Zeitliche und inhaltliche Konvergenz der Lösungsfindung als zentrale Herausforderung in hybriden Produktentwicklungsprozessen – eine empirische Analyse von Stanfords ME310-Prozessmodell

Frank Koppenhagen, Tim Blümel, Tobias Held, Christoph Wecht

Agilität und Konvergenz bei der Entwicklung physischer Produkte zu verbinden, stellt eine große Herausforderung dar. Basierend auf einem Design Thinking-Ansatz versucht das an der Stanford Universität entwickelte ME310-Produktentwicklungsprozessmodell den Widerspruch in den Zielsetzungen dieser beiden Entwicklungsprinzipien aufzulösen. Über einen qualitativen Fallstudienansatz haben wir die Praktikabilität des ME310-Prozessmodelles für die Entwicklung diskreter physischer Produkte untersucht. Der Beitrag beginnt mit der Beschreibung der grundlegenden entwicklungsmethodischen Prinzipien dieses hybriden Prozessmodelles. Anschließend werden die Ergebnisse unserer empirischen Untersuchung vorgestellt, in der wir die Koevolution von Problem- und Lösungsraum in zehn studentischen Projekten, in denen physische Produkte nach diesem Prozessmodell entwickelt wurden, detailliert nachvollzogen haben. Unsere explorative Studie zeigte, dass die Konvergenz der Lösungsfindung in den Entwicklungsprojekten nicht den theoretischen Vorgaben des Prozessmodelles entsprach, sondern dass es auch in den späten Phasen des Entwicklungsprozesses noch häufig zu Konzeptänderungen bzw. Änderungen des zugrundeliegenden Problemverständnisses kam. Abschließend werden mögliche Ansatzpunkte für die weitergehende Forschung formuliert, welche die Integration von Methoden aus der Konzeptphase des Systems Engineering in das ME310-Prozessmodell beinhalten.

Keywords: Design Thinking, Hybride Produktentwicklung, Agile Produktentwicklung, Produktentwicklungsprozesse, Produktentwicklungsmethodik

Einleitung

Obwohl Design Thinking in den letzten Jahren zunehmend auch als geeigneter Ansatz für die agile, nutzerzentrierte Entwicklung physischer Produkte gesehen wird (Brown: 2008; Luchs, Swan & Griffin: 2016), hat es bislang kaum Beachtung in dem akademischen Diskurs der ingenieurwissenschaftlichen Konstruktionsmethodik gefunden. Die

wenigen Arbeiten, die Design Thinking entwicklungsmethodisch analysieren, wie z. B. Gericke, Beinke, Gemmer & Blessing (2010) und Schüttoff, Herrmann, Roth & Binz (2019), vergleichen zudem meist unterschiedliche Betrachtungsebenen miteinander: eine Entwicklungssystematik zur Strukturierung eines vollständigen Produktentwicklungsprozesses auf der Seite des Systems Engineering mit einem iterativen Arbeitsprozess zur Lösung von Teilproblemen auf der Seite des Design Thinking. Dies ist zum einen auf das uneinheitliche Verständnis von Design Thinking zurückzuführen, das je nach akademischer Perspektive und praktischem Anwendungskontext stark variiert und eine entwicklungsmethodische Einordnung des Ansatzes erschwert. Zum anderen liegt es auch darin begründet, dass die meisten Design Thinking Prozessmodelle keinen vollständigen Produktentwicklungsprozess abbilden. Eine Ausnahme stellt das am Center for Design Research der Stanford Universität entwickelte ME310-Prozessmodell dar, welches der zentrale Untersuchungsgegenstand dieser Arbeit ist. Dieses hybride Produktentwicklungsprozessmodell basiert im Kern auf einem agilen Design Thinking-Ansatz, definiert aber gleichzeitig Meilensteine, um den gesamten Produktentwicklungsprozess zu strukturieren und die Konvergenz der Lösungsentwicklung zu steuern.

Ziel dieses Beitrages ist es, die Praktikabilität dieses Prozessmodelles für die Entwicklung diskreter physischer Produkte zu untersuchen. Da das ME310-Prozessmodell versucht, Agilität und Konvergenz bei der Neuproduktentwicklung zu verbinden, steht dabei die Frage im Mittelpunkt, wie die Lösungsentwicklung konvergiert, und ob diese Konvergenz den theoretischen Vorgaben des Prozessmodelles entspricht. Hierfür wurden zehn Produktentwicklungsprojekte analysiert, in denen Graduate Students der Stanford Universität physische Produkte auf Basis realer Aufgabenstellungen kooperierender Industrieunternehmen entwickelt haben. Im Rahmen dieser empirischen Untersuchung wurden die technischen Konzepte von 177 Prototypen und das ihnen zugrundeliegende Problemverständnis detailliert analysiert. Auf diese Weise konnte die Koevolution von Problem- und Lösungsraum in den Entwicklungsprojekten vollständig nachvollzogen und mit den theoretischen Vorgaben des ME310-Prozessmodelles verglichen werden.

Theoretischer Bezugsrahmen

Produktentwicklung als Transformationsprozess zwischen Problem- und Lösungsraum

Produktentwicklungsprozesse sind kreative Problemlösungsprozesse, bei denen bereits das zugrundeliegende Problem häufig nicht klar und abschließend definiert werden kann. Damit ist sowohl der Problemraum, als kognitiver Verständnisraum, als auch

der Lösungsraum, als technischer Möglichkeitsraum, grundsätzlich offen. Die Entwicklung von Problem- und Lösungsraum bedingt sich dabei gegenseitig, da das Problemverständnis von den bekannten oder denkbaren Lösungsmöglichkeiten abhängt (Rittel & Webber: 1973). Die Entwicklung eines Problemverständnisses und seine Abbildung durch Anforderungen auf unterschiedlichen Aggregationsebenen ist damit ein erster kreativer Akt, der den Lösungsraum begrenzt und bereits eine erste Vision der anzustrebenden Lösung enthält (Abbildung 1).

