase am Anfang „Establish System Need“ [DK11].

Die anderen Autoren und Standards arbeiten mit mehr als drei Phasen. In Ergänzung zum Modell von Blanchard weist Buede die Systemintegrationsphase extra aus [Bu09]. Maier und Rehtin benutzen ein detaillierteres Modell [MR09], das sich jedoch gut auf das Drei-Phasen-Modell von Driscoll und Kucik übertragen lässt.

Die Lebenszyklusphasenmodelle werden in den Büchern vor Allem dafür benutzt, einen Rahmen für die Strukturierung des Inhalts zu setzen. Der Fokus liegt auf der Systementwicklung, entspricht also den Bedürfnissen der projektdurchführenden Systemingenieure. Entsprechend werden die verwendeten Lebenszyklusphasenmodelle hinsichtlich einer Außenperspektive nur wenig diskutiert. Wie oben erwähnt, liegt die Bedeutung von Lebenszyklusphasenmodellen aber gerade in der Außensicht. Dennoch geben sie einen Hinweis hinsichtlich einer passenden Granularität für ein generalisiertes Lebenszyklusphasenmodell, das mit Wertschöpfungskettendarstellungen kompatibel ist.

Ferner schließt eine fehlende Darlegung, warum ein bestimmtes Lebenszyklusphasenmodell gewählt wurde, nicht aus, dass entsprechende Überlegungen angestellt wurden.

3 Generalisiertes Lebenszyklusphasenmodell

Aus dem bisherigen Gedankengang ergeben sich Kriterien für ein generalisiertes Lebenszyklusphasenmodell, die in zwei Richtungen weisen. Zum einen sollte das Lebenszyklusphasenmodell insbesondere die Sicht von außen auf eine Systementwicklung unterstützen. Zum anderen bedingt dies aber auch eine Kompatibilität mit den Wertschöpfungskettenmodellen der Innensicht. Dazu treten Kriterien hinsichtlich der Effizienz in der Durchführung von Systems-Engineering-Projekten, die bisher noch nicht angesprochen wurden.

Wenn hier von Außensicht die Rede ist, betrifft dies zunächst einmal den organisatorischen Rahmen in dem eine Systementwicklung abgewickelt wird. Es existieren also Managementfunktionen, die in die Systementwicklung selbst nicht eingebunden sind, aber Investitionsentscheidungen treffen und für die Organisation rechtlich bindende Verpflichtungen nach außen eingehen. Dieser Zustand dürfte universell gegeben sein, unabhängig davon, ob es sich um ein Privatunternehmen, eine öffentlich-rechtliche oder eine staatliche Einrichtung handelt.

Der Kontext, in dem eine Organisation eingebunden ist, variiert allerdings. Traditionell sind Lebenszyklusphasenmodelle im Rahmen von Auftragsentwicklungen angewendet worden. In den meisten Fällen bedeutet dies auch, dass entweder alle Kosten nach Aufwand plus Gewinn abgerechnet werden oder dies zumindest für einen Teil des Entwicklungsaufwandes gilt, nämlich für den Fall unvorhergesehener Risiken oder gewünschter Zusatzleistungen. Dies ist nicht nur typisch für Aufträge der öffentlichen Hand. Jeder Bauherr ist in einer ähnlichen Situation gegenüber Architekten und Baufirmen. Nun beansprucht Systems Engineering eine gewisse Universalität in der Anwendbarkeit. Ohne sich auf marktgetriebene Geschäftsmodelle einzustellen, wird dies nicht gelingen. Notwendig ist eine fruchtbare Zusammenarbeit von Marketing, Geschäftsplanung und Systems Engineering. Dieser Anspruch muss von einem generalisierten Lebenszyklusphasenmodell eingelöst werden.

Außerhalb des engeren Organisationsumfeldes sind in der Regel weitere Stakeholder zu berücksichtigen, wie zum Beispiel Kunden, Zulassungsbehörden oder Vertriebsorganisationen. In der Zusammenarbeit mit diesen Stakeholdern dient das Lebenszyklusphasenmodell als Grundlage für Absprachen und gegebenenfalls als Basis für die Erfüllung gesetzlicher Anforderungen.

