

Please quote as: Gürtler, M.; Kortler, S.; Helms, B.; Berkovich, M.; Leimeister, J. M.; Krcmar, H.; Shea, K.; Lindemann, U. & und Maik Maurer (2012): Von Anforderungslisten zum konzeptionellen Design – Funktionsbasierte Analyse von Anforderungen an Product-Service Systems. In: Dienstleistungsmodellierung 2012. 1. Aufl./Vol.. Hrsg./Editors: Thomas, O. & Nüttgens, M. Erscheinungsjahr/Year: 2012. Seiten/Pages: 75-96.

# Von Anforderungslisten zum konzeptionellen Design – Funktionsbasierte Analyse von Anforderungen an Product-Service Systems

Matthias Gürtler, Sebastian Kortler, Bergen Helms, Marina Berkovich, Jan Marco Leimeister, Helmut Krcmar, Kristina Shea, Udo Lindemann und Maik Maurer

*Die Verbreitung sogenannter Product-Service Systems (PSS), welche Sach- und Dienstleistungen in sich vereinen, nimmt kontinuierlich zu. Eine Aufgabe bzw. Anforderung kann hierbei durch eine beliebige Kombination der drei Domänen Produkt, Software und Service realisiert werden. Aufgrund der hohen resultierenden Anzahl an Freiheitsgraden und der Integration unterschiedlichster Disziplinen stellt die Anforderungsanalyse und Modellierung eine Herausforderung dar. Besondere Bedeutung kommt hierbei dem konzeptionellen Design zu, welches das Bindeglied zwischen Anforderungs- und Lösungselementen bildet. Der folgende Beitrag präsentiert ein Verfahren, das einen nahtlosen Übergang von der Anforderungsanalyse ins Design realisiert. Dieses ermöglicht die Erstellung eines Anforderungs- und Funktionsmodells, deren gezielte und homogene Konkretisierung sowie eine kontinuierliche Implementierung neuer oder veränderter Kundenanforderungen. Die Anwendbarkeit und Vorteile des Verfahrens werden an einem Praxisbeispiel verdeutlicht.*

## 1 Einleitung

Sowohl Produkthersteller als auch Dienstleistungsunternehmen sehen sich einem zunehmenden Preis- und Wettbewerbsdruck ausgesetzt. Dieser ist unter anderem ein Resultat neuer, oft internationaler Anbieter auf dem Markt, die durch Ausnutzung regionaler Lohnkostenunterschiede große Kostenvorteile erzielen (Leimeister und Glauner 2008). Zusätzlich wünscht sich eine wachsende Anzahl an Kunden individuelle Komplettlösungen anstelle von standardisierten Einzellösungen. Daneben erweitern Produkthersteller gezielt ihr Portfolio um Dienstleistungen, in der Absicht sich von der Konkurrenz zu differenzieren (Knackstedt et al. 2008; Leimeister 2012). Die Idee ist hierbei eine Wandlung der Unternehmen hin zu sog. „Lösungsanbietern“ (Leimeister und Glauner 2008). Der Fall, dass Produkte ohne konkretes „Problem“ entwickelt werden und erst durch ihre Existenz bzw. Nutzung neue Kundenbedürfnisse wecken, wie dies unter anderem im Sonderfor-

schungsbereich 768 (SFB 768: „Zyklusmanagement von Innovationsprozessen“) betrachtet wird, soll nicht Gegenstand dieses Beitrags sein.

Die abgestimmte Verzahnung von Sach- und Dienstleistung wird in der englischsprachigen Literatur mit Product-Service Systems (PSS) bezeichnet. Daneben existiert speziell in der deutschen Forschung eine Vielzahl alternativer Begriffe, die beispielsweise von Knackstedt et al. (2008) oder Thomas et al. (2008) näher erläutert werden. Aufgrund dieser Begriffsvielfalt im deutschsprachigen Raum wird im Folgenden die englische Bezeichnung PSS verwendet.