Abbildung 1: Problemverständnis, Anforderungen und Lösungsraum
(vgl. auch Ponn & Lindemann (2011))

Ein Produktentwicklungsprozess kann somit als Transformationsprozess beschrieben werden, bei dem ausgehend von einem tatsächlichen Nutzerbedürfnis zunächst ein Problemverständnis entwickelt und mit Hilfe von Anforderungen abgebildet werden muss, bevor die eigentliche Lösung entwickelt werden kann, die schließlich das Nutzerbedürfnis befriedigen soll (vgl. auch Abbildung 5). Bei jedem Schritt dieses Transformationsprozesses kann es zu Abweichungen und Informationsverlusten kommen, die zu einer späteren Inkongruenz zwischen dem Entwicklungsergebnis und dem Nutzerbedürfnis führen können. Agile Produktentwicklungsverfahren wie Design Thinking versuchen, einer solchen Diskrepanz durch eine parallele iterative Entwicklung von Problemverständnis und -lösung vorzubeugen. Diese Koevolution von Problem- und Lösungsraum (Dorst & Cross: 2001) wird durch Nutzerinteraktionstests gesteuert, in denen Prototypen unterschiedliche Aspekte des entstehenden Produktes repräsentieren, um Nutzerbedürfnisse zu verstehen und Anforderungen abzuleiten.

Semantische Ebenen und methodische Taxonomiestufen von Design Thinking

Das Verständnis von Design Thinking, das diesem Aufsatz zugrunde liegt, besteht aus einem Modell, welches drei semantische Ebenen umfasst, die hierarchisch aufeinander aufbauen. Diese semantischen Ebenen repräsentieren gleichzeitig unterschiedliche methodische Taxonomiestufen im Kontext der Produktentwicklung (Abbildung 2).

Abbildung 2: Semantische Ebenen und methodische Taxonomiestufen von Design Thinking

Die fundamentale Ebene beschreibt die handlungsleitenden Prinzipien, die konstituierend für die Denkhaltung und die Kultur des Design Thinking Ansatzes sind (vgl. z. B. Carlgren, Rauth & Elmquist (2016)). Die operative Ebene beschreibt den Kernzyklus von Design Thinking, einen iterativen Arbeitsprozess, der die operative Umsetzung der fundamentalen Prinzipien der grundlegenden Ebene darstellt. Dieser Kernzyklus stellt aus entwicklungsmethodischer Sicht einen Prozess zur Lösung von Teilproblemen dar, der im Sinne von Haberfellner, Weck, Fricke & Vössner (2019) auch als „Micro-logic“ verstanden werden kann. Für den Design Thinking Kernzyklus finden sich in der Literatur unterschiedliche Darstellungen und Beschreibungen, bei denen sowohl die Anzahl als

auch die Benennung der einzelnen Prozessschritte variieren. Dennoch sind diese Prozesse nahezu inhaltsgleich und verweisen im Kern auf ein ähnliches Vorgehen. Auf der dritten Ebene, der Projektebene, sind schließlich die übergeordneten Prozessmodelle bzw. Phasenmodelle angesiedelt, welche einen vollständigen Produktentwicklungsprozess strukturieren und damit die „*Macro-logic*“ (Haberfellner, Weck, Fricke & Vössner: 2019) repräsentieren. Im Gegensatz zum operativen Kernzyklus werden sie im Laufe eines Projektes nur einmal durchlaufen.

Design Thinking Prozessmodelle müssen zum einen die fundamentalen Prinzipien von Design Thinking auf die übergeordnete Projektebene zur Strukturierung des Gesamtprozesses übertragen und den iterativen Kernzyklus für die Lösung von Teilproblemen innerhalb der Phasen integrieren. Zum anderen müssen sie aber auch die Konvergenz der Lösungsentwicklung auf der Gesamtsystemebene über Meilensteine steuern. Die Meilensteine müssen zudem die Zusammenarbeit der Prozessbeteiligten synchronisieren und ihnen eine Orientierung in Bezug auf das Projektstadium geben, aus denen sie die anstehenden Aufgaben und die verbleibenden Freiheitsgrade für die Gestaltung Ihres Verantwortungsbereiches ableiten können.

Das einzige den Autoren derzeit bekannte Prozessmodell, das diese Kriterien erfüllt, ist das am Center for Design Research der Stanford Universität entwickelte ME310-Prozessmodell. Nach diesem Prozessmodell entwickeln Graduate Students, ausgehend von realen Aufgabenstellungen kooperierender Industrieunternehmen, in einer projektbasierten Lehrveranstaltung innovative Produkte (ME310 ist die Kursnummer dieser Lehrveranstaltung). In den Projekten, die über drei Quarter (jeweils zehn Wochen) gehen, arbeitet jeweils ein Team der Stanford Universität mit einem Team einer ausländischen Partneruniversität zusammen. Die Teams werden dabei von Professoren, Lehrbeauftragten und Course Assistants (studentische Tutoren, die die Lehrveranstaltung bereits selber absolviert haben) betreut.

Das ME310-Prozessmodell

Die Darstellung der einzelnen Phasen mit den ihnen zugeordneten Aktivitäten und Arbeitsergebnissen in Abbildung 3 stützt sich im Wesentlichen auf den ME310 ABC Course Reader (Kenyon, Cutkowsky & Leifer: o.J.). Nach einer intensiven Problemerkundung in der Needfinding Phase (NF) erfolgt durch die Analyse existierender Produkte in der anschließenden Benchmarking Phase (BM) bereits eine erste Verbindung von Problem- und Lösungsraum, mit der eine Produktvision entwickelt wird.