Die Forderung nach Kompatibilität von Lebenszyklusphasen- mit Wertschöpfungskettenmodellen führt zu der Schlussfolgerung, dass beide nicht identische Ziele verfolgen, sondern sich ergänzen sollten. Kossiakoff und Koautoren drücken dies in ihrer Definition für Lebenszyklusphasen sehr treffend aus [KSS11]: „These stages mark the more basic transitions in the system life cycle, as well as the changes in the type and scope of effort involved in systems engineering.“ Der zweite Teil der Aussage liefert

hier die entscheidenden Hinweise. In verschiedenen Lebenszyklusphasen mögen zwar Tätigkeiten und Methoden gleich sein, doch die verfolgten Ziele können sehr unterschiedlich ausfallen. Es gilt also, in einem Lebenszyklusphasenmodell die phasenspezifischen Zielkriterien zu definieren. In einem weiteren Schritt stellt sich dann die Frage, welche Wertschöpfungskettendarstellungen oder welche Abschnitte des übergeordneten Systems Engineering Value Streams [SF13] in der jeweiligen Phase durchlaufen werden.

Effizienzkriterien bilden eine zusätzliche Säule, die unbedingt in die Überlegungen mit einbezogen werden müssen. Die Entwicklung komplexer Systeme geht naturgemäß mit vielen Unsicherheiten einher. Während der Entwicklung vertieft sich das Verständnis für die Anforderungen, notwendige und parasitäre Funktionen sowie die Sinnfälligkeit der Systemarchitektur. Ganz ohne eine Phasengliederung führt dies schnell dazu, dass die Entwicklung in eine Vielzahl von Rückkopplungsschleifen unterschiedlicher Reichweite zerfällt, die nach Genialität zur Beherrschung des sich entwickelnden Chaos verlangen, weil die Implikationen einzelner Rückkopplungen nicht mehr absehbar und kontrollierbar sind.

Nun könnte man natürlich der Meinung sein, man tue, was man kann, und besser ginge es eben nicht. Diese Position kommt eher einer Kapitulation gleich. In der Realität kann immer wieder beobachtet werden, dass die Systemdefinition im dafür vorgesehenen Zeitrahmen nicht zu den erwarteten belastbaren Ergebnissen führt. Organisationen gehen dann immer wieder Verpflichtungen mit zu hohen Risiken ein, die später zu Terminüberschreitungen und hohen Mehrkosten führen. Einerseits ist diese Kritik aus einer A-Posteriori-Sicht wohlfeil und für die mehr oder weniger blamiert zurückbleibenden Systemingenieure nur eingeschränkt hilfreich. Andererseits ist aber auch zu konstatieren, dass im Systems Engineering ähnliche Effizienzsteigerungen wie bei der Umsetzung von Lean-Ansätzen in der Produktion selten zu beobachten sind.

Mit Bezug auf Taiichi Ohno, dem man wohl berechtigterweise den größten Teil der Urheberschaft des Produktionssystems von Toyota zuerkennen muss, ist man im Systems Engineering gut beraten, sich mit den beiden Prinzipien Autonomation und Just-In-Time auseinanderzusetzen [Oh88]. Die Domäne des Autonomation-Prinzips liegt vorwiegend in der Wertschöpfungskettenabbildung und sollte deshalb für die Lebenszyklusphasensicht nicht überbewertet werden. Anders sieht es beim Just-In-Time-Prinzip aus. Das Hineindenken in komplexe Systeme in Verbindung mit der Suche nach innovativen Systemlösungen benötigt mehr Zeit, als normalerweise zwischen konkreter Nachfragebildung und Verfügbarkeitszeitpunkt gewünscht wird. Ein effizientes Lebenszyklusphasenmodell kann dazu beitragen, die Zeitspanne zwischen konkreter Nachfrage und Verfügbarkeitszeitpunkt der technischen Lösung im Sinne des Just-In-Time-Prinzips zu optimieren. Im Gegenzug sollten Vorbereitungsphasen für die Systemkonzeption und Technologieentwicklung langfristig angelegt sein. Wirtschaftlich bedeutend ist, dass die Arbeit an Systemkonzepten mit minimalen Kapitalinvestitionen verbunden ist und Gelder für die Technologieentwicklung effizient und zielführend eingesetzt werden.