Ein PSS wird durch die Integration der drei Domänen Produkt, Software und Dienstleistung gebildet. Diese können prinzipiell beliebig kombiniert werden zur Realisierung einer bestimmten Anforderung oder Aufgabe. Der hieraus resultierende Freiheitsgrad zusammen mit der hohen Anzahl an unterschiedlichen, beteiligten Fachdisziplinen führt zu einer hohen Komplexität in der PSS-Entwicklung. Hierzu trägt zusätzlich auch das vergrößerte Funktionsspektrum bei, welches sich aus dem im Gegensatz zum Ausgangsprodukt vergrößerten Leistungsangebot ergibt. Wie bereits bei der Entwicklung „klassischer“ Produkte ist daher die Anforderungsermittlung und -verwaltung eine wichtige Voraussetzung für den Erfolg eines Produktes (Lindemann 2009). Das Requirements Engineering (RE) stellt hierbei einen kritischen Teil des Entwicklungsprozesses dar, dessen unzureichende Durchführung große Zusatzkosten und Aufwand durch späte Änderungen nach sich ziehen kann (Byrd et al. 1992). Deswegen wird eine integrierte und abgestimmte Analyse von Anforderungen benötigt, welche eine lösungs- bzw. domänenneutrale Anforderungs- und Funktionsmodellierung ermöglicht.

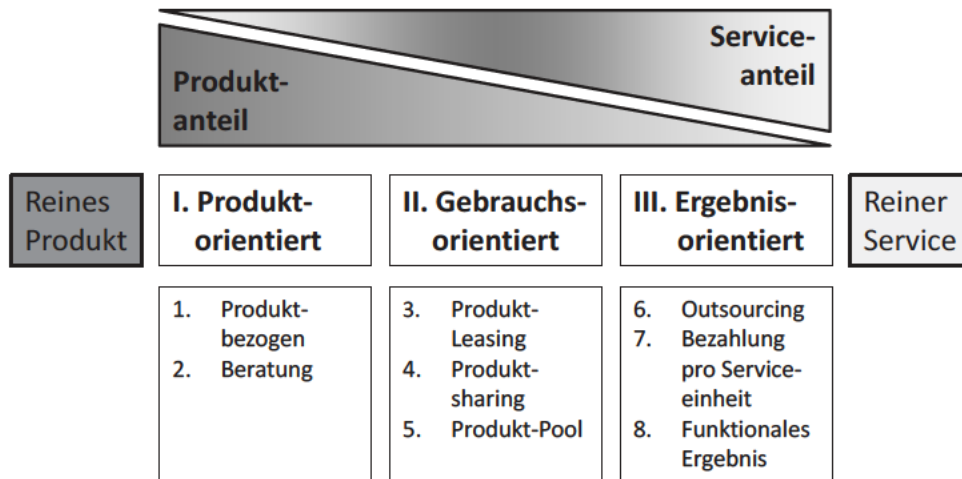
Im Folgenden wird ein matrizenbasierter Ansatz vorgestellt, der von Kortler et al. (2010) entwickelt wurde und der ein effizientes Iterationsmanagement im Rahmen der Konzepterstellung des PSS ermöglicht. Nachgelagerte Entwicklungstätigkeiten werden hierbei nicht betrachtet. Die Anwendbarkeit des Ansatzes wird anschließend anhand eines Beispiels demonstriert.

Die Kernidee des Ansatzes bildet die Erstellung eines möglichst homogengranularen Anforderungs- und Funktionsmodells. Dies unterstützt die anschließende Entwicklung, da einerseits zu abstrakte und dadurch schlecht erfüllbare bzw. messbare Anforderungen und andererseits zu abstrakte und kaum realisierbare Funktionen vermieden werden. Die hohe Homogenität ermöglicht zudem eine effiziente Implementierung neuer oder veränderter Anforderungen sowie die Generierung alternativer Teillösungen basierend auf unterschiedlichen Domänenkombination bzw. -ausprägung.