	Phase	Aktivitäten	Ergebnisse
Teilsystem	Needfinding	Nutzerzentrierte Problemraumerkundung Ermitteln der Nutzer und Nutzerbedürfnisse sowie des Nutzungskontextes	Initiales Problemverständnis
	Benchmarking	Erste Verbindung von Problem- und Lösungsraum Analyse der Stärken und Schwächen existierender Produkte	Produktvision
	CEP / CFP	Koevolution von Problem- und Lösungsraum CEP: Anforderungen ableiten und validieren CFP: Tauglichkeit von Wirkprinzipien verifizieren	Anforderungen (Abschließendes Problemverständnis)
	Dark Horse Prototyp	Überprüfen des Problemverständnisses Aufbauen und Testen riskanter und radikal neuer Lösungen am Rande und außerhalb des vorläufigen Lösungsraumes	Validiertes Problemverständnis
Gesamtsystem	Funky System Prototyp	Aufbau und Erprobung eines Wirkkonzeptes Verknüpfung von Funktionsträger aus der CFP- und ggfs. aus der Dark Horse-Phase zu einem Wirkkonzept, Konzeptabsicherung	Konzept (Prinzipielle Lösung)
	Functional System Protot.	Aufbau und Erprobung einer vollständigen Lösung Vervollständigung Funktionsumfang, Funktionsoptimierung, Konsolidierung der Systemintegration, formale Ästhetik	Gesamtentwurf (Gestalterische Lösung)
	Final Prototyp	Aufbau der finalen Produktrepräsentation Abbildung des vollständigen Nutzererlebnisses des späteren Produktes, Detailgestaltung, Ausführungs- u. Nutzungsunterlagen	Vollständige Produktrepräsentation

Abbildung 3: Phasen, Aktivitäten und Arbeitsergebnisse des ME310-Prozessmodelles

Der in einigen Veröffentlichungen, z. B. Domingo, et al. (2020), zusätzlich aufgeführte *Part-X-is-finished-Prototype* bezeichnet die Fertigstellung eines von den Studierenden selbstgewählten Bauteils in der Phase des Final Prototyps. Dieser nur auf ein Bauteil bezogene physisch realisierte Design Freeze soll den Kreischluss wechselseitiger geometrisch struktureller Abhängigkeiten in der Produktarchitektur durchbrechen und den Kristallisationspunkt für den Aufbau des Final Prototyps darstellen. Der *Part-X-is-finished-Prototype* ist damit als Zwischenmeilenstein innerhalb der Final Prototyp Phase zu betrachten.

Die eigentliche Lösungsentwicklung beginnt dann in der CEP-/CFP-Phase auf Teilsystemebene. Die Critical Experience Prototypen (CEP) sollen kritische Kernelemente der Nutzererfahrung der entwickelten Produktvision erlebbar machen. CEPs dienen zum Verständnis des Problemraumes und werden zur Ableitung und Validierung von Nutzeranforderungen eingesetzt. Hierzu werden häufig „Wizard-of-Oz“-Prototypen verwendet, bei denen das Nutzererlebnis einer Funktion simuliert wird, ohne dass für diese Funktion bereits ein technischer Funktionsträger entwickelt wurde. Mit Hilfe der Critical Function Prototypen (CFP) wird die Tauglichkeit von Wirkprinzipien lösungsbestimmender Funktionsträger eines möglichen Gesamtkonzeptes bewertet. In dieser Phase werden mehrere CEPs und CFPs gebaut und iterativ weiterentwickelt. Die CEP-/CFP-Phase endet mit der Formulierung funktionaler und physischer Anforderungen.

Diese Anforderungen bilden das abschließende Problemverständnis ab und beinhalten eine „*kohärente Vision*“ (Domingo, et al.: 2020) des zu entwickelnden Produktes, die Nutzerbegehren und technische Machbarkeit miteinander verbindet.

Die anschließende Phase des Dark Horse Prototyps (DHP) soll das durch die Anforderungen abgebildete Problemverständnis überprüfen und verhindern, dass der Lösungsraum durch bestehende Konventionen oder dem Streben nach geringem Risiko zu früh zu stark eingegrenzt wird. Dazu werden Prototypen aufgebaut, die ein besonders riskantes, radikales, unkonventionelles oder vielleicht auch zunächst nicht umsetzbar erscheinendes Lösungsprinzip beinhalten (Bushnell, Steber, Matta, Cutkosky & Leifer: 2013). Die Erstellung dieser Dark Horse Prototypen zwingt das Entwicklungsteam, sich von bewussten oder unbewussten Vorfixierungen zu lösen und das bisherige Problemverständnis kritisch zu hinterfragen (Domingo, et al.: 2020). Das Ergebnis dieser Phase soll ein validiertes Problemverständnis sein, das einen festen, belastbaren Rahmen für die nachfolgende Lösungsentwicklung auf der Gesamtsystemebene darstellt. Der Funky System Prototyp (FKP) ist der erste Prototyp auf der Gesamtsystemebene, mit dem das Konzept definiert und die Konzepttauglichkeit abgesichert werden soll. Hierzu werden die erfolversprechendsten Funktionsträger aus der CFP-Phase (und ggfs. der Dark Horse Phase) zu einer Gesamtlösung kombiniert. Der Funky System Prototyp bildet in der Regel noch nicht den vollständigen Funktionsumfang ab, sondern konzentriert sich auf die lösungsbestimmenden Hauptfunktionen. Er dient nur der technisch funktionalen Verifikation des Wirkkonzeptes. Formal ästhetische Gestaltungsaspekte spielen bei dem Funky System Prototyp noch keine Rolle. In der anschließenden Phase des Functional System Prototyps (FCP) soll das Konzept des Funky System Prototyp detailliert und optimiert werden. Der Functional System Prototyp soll den vollständigen Funktionsumfang abbilden und dient der Konsolidierung der Systemintegration sowie der Optimierung auf Gesamt- und Teilfunktionsebene. Er stellt den Gesamtentwurf dar und soll bereits eine produktnahe Wertanmutung aufweisen. Der Final Prototyp stellt schließlich den Abschluss der Entwicklung dar und soll das vollständige Nutzererlebnis eines späteren industriell realisierten Produktes repräsentieren.