Ein weiterer Effizienzgesichtspunkt betrifft die Reichweite von Rückkopplungsschleifen. Je weiter in der Wertschöpfungskette zurückgegangen werden muss, umso kostspieliger und langwieriger sind einzelne Iterationen (siehe [SF13] für eine Kategorisierung der

Rückkopplungsschleifen in der dynamischen Sicht auf das V). Dabei sollte von einer rigiden Forderung, weitreichendere Rückkopplungen zu verbieten, Abstand genommen werden. Es hilft schon viel, wenn bei entsprechenden Entscheidungen ein qualitativ und quantitativ abgesichertes Bewusstsein für die vollständigen Implikationen einer Iteration vorhanden ist. Soweit dabei frühere Lebenszyklusphasen, die als abgeschlossen gelten, tangiert sind, können über die Transitionskriterien zwischen den Phasen auch die einzubindenden Entscheidungsebenen in der Organisation festgelegt sein.

Aus der Abwägung der genannten Kriterien wird das Phasenmodell in Abbildung 1 vorgeschlagen. Es besteht aus drei Phasen, die den Lebenszyklus eines konkreten Systems abbilden, und einer dazu teils parallel ablaufenden Technologieentwicklung, die nicht an einem einzelnen Produkt ausgerichtet ist, sondern Produktfamilien und organisationsweite Produktportfolios im Fokus hat. Das Phasenmodell gilt für das Gesamtsystem, das eine Organisation als Produkt ausliefert oder als Dienstleistung verfügbar macht. Die einzelnen Phasen werden in den nachfolgenden Kapiteln in ihren Grundzügen skizziert.

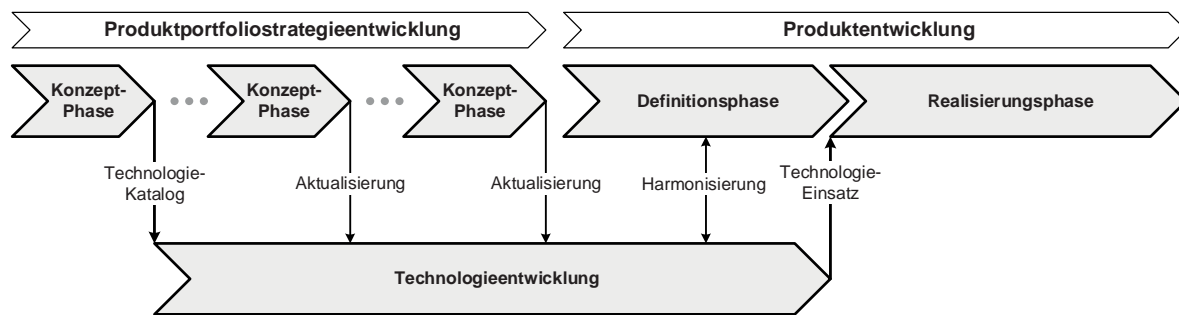


Bild1. Lebenszyklusphasenmodell für die Systementwicklung

4 Konzeptphase

Der Zweck der Konzeptphase liegt in der Unterstützung zur Entwicklung einer nachhaltigen Produktportfoliostrategie. Einzelne Element im Produktportfolio durchlaufen langfristig einen Reifeprozess, bis sich eine Gelegenheit zur Vermarktung bietet, die dann die Produktentwicklung auslöst. Die Konzeptphase richtet den Fokus auf das Zusammenspiel von Marketing, Produktportfoliostrategieplanung und einer auf Entwicklung, Produktion und Vermarktung ausgerichteten Geschäftsplanung.

Es wäre in diesem Sinne irreführend von einer einzeln durchgeführten Konzeptphase zu reden. Bei nicht unüblichen Vorlaufzeiten von fünf bis zehn Jahren kann nicht davon ausgegangen werden, dass über diese Zeiträume hinweg eine kontinuierliche Finanzierung von Konzeptarbeiten für ein bestimmtes Element im Produktportfolio gewährleistet ist. Im Gegenteil scheint es sinnvoll zu sein, die Produktportfoliostrategie kontinuierlich fortzuschreiben und dabei vor Allem auf die Marktaussichten und sich bietende Geschäftsmöglichkeiten zu reagieren und Investitionen in Konzeptarbeiten

flexibel zu steuern. Da der Faktor Zeit bei der Durchdringung komplexer Probleme eine wesentliche Rolle spielt, kann es auch effizient sein, ein Systemkonzept für eine Weile liegen zu lassen. Wenn später die Arbeit wieder aufgenommen wird, werden die involvierten Systemingenieure in der Zwischenzeit erhaltene Informationen in den Systemkontext eingebunden und so Wissen generiert haben, für das keine expliziten Budgets ausgewiesen werden brauchten.