## **2 Requirements Engineering im Kontext von PSS**

Wie im vorherigen Abschnitt angesprochen, setzen sich PSS aus einer Sach- und einer Dienstleistung zusammen. Entsprechend der Ausprägung der jeweiligen Anteile lassen sich gemäß Tukker (2004) drei PSS-Kategorien und acht verschiedene Arten an PSS unterscheiden, wie in Abb. 1 dargestellt. Hierbei bewegen sich pro-

duktorientierte PSS mit produktbezogenen Services und Produkt-/Benutzungsberatung relativ nahe am klassischen, rein materiellen Produkt. Demgegenüber stehen die ergebnisorientierten PSS, die mehr oder weniger losgelöst von einem materiellen Produkt ein Ergebnis für ein Kundenbedürfnis liefern. Dazwischen bewegen sich die gebrauchorientierten PSS, die zwar auf materiellen Produkten basieren jedoch nicht primär deren Verkauf sondern deren Vermietung oder Leasing beinhalten. Die Eigentumsrechte bleiben hierbei beim jeweiligen PSS-Anbieter (Tukker 2004, Baines et al. 2007).



**Abb. 1.** PSS-Kategorien und Arten (nach Tukker 2004)

Mit dem Erwerb eines PSS erhält der Kunde „ein auf seine Nutzungsanforderungen abgestimmtes und zuverlässig einsetzbares Gesamtsystem“ (Böhmman und Krcmar 2007). Um dieses flexibel an Kundenwünsche anzupassen, können spezifische Funktionen der PSS in Module gekapselt werden, die je nach Bedarf standardisiert oder kundenindividuell gestaltet werden können (Böhmman et al. 2008). Der Erfolg eines Lösungsanbieters hängt davon ab, inwieweit er die Realisierung der einzelnen Funktionen in Abhängigkeit der jeweiligen Kundenwünsche adäquat auf die drei zuvor genannten Domänen oder Kombination von ihnen zu verteilen vermag. Hierbei kommt speziell der Fähigkeit große Bedeutung zu, effizient auf neue oder veränderte Kunden-/Marktwünsche einzugehen, welche nicht nur während der Entwicklung sondern auch während der Nutzungsphase des PSS auftreten können. Dies betrifft auch oder gerade besonders Wünsche, die eine Änderung der realisierenden Domäne erfordern.

Das Requirements Engineering hilft hierbei durch die Erhebung und Verwaltung der Kunden- und Marktwünsche. Diese oft unpräzise formulierten Wünsche müssen in einem ersten Schritt in technisch greifbare Spezifikationen, das heißt in „die Sprache der Entwickler“ überführt werden (INCOSE 2006). Die Spezifikationen dienen einerseits der Entwicklung eines gemeinsamen, disziplinübergreifenden Problemverständnisses und andererseits der späteren Verifikation der im fertigen PSS umgesetzten Anforderungen. Bei einer Vernachlässigung dieses Schrittes

besteht die Gefahr, implizite Anforderungen nicht zu berücksichtigen, was zu einer Beeinträchtigung des PSS oder späten Änderungen führen kann (Pohl 2007).

Eine abgestimmte Komponentenentwicklung eines PSS setzt eine effektive technisch-organisatorische Zusammenarbeit der beteiligten Disziplinen voraus (Böhm und Krömer 2007). Das bedeutet, dass die einzelnen Disziplinen bei der allgemeinen und anschließend domänenspezifischen Konzeption und Entwicklung, wie zuvor erwähnt, ein gemeinsames Problemverständnis gewinnen müssen. Die Grundlage für eine integrierte Entwicklung muss daher bereits in den Anforderungen gelegt werden (Spath und Demuß 2003). Diese gilt es lösungs- und domänenneutral zu analysieren und zu modellieren, um die Generierung neuer, innovativer Lösungen zu unterstützen und einen späten Wechsel im PSS-Lebenszyklus zwischen verschiedenen Lösungsalternativen zu ermöglichen. In der Literatur als auch in der Praxis ist die Bedeutung einer domänenübergreifenden Analyse und Modellierung zwar bekannt, allerdings fehlen konkrete Vorgehensweise für PSS weitgehend (Berkovich et al. 2009).