Grundlegende Prinzipien des ME310-Prozessmodelles

In diesem Abschnitt sollen die beiden Grundprinzipien des ME310-Prozessmodelles erläutert werden, die seine entwicklungsmethodische Einordnung begründen: (1) Die Koevolution von Problem- und Lösungsraum und (2) die prototypenbasierte Lösungsentwicklung.

Koevolution von Problem- und Lösungsraum

Während die Ansätze des klassischen Systems Engineering meist eine klare Trennung von Problemanalyse und Lösungsentwicklung vorsehen, ist das ME310-Prozessmodell über weite Strecken von einer gemeinsamen Entwicklung dieser beiden Sphären geprägt (Abbildung 4).

Abbildung 4: Problem- und Lösungsraumentwicklung im ME310-Prozessmodell (oben) und bei Pahl/Beitz (unten)

Hierzu werden in der CEP-/CFP-Phase zwei parallele Regelkreise etabliert: ein Validierungs- und ein Verifizierungsregelkreis (Abbildung 5). Die Nutzerinteraktionstests auf Basis der CEPs verbessern das Problemverständnis und ermöglichen die Ableitung und Validierung von Anforderungen, die zum Teil erst in der praktischen Erprobung früher Produktrepräsentationen deutlich werden (Leifer & Steinert: 2011). Die CEPs etablieren damit einen Validierungs-Regelkreis, der verhindert, dass am Kunden „vorbeientwickelt“ wird. Gleichzeitig sichern die CFPs die Tauglichkeit von Wirkprinzipien und die Anforderungserfüllung von Funktionsträgern für lösungsbestimmende Hauptfunktionen ab. Die CFPs etablieren damit einen Verifizierungs-Regelkreis zwischen den Anforderungen und dem Entwicklungsgegenstand. In einem iterativen Entwicklungsprozess, der von einer stringenten Nutzerzentrierung geleitet wird, setzen die CEPs und die CFPs in ihrem Zusammenwirken das Prinzip der Koevolution von Problem- und Lösungsraum um. Diese Phase des Prozessmodelles zeigt somit eine methodische Selbstähnlichkeit zum iterativen Kernzyklus von Design Thinking.

Abbildung 5: Validierungs- und Verifizierungsregelkreis durch CEP und CFP

Das Formulieren von Anforderung, also die Abbildung des Problemverständnisses, erfolgt im ME310-Prozessmodell im Gegensatz zum Systems Engineering nicht vor Beginn der Lösungsentwicklung, sondern erst nach Abschluss der Lösungsentwicklung auf Teilsystemebene. Ähnlich wie in der ursprünglichen Entwicklungsmethodik von Pahl/Beitz (Pahl, Beitz, Feldhusen & Grote: 2007) wird dieses Problemverständnis zunächst kritisch überprüft, bevor es als fester Rahmen für die weitere Lösungsentwicklung gesetzt wird. Anders als bei Pahl/Beitz geschieht dies jedoch nicht über eine theoretische Abstraktion der Problemstellung, sondern über das Aufbauen und Bewerten konkreter Lösungen am Rande oder außerhalb der vorläufigen Lösungsraumbegrenzung, sprich der Dark Horse Prototypen. Beide Ansätze verfolgen jedoch dasselbe Ziel: Das Vermeiden einer zu frühen, zu engen Eingrenzung von Problemverständnis und Lösungsraum.

Prototypenbasierte Lösungsentwicklung

In der ME310-Methodik ist der Aufbau der Prototypen integraler und konstituierender Bestandteil der Entwicklungsmethodik. Innerhalb der Phasen werden sie nicht nur zum Verifizieren und Optimieren, sondern auch zum Explorieren und Kommunizieren

von Lösungen eingesetzt. Gleichzeitig definieren die unterschiedlichen Prototypenarten die Ziele für die Meilensteine, die den Gesamtprozess strukturieren. Die Prototypen sind damit phasen- und prozessübergreifend die Haupttreiber des Entwicklungsprozesses.

Abbildung 6: Theoretischer Konvergenzpfad des ME310-Prozessmodelles

Da die Realisierung eines komplexen physischen Produktes mit einem großen Aufwand verbunden ist, kann die Entwicklung auf der Gesamtsystemebene in der Regel nicht mehr vollständig agil erfolgen. Im ME310-Prozessmodell verläuft die Entwicklung daher nur während der Teilfunktionsprototypenphasen (CEP/CFP und DHP) vollständig agil. In dieser Phase bleiben das Problemverständnis und damit auch die Lösungsraumbegrenzung volatil. Während der Systemprototypenphasen wird hingegen eine zunehmende Konvergenz und Konsolidierung des Entwicklungsergebnisses angestrebt, die durch das Ergänzen und Detaillieren von Anforderungen gesteuert wird, was eine zunehmende Lösungsraumbegrenzung zur Folge hat (Abbildung 6). Das ME310-Prozessmodell ist damit aus entwicklungsmethodischer Sicht ein agiles Stage-Gate-Hybrid-Prozessmodell (Cooper & Sommer: 2016). Die frühen Teilfunktionsprototypen werden im Rahmen des ME310-Prozesses meist ohne vorausgehende virtuelle Entwicklung aufgebaut. Eine intensive virtuelle Entwicklung setzt in der Regel erst vor dem Aufbau der späteren Systemprototypen ein.