Eine Organisation mag bestimmte Reifegrade für Konzepte definieren. Für jede Iteration der Konzeptphase sind die Eingangsvoraussetzungen festzuhalten und die erwarteten Ergebniskriterien vorab zu definieren. Auf die Art der durchzuführenden Systems-Engineering-Tätigkeiten hat dies nur bedingt Einfluss, aber auf die erwartete Reife und den inhaltlichen Schwerpunkt der Ergebnisse. Die Haupttätigkeiten betreffen die Analyse der Systemumgebung für das Gesamtsystem und die Ableitung von operationellen Anforderungen, die Kunden und Nutzern der zukünftigen Produkte und Dienstleistungen Vorteile versprechen. Dazu werden Systemkonzepte definiert, die die vorgesehenen Missionen in der Systemumgebung effektiv erfüllen können. Ein weiterer Tätigkeitsschwerpunkt ist die Frage der Realisierbarkeit des vorgeschlagenen Systemkonzepts. Technische und nicht-technische Risiken sind möglichst umfassend zu identifizieren und zu bewerten. Wie diese Risiken behandelt und bewältigt werden sollen, ist ebenfalls zu klären. Zu den Realisierbarkeitsuntersuchungen gehört auch eine Aufstellung aller für die Umsetzung des Systemkonzeptes benötigten Technologien inklusive ihres momentanen Reifegrades und einer Prognose ihrer zukünftigen Verfügbarkeit.

Um die Investitionen in Konzeptarbeiten gering zu halten, sind hohe Ansprüche an die beteiligten Systemingenieure zu stellen. Die Hauptaufgabe ist, bisher unbekannte Probleme zu identifizieren und für alle Probleme, für die es bisher keine passenden Lösungen gibt, gangbare Wege aufzuzeigen. Kontraproduktiv, aber weniger herausfordernd wäre, bekannte Lösungsansätze und Lösungen nachzuvollziehen.

5 Definitionsphase

Die Definitionsphase dient der Definition einer effizienten Systemlösung. Zum Ende der Definitionsphase sollte eine qualitativ hochwertige Aussage zu den Lebenszykluskosten vorliegen. Dies erfordert die Definition aller Systemelemente in der Systemarchitektur ausschließlich der Systemelemente auf der Implementierungsebene, die von Zulieferern bezogen werden. Ein paralleles Arbeiten an allen Systemelementen ist normalerweise erforderlich. Die Fähigkeit zur agilen Durchführung von Iterationen für einzelne Systemelemente und über die Systemarchitektur hinweg ist eine wesentliche Voraussetzung, um in der Erreichung des Just-In-Time-Prinzips voranzukommen. Vermieden werden muss, dass einzelne Systemingenieure oder Teams auf Arbeit warten oder genötigt sind, ohne einen klar definierten Bezugspunkt zu arbeiten (siehe [Sc12] für eine Darstellung, wie dies erreicht werden kann). Ebenfalls hinderlich wären die Vermischung von Systemdefinition und Technologiearbeiten, da Technologiearbeiten viel Zeit Kosten und damit befasste Systemingenieure und Teams die Agilität in der Entwicklung der Systemarchitektur ausbremsen würden. Deshalb sollten Technologiearbeiten getrennt von der Systemdefinition durchgeführt werden.

Verträge mit Zulieferern sollten nur abschließend konkretisiert werden, wenn die Definitionsphase abgeschlossen ist, da alle späteren Änderungen auch die Zuliefererverträge betreffen und somit die Risiken für Zeit und Kosten massiv wachsen. Im Prinzip gilt die gleiche Überlegung auch für Kundenverträge.

Der in diesem Kapitel häufig verwendete Konjunktiv ist durch den gegenwärtig noch deutlich existierenden Abstand zur Wirklichkeit begründet. Es sind erhebliche Fortschritte in Arbeitsweise und Selbstverständnis der Systemingenieure notwendig, um das Just-In-Time-Prinzip im Systems Engineering zum Erfolg zu führen. Aber warum sollte hochqualifizierten Systemingenieuren nicht gelingen, was hochqualifizierten Facharbeitern in der Produktion gelungen ist?