### **3 Existierende Vorgehensmodelle zur Entwicklung von PSS**

Wie Gräßle et al. (2010) formulieren, stellt die Entwicklung von PSS „kein revolutionäres Entwicklungsparadigma dar, sondern eine evolutionäre Weiterentwicklung“ der klassischen Produktentwicklung und des Service Engineerings. Bereits der traditionelle Entwicklungsprozess ebenso wie das Service Engineering sind gekennzeichnet durch einen hohen Anteil an Kreativität, Vielgestaltigkeit, Beteiligung verschiedener Personen und eine Abhängigkeit von menschlichem Urteilsvermögen, welche maßgeblich zur großen Komplexität des Produktes und des Entwicklungsprozesses beitragen (Gräßle et al. 2010). Um dieser zu begegnen, wurden verschiedene Vorgehensmodelle mit mehr oder weniger konkreten Handlungsanweisungen, wie beispielsweise im Gesundheitswesen der AALSDA-Ansatz (Menschner et al. 2011), für die Produktentwicklung die VDI-Richtlinie 2221 oder das Münchner-Vorgehens-Modell erarbeitet (Lindemann 2009). Diese Vorgehensmodelle lassen sich jedoch nur bedingt auf die speziellen Eigenschaften und Entwicklungsanforderungen von PSS übertragen.

Essentiell für ein Vorgehensmodell zur Entwicklung von PSS sind vor allem, wie zuvor und ebenso bei Gräßle et al. (2010) dargestellt:

- die gleichberechtigte Behandlung von Sach- und Dienstleistung unter Vermeidung einer zu frühen Festlegung einzelner Entwicklungsaspekte auf bestimmte Domänen,
- die Integration des Kunden in den Entwicklungsprozess, um die PSS-Akzeptanz bei diesem zu erhöhen,
- die Ableitung technischer Spezifikationen aus den Kundenwünschen und Verteilung auf verschiedene wissenschaftliche Disziplinen und

- die Möglichkeit, dass Entwicklung und Erbringung eines PSS durch ein oder auch mehrere Unternehmen erfolgen kann, wie es auch im SFB 768 untersucht wird.

Thomas et al. (2008) und speziell Gräßle et al. (2010) entwickeln einen Vergleichsrahmen basierend auf unterschiedlichen Bewertungsmerkmalen, um verschiedene existierende PSS-Engineering (PSSE) Vorgehensmodelle zu vergleichen und zu bewerten. Hierbei kommen sie zu dem Ergebnis, dass die betrachteten Vorgehensmodelle noch einen hohen Grad an Verbesserungspotenzial aufweisen, der sich vor allem durch eine „zumeist fehlende Evaluierung und unzureichende Detaillierung“ äußert.

Weber et al. (2004) entwickelten aufbauend auf dem Konzept des Property-Driven Development/Design (PDD) ein Schalenmodell des PSSE. Ein wichtiger Aspekt des Modells ist die auf Andreasen et al. (1995) zurückgehende Unterscheidung zwischen Eigenschaften und Merkmalen eines Produktes/PSS. Eigenschaften stellen hierbei sog. „äußere Eigenschaften“ dar, welche das Produktverhalten beschreiben, vom Kunden festgelegt und vom Entwickler nur indirekt beeinflusst werden können. Merkmale hingegen stellen sog. „innere Eigenschaften“ dar, beinhalten beispielsweise die Struktur, Geometrie und Material eines PSS und können direkt vom Entwickler beeinflusst werden (Weber et al. 2004). Die Idee hierbei ist, dass der Kunde bestimmte Wünsche an das Produkt besitzt, jedoch an den Details der technischen Umsetzung (in den meisten Fällen) nicht interessiert ist. Das Schalenmodell selbst beschreibt einen iterativen Prozess, der ausgehend von initialen Kundenwünschen in einem Synthese-Schritt PSS-Merkmale entwickelt. Aus diesen werden in einem anschließenden Analyse-Schritt resultierende PSS-Eigenschaften abgeleitet. Diese Ist-Eigenschaften werden kontinuierlich mit den aktualisierten Kundenwünschen verglichen, die den Soll-Zustand des PSS verkörpern. Auf Basis des ermittelten Deltas zwischen Ist- und Soll-Eigenschaften wird der weitere Entwicklungsbedarf ermittelt und die PSS-Merkmale angepasst.