Für den Prozess der kreativen Lösungsentwicklung bringt der frühe Einsatz von Prototypen zahlreiche Vorteile (Brereton & McGarry: 2000; Viswanathan & Linsey: 2012). In Bezug auf die Lösungsraumexploration stellt ein prototypenbasierter Ansatz jedoch ein zweischneidiges Schwert dar. Da die Entwicklung im klassischen Systems Engineering in der frühen Phase in der Regel weitgehend virtuell, sprich ohne den Aufbau physischer Prototypen, erfolgt, ermöglicht sie zumindest bis zur Konzeptauswahl eine parallele Entwicklung, also die Erarbeitung und Evaluation konkurrierender Lösungen, sowohl auf der Teilsystem- als auch auf der Gesamtsystemebene. Die für das Systems Engineering kennzeichnende Dekomposition des Gesamtsystems erlaubt dabei eine systematische Variation und Kombination von Lösungsbestandteilen, die das theoretische Explorieren eines großen Lösungsraumes ermöglicht. Erst ab der Konzeptauswahl erfolgt die Entwicklung der weiteren Lösung nach einem „point-based“ (Sobek II, Ward & Liker: 1999) orientierten Ansatz. Im Rahmen der ME310-Entwicklungsmethodik findet die Entwicklung von Beginn an auf der Basis von Prototypen statt. Dies generiert zwar – bezogen auf die einzelne realisierte Lösung – einen größeren Erkenntnisgewinn als eine rein virtuelle Entwicklung, ist aber mit einem deutlich größeren Aufwand verbunden, vor allem auf der Ebene der Systemprototypen. Eine parallele Entwicklung ist im ME310-Prozessmodell daher nur auf der Ebene der Teilfunktionsprototypen vorgesehen. Ein paralleles Aufbauen und Bewerten konkurrierender Konzepte auf der Gesamtsystemebene erfolgt nicht, da dies mit einem prohibitiven Aufwand verbunden wäre. Insofern ist der Ansatz der ME310-Entwicklungsmethodik stärker point-based orientiert als die konventionelle Entwicklungsmethodik. Der zum Zeitpunkt des Konzeptentscheides explorierte Lösungsraum, also das Wissen über den Entwicklungsgegenstand, ist insgesamt kleiner. Dies kann die Stabilität des Konzeptentscheides reduzieren.

Außerdem erfährt die Gesamtsystementwicklung im Rahmen des ME310-Prozessmodells keine methodische Unterstützung. Vor dem Aufbau der CEPs und CFPs erfolgt keine funktionale Modellierung des Gesamtsystems, also kein Erstellen von Funktionsstrukturen. Damit fehlt zum einen die Basis für eine explizite funktionale Dekomposition des Gesamtsystems. Zum anderen erschwert es die Analyse der funktionalen Beziehungen im Gesamtsystem und damit auch die Beurteilung der technischen Wechselwirkungen zwischen den später definierten Funktionsträgern. Das ME310-Prozessmodell bietet somit keine explizite methodische Unterstützung für zwei Aufgaben, die von zentraler Bedeutung für den Konzeptentwurf sind: (1) das Identifizieren der lösungsbestimmenden Elemente der Produktvision, die in der CEP-/CFP-Phase aufgebaut werden sollen, und (2) das sinnvolle und verträgliche Kombinieren ausgewählter CFPs zur ersten Gesamtlösung, dem Funky System Prototype. Beides muss von dem

Entwicklungsteam implizit berücksichtigt werden, was insbesondere bei komplexen Systemen eine große Herausforderung darstellt. Denn eine Gesamtsystemmodellierung und eine darauf aufbauende Dekomposition in Teilsysteme ist eine wichtige Voraussetzung für den Umgang mit Komplexität, da die Entwicklungsaufgabe auf diese Weise in handhabbare Teilaufgaben zerlegt werden kann (Kopenhagen: 2004). Das Fehlen einer solchen expliziten Vorgehensweise zur Komplexitätsreduktion limitiert die Anwendungsmöglichkeiten des ME310-Ansatzes.

Methodik

Im Rahmen der empirischen Untersuchung wurden die Projektberichte, in denen die studentischen Teams den Entwicklungsverlauf der Projekte zum Abschluss eines jeden Quarters detailliert dokumentieren, ausgewertet. Da sich die Untersuchung auf die Entwicklung physischer Produkte konzentriert, wurde zunächst eine Vorauswahl auf Basis des Entwicklungsgegenstandes getroffen. Hierzu wurden 124 Entwicklungsprojekte aus den Jahren 2006-2019 im Hinblick auf den Entwicklungsgegenstand klassifiziert und in die Kategorien physische Produkte, Softwareapplikationen und Service-/Geschäftsprozessmodelle unterteilt. Von den 55 Projekten, welche die Entwicklung physischer Produkte zum Ziel hatten, wurden schließlich zehn Projekte aus dem Zeitraum 2014–2019 ausgewählt. Bei dieser Auswahl wurde zum einen darauf geachtet, dass die Entwicklungsdokumentation ein lückenloses Nachvollziehen der Entwicklungspfade zuließ. Zum anderen sollte kein Industriesponsor mehrfach vertreten sein, um den unternehmensspezifischen Einfluss auf die Untersuchung zu begrenzen. Folgende Entwicklungsprojekte haben Eingang in die empirische Analyse gefunden (jeweils gekennzeichnet durch den Namen des Industriesponsors und das Jahr der Projektfertigstellung): VolvoCE (2014), Mabe (2014), Ford (2016), ShoeInn (2016), Renault (2016), IKEA (2016), Audi (2017), Safran (2018), Panasonic (2019), Volkswagen (2019). Die empirische Auswertung der Entwicklungsverläufe umfasst damit dreißig Projektberichte mit insgesamt 3.578 Seiten.