6 Realisierungsphase

Die Realisierungsphase umfasst die Entwicklung der Systemelemente auf der Implementierungsebene, für die in der Regel andere Unternehmen eingebunden sind, die Systemintegration zum Gesamtsystem und in gewissen Fällen auch die Überführung in die reale Systemumgebung. In der Realisierungsphase ergeben sich durch den kontinuierlichen Wissenszuwachs immer wieder Änderungen der Systemdefinition mit der Folge, dass Tätigkeiten, die erstmals in der Definitionsphase durchgeführt wurden, wieder aufgenommen werden müssen. In der Regel sollte dies aber nicht als ein Rücksprung in die Definitionsphase gewertet werden. Kritisch wird es dann, wenn sich Änderungsbedarf am Verständnis der Systemumgebung und den operationellen Anforderungen an das Gesamtsystem ergeben, da dies letztlich Rücksprünge zu den Konzeptarbeiten erfordert und das geplante Produktportfolio in Frage gestellt ist.

Wie für die Definitionsphase, ist in der Realisierungsphase die Gesamtsystemperspektive des Lebenszyklusphasenmodells aus den gleichen Gründen wichtig. Nur dann kann es umfassend als Instrument zur Überwachung, Koordination und Synchronisation aller parallel ausgeführten Entwicklungstätigkeiten genutzt werden.

7 Technologieentwicklung

Ungeachtet staatlicher und möglicherweise von Kunden erbrachter Förderung bedeutet Technologieentwicklung den Einsatz eigener finanzieller, materieller und personeller Ressourcen. Das Risiko, die Aufwendungen nicht durch spätere Einnahmen rechtfertigen zu können, führt latent zu zurückhaltender Ausgabefreudigkeit. Andererseits ist mangelnde Reife von Einzeltechnologien einer der größten Risikofaktoren in der Produktentwicklung [GAO99]. In diesem Spannungsfeld passiert es immer wieder, dass Technologie- und Produktentwicklung miteinander verschmolzen werden, was oben bereits unter anderen Gesichtspunkten abgelehnt wurde.

Ein anderer Ansatz zur Risikominderung in der Technologieentwicklung bietet die Perspektive, Technologieentwicklung nicht nur im Hinblick auf einzelne Produkte zu definieren, sondern auf das gesamte Produktportfolio zu beziehen. Die Identifikation und

Planung einzelner Technologieprojekte fußt in diesem Fall auf den Technologiekatalogen aller Systemkonzepte und den vorhandenen innovativen Ideen. In die Priorisierung von Technologieprojekten mag dann auch einfließen, für wie viele Systemkonzepte einzelne Technologien relevant sind.

Der Fortschritt einzelner Technologieprojekte sollte nach den Reifegradstufen gemäß der ursprünglich von der NASA vorgeschlagenen Technology Maturity Level (TRL) [Ma95] bewertet werden. Unterschiedliche Einsatzszenarien einer Technologie besitzen dabei einen Einfluss auf die qualitativen und quantitativen Bewertungskriterien. Dies kann abhängig vom Zielprodukt zu unterschiedlichen Einstufungen führen. Bevor eine Technologie in der Produktentwicklung verwendet wird, sollte ein entsprechend hoher Reifegrad erreicht sein. In die Realisierungsphase sollte nicht mit unreifen Technologien eingestiegen werden. Die Technologieentwicklung erstreckt sich somit über Konzeptphase und Definitionsphase. Eine kontinuierliche Harmonisierung zwischen der Fortschreibung der Produktportfoliostrategie, der Systemdefinition und den Technologieprojekten ist erforderlich, um die mit der zeitlichen Parallelisierung einhergehenden Risiken zu beherrschen. Im ungünstigsten Fall kann es noch in der Definitionsphase zu neuen Erkenntnissen kommen, die eine Rejustierung der Technologieprojekte erfordern.

8 Schlussbemerkungen

In diesem Aufsatz wurde ein Lebenszyklusphasenmodell vorgeschlagen, das die Verbindung zwischen Wertschöpfungskettenmodellen und dem organisatorischen Unternehmensumfeld herstellt. Lebenszyklusphasenmodelle haben in dieser Funktion weiterhin eine hohe Bedeutung.