Ähnlich verhält es sich mit dem von Thomas et al. (2008) vorgestellten Verfahren, welches auf dem Schalenmodell von Weber basiert.

Beiden Vorgehensmodellen ist ihr relativ hohes Abstraktionsniveau gemein, welches so gut wie keine konkreten Arbeitsanweisungen oder Hilfsmittel nennt.

In dieser Hinsicht zeichnet sich das umfangreiche Vorgehensmodell zur Entwicklung von Industriellen Produkt-Service Systemen (IPS<sup>2</sup>) aus, welches von Sadek (2009) in seiner Dissertation präsentiert wurde. Diese sog. „heterogene IPS<sup>2</sup>-Konzeptmodellierung“ betrachtet hierbei die integrierte Entwicklung von Sach- und Dienstleistungen mit besonderem Fokus auf der Dynamik des Entwicklungsprozesses.

Im Gegensatz zu anderen Vorgehensmodellen liegt der Fokus des in diesem Beitrag vorgestellten Verfahrens auf der Phase der Konzepterstellung des PSS mit der Erstellung eines Anforderungs- und Funktionsmodells. Vor- oder nachgelagerte Phasen des Produktlebenszyklus, wie die Erhebung der Kundenwünsche oder die letztendliche Entwicklung des PSS, werden bewusst ausgeklammert. Ähnlich

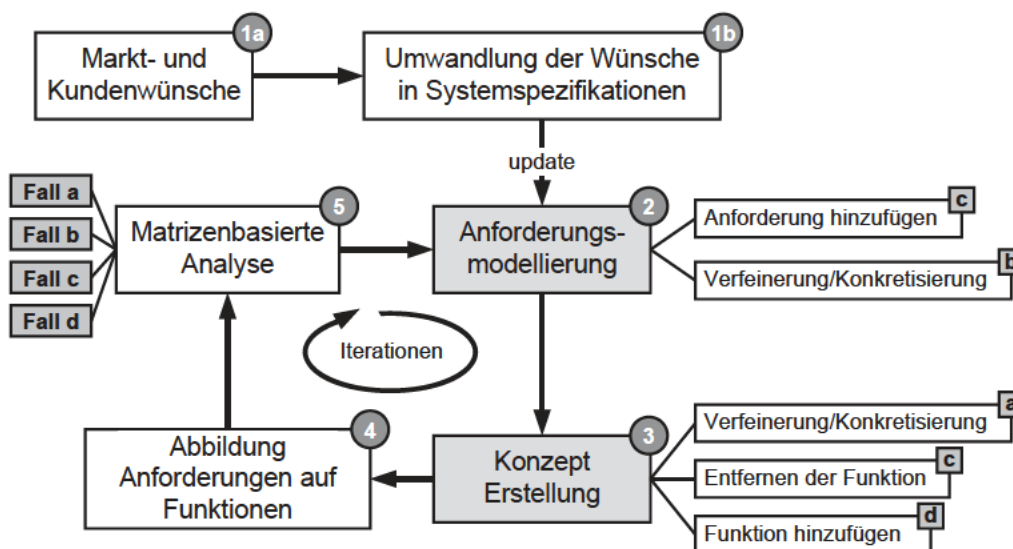


wie bei Weber (2004) oder Sadek (2009) wird die große Wichtigkeit der sich verändernden Kundenwünsche durch deren effiziente Implementierung berücksichtigt. Hierbei ist das präsentierte Verfahren durch den Einsatz von Matrixmethoden gegenüber den anderen Vorgehensmodellen charakterisiert, welche ein effizientes Interaktions-Management im Zuge der PSS-Architekturerstellung erlauben.