Auf Basis der Projektberichte wurde das technische Wirkkonzept eines jeden Prototyps und das ihm zugrundeliegende Problemverständnis bestimmt. Das zugrundeliegende Problemverständnis wurde den verbalen Beschreibungen in den Projektberichten entnommen. Für die Analyse des technischen Wirkkonzeptes der Prototypen, das durch die Wirkprinzipien der Funktionsträger und ihre Verknüpfung zur Erfüllung der löungsbestimmenden Hauptfunktionen des Prototyps definiert wird, wurden darüber hinaus noch die in den Berichten abgebildeten Entwicklungsartefakte, wie z. B. Skizzen, Diagramme, technische Zeichnungen, Screenshots von 3D-CAD-Modellen und Fotos

der aufgebauten Prototypen ausgewertet. Auf diese Weise wurden die Prototypenpfade in den Entwicklungsprojekten nachvollzogen und festgestellt, wann es zu Änderungen im Problemverständnis oder zu Konzeptänderungen im Lösungsraum kam. Darüber hinaus wurde untersucht, ob diese Änderungen durch Erkenntnisse aus der Bewertung des vorangegangenen Prototyps beeinflusst wurden und ob sie teamintern, also durch die Mitglieder des Entwicklungsteams selber, oder durch die Mitglieder des Teaching Teams bzw. Vertreter des Industriesponsors veranlasst wurden. Der gesamte beschriebene Auswahl- und Analyseprozess wurde dabei zunächst unabhängig von zwei Wissenschaftlern mit mehrjähriger Berufserfahrung in der industriellen Produktentwicklung durchgeführt. Traten Differenzen in ihrer Bewertung auf, so wurde unter Hinzunahme eines dritten Experten eine finale Festlegung vorgenommen.

Ergebnisse und Diskussion

Abbildung 7 zeigt exemplarisch die Prototypenpfade eines Projektes, in denen die Zusammenhänge zwischen den Prototypen, Konzeptänderungen sowie Änderungen des zugrunde liegenden Problemverständnisses dargestellt sind. Die Abbildung illustriert auch, mit welchem Detaillierungsgrad die Entwicklungsverläufe der einzelnen Projekte analysiert wurden. In den folgenden beiden Abschnitten wird die Gesamtauswertung für die Problem- und Lösungsraumentwicklung über alle Projekte vorgestellt.

Die Entwicklung des Problemverständnisses

Abbildung 8 zeigt, wie oft und in welchen Phasen das Problemverständnis in den Projekten geändert wurde und in welcher Phase das finale Problemverständnis definiert wurde. In der CEP-/CFP-Phase, in der nach dem ME310-Prozessmodell die eigentliche Koevolution von Problem- und Lösungsraum stattfinden soll, kam es nur relativ selten zu einer Anpassung des zugrundeliegenden Problemverständnisses. Die meisten Änderungen des Problemverständnisses traten in der Dark Horse Phase auf. Dies steht in Übereinstimmung mit dem Prozessmodell, da das kritische Hinterfragen des Problemverständnisses in dieser Phase intendiert ist. Das Problemverständnis wurde jedoch entgegen dem Prozessmodell auch in den nachfolgenden Systemprototypenphasen, insbesondere in der Funky System Prototyp Phase, noch häufig geändert. Die meisten Änderungen wurden dabei teamintern ausgelöst (63 %). Nur bei vier von zehn untersuchten Projekten stand das finale Problemverständnis dem Prozessmodell entsprechend tatsächlich nach Abschluss der Dark Horse Phase fest. Bei drei Projekten wurde das Problemverständnis hingegen erst in der Phase des Final Prototyps endgültig festgelegt.

Abbildung 7: Visualisierung der Prototypenpfade des VW 2019 Projektes

Abbildung 8: Änderungen des Problemverständnisses und Phase der finalen Definition

Eine Erklärung für die häufigen Problemraumänderungen in den Systemprototypenphasen kann darin bestehen, dass die in der CEP- und CFP-Phase aufgebauten Prototypen jeweils nur die Wirkprinzipien oder die Nutzererfahrung einzelner Teilfunktionen abbilden. Sie bieten damit keine ausreichende Basis für Fragestellungen, die zur Entwicklung eines umfassenden Problemverständnisses auf der Gesamtsystemebene

beitragen können. Dies könnte auch die relativ hohe Zahl an Problemraumänderungen in der Funky System Prototyp Phase erklären, in der zum ersten Mal ein Prototyp aufgebaut wird, der zumindest die lösungsbestimmenden Hauptfunktionen und ihr Zusammenwirken auf der Gesamtsystemebene abbildet.