Zur Abdeckung der Systementwicklung wurden drei Phasen vorgeschlagen, die sich an den in der Literatur vorgeschlagenen Einteilungen orientieren, aber im Kontext einer marktgetriebenen Systementwicklung jeweils eine spezifische Ausprägung gewinnen. Dies führt im Vergleich zu anderen Darstellungen durchaus zu einer Bedeutungsverschiebung der einzelnen Phasen.

Technologie- und Systementwicklung sind voneinander getrennt. So lassen sich der Aufwand für die Fortschreibung der Produktportfoliostrategie minimieren und die Definitionsphase im Sinne des Just-In-Time Prinzips optimieren. Technologieentwicklung wird als Unternehmensaufgabe betrachtet, die sich am gesamten Produktportfolio einer Organisation ausrichtet.

Literaturverzeichnis

- [Bl04] Blanchard, B. S.: System Engineering Management. 3rd Edition. John Wiley and Sons, Hoboken, NJ, 2004.
- [BF06] Blanchard, B. S.; Fabrycky, W. J.: Systems Engineering and Analysis. 4th Edition. Prentice Hall, Upper Saddle River, NJ, 2006.
- [Bu09] Buede, D. M.: The Engineering Design of Systems – Models and Methods. 2nd Edition. John Wiley and Sons, Hoboken, NJ, 2009.
- [DK11] Driscoll, P. J.; Kucik, P.: System Life Cycle. In (Parnell, G. S.; Driscoll, P. J.; Hendersen, D. L. Hrsg.): Decision Making in Systems Engineering and Management. 2nd Edition. John Wiley and Sons, Hoboken, NJ, 2011.
- [FM91] Forsberg, K.; Mooz, H.: The Relationship of System Engineering to the Project Cycle. Proceedings of the National Council on Systems Engineering (NCOSE) Conference, Chattanooga, TN, 1991; S. 57-65.
- [FMC05] Forsberg, K.; Mooz, H.; Cotterman, H.: Visualizing Project Management: Models and Frameworks for Mastering Complex Systems. 3rd Edition. John Wiley and Sons Hoboken, NJ, 2005.
- [GAO99] GAO/NSIAD-99-162: Best Practices – Better Management of Technology Development Can Improve Weapon System Outcomes. 1999.
- [HNB02] Haberfellner, R., Nagel, P., Becker, M., Büchel, A., von Massow, H.: Systems Engineering - Methodik und Praxis. 11. Auflage. Verlag Industrielle Organisation, Zürich, 2002.
- [Ha62] Hall, A. D.: A Methodology for Systems Engineering. Van Nostrand-Reinhold, New York, NY, 1962.
- [Ha11] Haskins, C. (Hrsg.): Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. Version 3.2.2. INCOSE, San Diego CA, 2011.
- [Hy09] Hybertsen, D. W.: Model-Oriented Systems Engineering Science. CRC Press, Boca Raton, FL, 2009.
- [II08-1] ISO/IEC 12207-2008: Systems and Software Engineering – Software Life Cycle Processes. 2008.
- [II08-2] ISO/IEC 15288-2008: Systems and Software Engineering – System Life Cycle Processes. 2008.
- [III10] ISO/IEC TR 24748-1: Systems and Software Engineering – Life Cycle Management – Part 1: Guide for Life Cycle Management. 2010.
- [KSS11] Kossiakoff, A.; Sweet, W. N.; Seymour, S. J.; Biemer, S. M.: Systems Engineering – Principles and Practices. 2nd Edition. John Wiley and Sons, Hoboken, NJ, 2011.
- [LB03] Larman, C.; Basili, V. R.: Iterative and Incremental Development: A Brief History. IEEE Computer (June 2003), 2003; S. 2-11.
- [MR09] Maier, M. W.; Rechtin, E.: The Art of Systems Architecting. 3rd Edition. CRC Press, Boca Raton, FL, 2009.
- [Ma95] Mankins, J. C.: Technology Readiness Levels. White Paper. NASA, 1995.
- [Oh88] Ohno, T.: Toyota Production System: Beyond Large Scale Production. CRC Press, Boca Raton, FL, 1988.
- [Sc12] Scheithauer, D.: Managing Concurrency in Systems Engineering. Proc. 22nd INCOSE International Symposium, Rome, 2012.
- [SF13] Scheithauer, D.; Forsberg, K.: V-Model Views. Proc. 23rd INCOSE International Symposium, Philadelphia, PA, 2013.