#### 4 Iterative Anforderungs- und Funktionsverfeinerung mit Hilfe von Matrix-Methoden

Das von Kortler et al. (2010) entwickelte Verfahren basiert auf der von Danilovic und Browning (2004, 2007) erarbeiteten Domain Mapping Matrix (DMM), welche eine Weiterentwicklung der von Steward (1981) vorgestellten Design Structure Matrix (DSM) darstellt. Während sich mit einer DSM Abhängigkeiten zwischen Elementen einer Domäne darstellen lassen, können mit einer DMM Beziehungen zwischen Elementen einer Domäne und Elementen einer anderen Domäne abgebildet werden, in diesem Fall zwischen Anforderungen und Funktionen.

Dem Verfahren liegt das Prinzip der Analyse von Anforderungen und Funktionen zugrunde, welches eine iterative und wechselseitige Konkretisierung von Anforderungen und eine damit in Verbindung stehende Verfeinerung von Funktionen realisiert. Durch die Funktionen wird die Verbindung zwischen Anforderungen und Lösungselementen hergestellt, wodurch eine nahtlose Integration der Anforderungsanalyse in die Entwicklung stattfindet. Mittels des vorgestellten Verfahrens lässt sich dieser iterative Prozess effizienter durchlaufen und eine homogene Granularität von Funktionen und Anforderungen erreichen.



**Abb. 2.** Prozessmodell der wechselseitigen Anforderungs- und Funktionsverfeinerung (nach Kortler et al. 2010)

Das in Abb. 2 dargestellte Vorgehensmodell umfasst fünf iterative Prozessschritte, die für jede Iterationsschleife durchlaufen werden.

*Schritt 1:* Im ersten Schritt werden die Markt- und Kundenwünsche gewonnen (1a) und anschließend in technisch greifbare Spezifikationen bzw. technische Anforderungen transformiert (1b). Hierbei ist es möglich, dass ein einzelner Markt- oder Kundenwunsch in ein oder mehrere Anforderungen aufgespalten wird. Während in der ersten Iterationsschleife die initialen Markt- und Kundenwünsche erfasst werden, werden in allen folgenden Iterationsschleifen auf diesem Weg neue oder veränderte Wünsche berücksichtigt.

*Schritt 2:* Im Rahmen der Anforderungsmodellierung werden die zu Beginn der ersten Iterationsschleife unstrukturiert vorliegenden Anforderungen gruppiert, kategorisiert und hierarchisiert (Ponn und Lindemann 2008). In den folgenden Iterationsschleifen werden auf Basis des Modells bestehende Anforderungen verfeinert oder um neue ergänzt. Zudem ist es möglich, dass sich an dieser Stelle aufgrund veränderter Kundenwünsche veränderte oder komplett neue Anforderungen ergeben oder bestehende wegfällen. Hierbei ist es essentiell zu beachten, dass eine Veränderung oder Streichung einer Anforderung nicht eigenständig/autonom vom Entwickler durchgeführt werden darf, da diese einen konkreten Markt- oder Kundenwunsch repräsentiert.

*Schritt 3:* Im Zuge der anschließenden Konzept-Erstellung wird ein Funktionsmodell erstellt, welches zur Erfüllung des Anforderungsmodells des zweiten Schrittes dient. Die Struktur des Funktionsmodells stellt hierbei eine logische Architektur dar, die definiert, was das zu entwickelnde System zu leisten hat und wie Eingabeinformationen in das System in Ausgabeinformationen unter Verwendung von Kontrollflussinformationen transformiert werden. In diesem Fall wird speziell die relationale Funktionsstruktur betrachtet. Diese stellt analog dem Anforderungsmodell funktionale Zusammenhänge in einer geclusterten Hierarchie von Funktionen dar (Ponn und Lindemann 2008). Hierbei wird die Gesamtfunktion des zu entwickelnden PSS in Teilfunktionen zerlegt, die es erlauben, einzelne Teilausschnitte des Systems in funktionaler Hinsicht zu analysieren und in späteren Entwicklungsschritten Teillösungen abzuleiten. Neben der hierarchischen Beziehung können noch andere Relationen zwischen einzelnen Funktionen bestehen, beispielsweise kausale Beziehungen. Diese geben an, welche anderen Funktionen eine Funktion zur Durchführung benötigt bzw. welche anderen Funktionen von ihr abhängen. Im Rahmen des Verfeinerungsprozesses werden in jeder Iterationsschleife bestehende Funktionen in Teilfunktionen aufgespalten, neue Funktionen hinzugefügt oder nicht benötigte Funktionen entfernt.