Die Konzeptentwicklung

Abbildung 9 zeigt, beginnend mit der Funky System Prototyp Phase, wann und wie häufig das technische Lösungskonzept geändert wurde und in welcher Phase das Konzept des Final Prototyps entwickelt wurde. Das initial in der Funky System Prototyp Phase definierte Konzept wurde noch in der Funky System Prototyp Phase selber häufig gewechselt. Auch nach Abschluss der Funky System Prototyp Phase zeigte sich eine hohe Volatilität der technischen Konzepte. Bemerkenswert ist dabei vor allem, dass es auch noch in der Phase des Final Prototyps zu 16 Konzeptänderungen kam, die sich auf acht Entwicklungsprojekte verteilten. Auch die Konzeptänderungen wurden zum größten Teil (74 %) teamintern ausgelöst. Nur bei einem Projekt stand das finale Konzept den Vorgaben des Prozessmodelles entsprechend bereits nach Abschluss der Funky System Prototyp Phase fest. Bei acht Projekten wurde das Konzept des Final Prototyps hingegen erst in der Phase des Final Prototyps definiert. Darüber hinaus konnte festgestellt werden, dass nur in einem von zehn Projekten überhaupt ein Funktionsträger aus der CFP-Phase Bestandteil des Funky System Prototyps war. In allen betrachteten Projekten wurden in den späteren Systemprototypenphasen erneut CEPs und/oder CFPs aufgebaut.

Erwartungsgemäß zeigt sich, dass Änderungen des zugrundeliegenden Problemverständnisses, die häufig einen zumindest teilweisen Neustart der Entwicklung erforderlich machen, auch Änderungen des technischen Lösungskonzeptes nach sich ziehen. Die Konzeptentwicklung zeigte dabei im Vergleich zur Definition des Problemverständnisses eine insgesamt höhere und länger andauernde Volatilität. So wurde das Konzept im Durchschnitt noch 1,7-mal nach dem Festlegen des finalen Problemverständnisses geändert.

Eine Ursache für die geringe Übernahme von Funktionsträgern aus der CFP-Phase in den Funky System Prototyp kann in der fehlenden methodischen Unterstützung des Gesamtsystementwurfes liegen, die weiter oben beschrieben wurde. Da im ME310-Prozessmodell eine explizite Analyse der Wirkzusammenhänge zwischen den Funktionsträgern fehlt, wird die Beurteilung, ob diese sinnvoll und verträglich zu einer Gesamtlösung kombiniert werden können, erheblich erschwert. Die häufigen Konzeptwechsel auf der Gesamtsystemebene liegen vermutlich in der stärkeren point-based-

Orientierung begründet, die mit dem prototypenbasierten Entwicklungsansatz verbunden ist. Im Gegensatz zum Systems Engineering erfolgt keine parallele Entwicklung und Bewertung unterschiedlicher Konzepte, so dass der explorierte Lösungsraum auf der Gesamtsystemebene zum Zeitpunkt des Konzeptentscheides vergleichsweise kleiner ist, was sich nachteilig auf die Stabilität der Konzeptentscheidung auswirkt.

Abbildung 9: Änderungen des Konzeptes und Phase der finalen Definition

Zusammenfassung und Ausblick

Durch die eingehende Analyse der Projektberichte wurden die Entwicklungspfade von zehn studentischen ME310-Projekten detailliert nachverfolgt und mit den theoretischen Vorgaben des ME310-Prozessmodelles verglichen. Hierbei stellte sich heraus, dass die beobachteten Entwicklungsverläufe nicht dem Konvergenzpfad folgten, der durch das Prozessmodell vorgegeben wird. Auch in den späten Phasen der Entwicklung kam es noch häufig zu Änderungen der technischen Konzepte und des zugrundeliegenden Problemverständnisses. Eine mögliche Erklärung für die späte Konvergenz der Lösungsentwicklung kann in der unzureichenden methodischen Unterstützung der Gesamtsystementwicklung und der fehlenden theoretischen Exploration eines größeren Lösungsraumes vor dem Konzeptentscheid bestehen. Beides erhöht die prognostische Unsicherheit bei der Konzeptdefinition und reduziert damit die Stabilität des Konzeptentscheides.

Im Rahmen weiterführender Arbeiten sollten die Ergebnisse unserer Untersuchung durch die Analyse zusätzlicher Projektberichte zunächst auf einer breiteren empirischen Basis validiert werden. Um die festgestellten methodischen Defizite des ME310-Prozessmodelles zu überwinden, sollte darüber hinaus die Möglichkeit untersucht werden, ausgewählte Methoden aus der Konzeptphase des Systems Engineering in das ME310-Prozessmodell zu integrieren. Aufbauend auf einer theoretischen Gesamt-

systemmodellierung mit expliziter funktionaler Dekomposition könnten dabei z.B. ausgewählte Funktionen mit Hilfe von CFP und CEP dargestellt und erprobt werden. Durch die nachfolgende parallele virtuelle Entwicklung mehrerer Konzepte könnte dann ein größerer Lösungsraum theoretisch exploriert werden. Die praktische Erprobung der Wirkprinzipien sowie der mit bestimmten Teilfunktionen verbundenen Nutzererfahrung würden dabei zu einem umfassenderen Verständnis des Lösungsraumes beitragen und das Wissen über den Entwicklungsgegenstand vor der Konzeptauswahl vergrößern. Eine wesentliche Herausforderung besteht jedoch darin, das Prinzip der prototypenbasierten Koevolution von Problem- und Lösungsraum, das sich im ME310-Prozessmodell auf die Teilsystemebene beschränkt, mit vertretbarem Aufwand auf die Gesamtsystemebene zu übertragen. Ein möglicher Ansatz könnte hierbei in der frühen Definition modularer Konzeptarchitekturen liegen, die für den Aufbau niedrig aufgelöster Systemprototypen genutzt werden. Im Ergebnis könnte ein neues hybrides Prozessmodell entstehen. Um die Praktikabilität eines solchen Prozessmodells nachzuweisen, ist eine umfassende empirische Analyse erforderlich, die auch den Einfluss verschiedener Entwicklungsobjekte und Entwicklungsorganisationen untersucht.