*Schritt 4:* Dieser Schritt beinhaltet die matrixenbasierte Abbildung von Anforderungen auf die sie erfüllenden Funktionen. Die Zuordnung erfolgt basierend auf der Erfahrung der Entwickler, durch Experten-Interviews oder Workshops (Lindemann 2009). Die Funktionen werden hierbei in der DMM zeilenweise und die Anforderungen spaltenweise aufgetragen. Ein Matrixeintrag bedeutet, dass eine Beziehung zwischen den beiden, jeweils betrachteten Elementen besteht.



		Anforderungen									
		A1	A2	A3	A4	A5	A6	A7	A8	A9	Aktiv-Σ
Funktionen	erfüllt ↑										
	F1	1		1	1		1		1	1	6
	F2				1				1		2
	F3	1		1							2
	F4	1	1					1			3
	F5										0
	F6	1								1	2
F7	1					1				2	
	Passiv-Σ	5	1	2	2	0	2	1	2	2	

a: Verfeinerung von F1  
 b: Verfeinerung von R4  
 c: Unnötige Funktion F5  
 d: Fehlende Funktion F8

**Abb. 3.** Abbildung von Funktionen auf Anforderungen (nach Kortler et al. 2010)

*Schritt 5:* Der fünfte Schritt umfasst die Analyse mit matrixbasierten Methoden, aus deren Ergebnissen sich der weitere Konkretisierungsbedarf ableiten lässt. Wie in Abb. 3 dargestellt, erfolgt die Bewertung mittels Metriken. In diesem Fall wird für jede Funktion ihre zeilenweise Aktivsumme gebildet, welche die Anzahl der durch sie erfüllten Anforderungen angibt. Analog verhält es sich mit der spaltenweisen Passivsumme, die ein Maß für die Menge an Funktionen ist, welche die jeweilige Anforderung erfüllen. Als Analyseergebnis ergeben sich folgende vier Fälle:

1. *Funktionen mit einer hohen Aktivsumme* weisen auf einen Verfeinerungsbedarf hin. Dem liegt die Annahme zugrunde, dass je mehr Anforderungen eine Funktion erfüllt, desto abstrakter sie formuliert ist.
2. *Anforderungen mit einer hohen Passivsumme* liefern ein Indiz für eine Konkretisierung der betreffenden Anforderung, da sie aufgrund ihrer abstrakten Formulierung eine große Anzahl an erfüllenden Funktionen benötigt.
3. *Funktionen mit einer Aktivsumme gleich 0* erfüllen keine einzige Anforderung. Daher ist zu prüfen, ob die Funktion nicht zu streichen ist oder ob ihre Existenz auf eine vergessene, implizite Anforderung hinweist, die in das Anforderungsmodell hinzu zufügen ist.
4. *Anforderungen mit einer Passivsumme gleich 0* geben an, dass sie bislang durch keine Funktion realisiert werden und dementsprechend neue Funktionen zu erstellen und in das Funktionsmodell zu integrieren sind.

Zusammenfassend lässt sich festhalten, dass die vorgestellte Matrix-Methode eine effiziente Durchführung der Modellverfeinerung unterstützt und ebenfalls die Einbindung neuer oder veränderter Markt- und Kundenwünsche zu jeder Zeit des Prozesses ermöglicht. Die DMM-Analyse deckt hierbei zuverlässig zu abstrakte oder fehlende Anforderungen und Funktionen auf. Das Ergebnis der Iterations-

schleifen bildet eine homogen-granulare Anforderungs- und Funktionsstruktur, die die Ausgangsbasis für alle folgenden Entwicklungsphasen bildet.

Im Folgenden soll die Anwendbarkeit der vorgestellten Matrix-Methode am Praxisbeispiel eines entwickelten PSS veranschaulicht werden.