Literaturverzeichnis

- Brereton, M. & McGarry, B. (2000): An Observational Study of How Objects Support Engineering Design Thinking and Communication: Implications for the design of tangible media. In: Turner, T. & Szwillus, G. (Hrsg.): *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, The Hague, The Netherlands, 01.-06.04.2000, 217–224.
- Brown, T. (2008): Design Thinking. In: *Harvard Business Review*, 86 (6), 84–92.
- Bushnell, T., Steber, S., Matta, A., Cutkosky, M. & Leifer, L. (2013): Using a “Dark Horse” Prototype to manage innovative Teams. In: Fernandes, A. A., Natal Jorge, R. M., Patrício, L. & Medeiros, A. (Hrsg.): *3rd Int. Conf. on Integration of Design, Engineering & Management for Innovation*, Polo, Portugal, 04.-06.09.2013.
- Carlgren, L., Rauth, I. & Elmquist, M. (2016): Framing Design Thinking: The Concept in Idea and Enactment. In: *Creativity and Innovation Management*, 25 (1), 38–57.
- Cooper, R. G. & Sommer, A. F. (2016): The Agile-Stage-Gate Hybrid Model: A Promising New Approach and a New Research Opportunity. In: *Journal of Product Innovation Management*, 33 (5), 513–526.
- Domingo, L., Moore, D., Sirkin, D., Toye, G., Leifer, L. & Cutkosky, M. (2020): Strategic Prototyping to learn in Stanford University’s ME310 Design Innovation Course. In: Marjanović, D., Storga, M., Pavković, N. & Bojčetić, N. (Hrsg.): *DS 102: Proceedings of the DESIGN 2020 16th International Design Conference*, online, 26.-29.10.2020, 1687–1696.
- Dorst, K. & Cross, N. (2001): Creativity in the design process: co-evolution of problem–solution. In: *Design Studies*, 22 (5), 425–437.

- Gericke, K., Beinke, C., Gemmer, P. & Blessing, L. (2010): Entwicklungsmethodik nach Pahl Beitz und Design Thinking. Vergleich und Einordnung. In: Krause, D., Paetzold, K. & Wartack, S. (Hrsg.): *DFX 2010: Proceedings of the 21st Symposium on Design for X*, Buchholz, Hamburg, Deutschland, 23.-24.09.2010, 117–130.
- Haberfellner, R., Weck, O. L. de, Fricke, E. & Vössner, S. (2019): *Systems Engineering, Fundamentals and Applications*. Cham, Switzerland: Springer Nature.
- Kenyon, D., Cutkowsky, M. & Leifer, L. (o.J.): *ME310 ABC Course Reader*. Stanford, California, USA.
- Kopenhagen, F. (2004): *Systematische Ableitung modularer Produktarchitekturen, Komplexitätsreduzierung in der Konzeptphase*. Aachen: Shaker.
- Leifer, L. J. & Steinert, M. (2011): Dancing with ambiguity: Causality behavior, design thinking, and triple-loop-learning. In: *Information Knowledge Systems Management*, 10 (1-4), 151–173.
- Luchs, M. G., Swan, K. S. & Griffin, A. (Hrsg.) (2016): *Design thinking, New product development essentials from the PDMA*. Hoboken, New Jersey: Wiley.
- Pahl, G., Beitz, W., Feldhusen, J. & Grote, K.-H. (2007): *Pahl / Beitz Konstruktionslehre, Grundlagen erfolgreicher Produktentwicklung; Methoden und Anwendung*. 7. Aufl. Berlin: Springer.
- Ponn, J. & Lindemann, U. (2011): *Konzeptentwicklung und Gestaltung technischer Produkte, Systematisch von Anforderungen zu Konzepten und Gestaltlösungen*. 2. Aufl. Berlin: Springer.
- Rittel, H. W. J. & Webber, M. M. (1973): Dilemmas in a General Theory of Planning. In: *Policy Sciences*, 4 (2), 155–169.
- Schüttoff, M., Herrmann, T., Roth, D. & Binz, H. (2019): Analyse und Beurteilung der unterschiedlichen Einsatzzwecke und Anwendungsgrenzen von Design Thinking. In: Binz, H., Bertsche, B., Bauer, W. & Riedel, O. (Hrsg.): *Stuttgarter Symposium für Produktentwicklung SSP 2019, Agilität und kognitives Engineering*, Stuttgart, 16.05.2019, 193–202.
- Sobek II, D. K., Ward, A. C. & Liker, J. K. (1999): Toyota´s Principles of Set-Based Concurrent Engineering. In: *Sloan Management Review*, 40 (2), 67–83.
- Viswanathan, V. K. & Linsey, J. S. (2012): Physical Models and Design Thinking: A Study of Functionality, Novelty and Variety of Ideas. In: *Journal of Mechanical Design*, 134 (9), 1–13.

Kontakt

Prof. Dr.-Ing. Frank Kopenhagen,
 Prof. Dr. Tobias Held
 Hochschule für Angewandte Wissenschaften Hamburg
 Department Maschinenbau und Produktion
 Berliner Tor 21, 20099 Hamburg
www.haw-hamburg.de

Tim Blümel, B. Sc.
 Universität Stuttgart
 Keplerstraße 7
 70049 Stuttgart
www.uni-stuttgart.de

Univ.-Prof. Dr. Dipl.-Ing. Christoph H. Wecht, MBA
 New Design University Privatuniversität St. Pölten
 Mariazeller Straße 97a, 3100 St. Pölten, Österreich
www.ndu.ac.at