## 5 Vorstellung des Praxisbeispiels

Im Rahmen eines Industrieprojekts wurde die Entwicklung und Markteinführung des im Folgenden vorgestellten PSS „Mobiler Ruheraum“ untersucht.

Den Ausgangspunkt für die Entwicklung des PSS bildet das initiale Kundenbedürfnis nach einer Ruhe- bzw. Erholungsmöglichkeit in einem öffentlichen Raum, wie beispielsweise einem Bahnhof, Flughafen oder ähnlichem. Wie in Abb. 4 dargestellt, wendet sich der Nutzer oder eine Nutzergruppe mit genanntem Wunsch an den betrachteten Lösungsanbieter. Dieser bietet dem Nutzer eine Lösung für sein Problem/Wunsch, indem er einen mobilen Ruheraum entwickelt und konstruiert. Zugleich entwirft er auch unterstützende Dienstleistungen wie Reinigung, Wartung, usw. Der Nutzer hat schließlich die Möglichkeit, den Ruheraum mittels eines Buchungssystems zu reservieren und auf diese Weise die gewünschte Erholung zu erhalten.

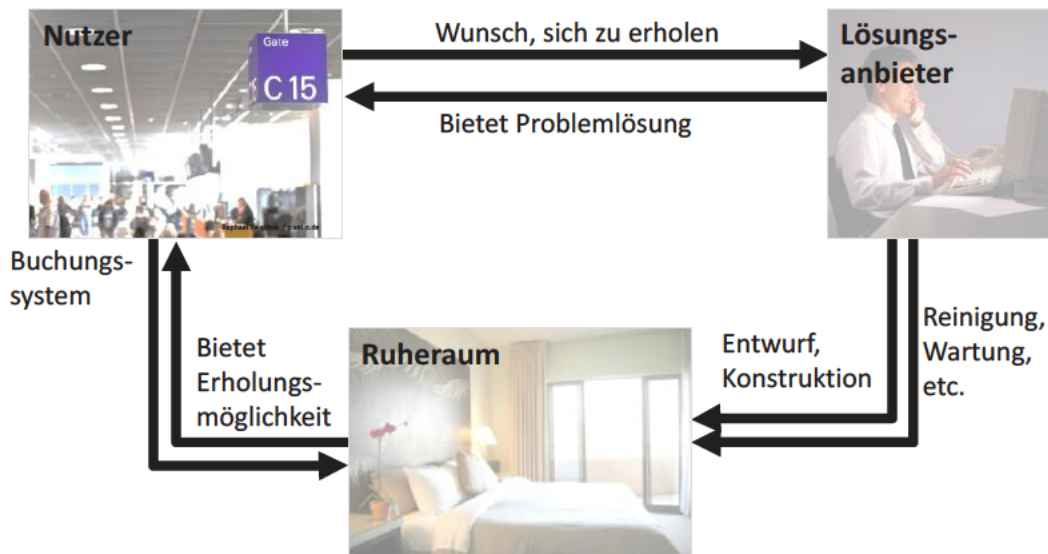


Abb. 4. PSS „Mobiler Ruheraum“

Entsprechend Abb. 2 musste zur Entwicklung des PSS-Konzepts das initiale und sehr abstrakte Kundenbedürfnis auf verschiedene Markt- und Kundenwünsche herunter gebrochen werden, die von verschiedenen Stakeholdern, Mitarbeitern, aus gesetzlichen Vorschriften, Umweltschutzaspekten, usw. stammten. Diese wurden anschließend in technische Anforderungen spezifiziert und in der ersten Iterationsschleife in weitere untergeordnete Teilanforderungen aufgespalten. Das Ergebnis war eine vorläufige Anforderungsliste, wie in Abb. 5 gezeigt, welche

noch relativ grobe Anforderungen, wie „311: Der Gast soll die Kabine im Vorfeld reservieren können“ oder „371: Dem Gast sollen verschiedenen Erfrischungsgetränke zur Verfügung stehen“ beinhaltet.

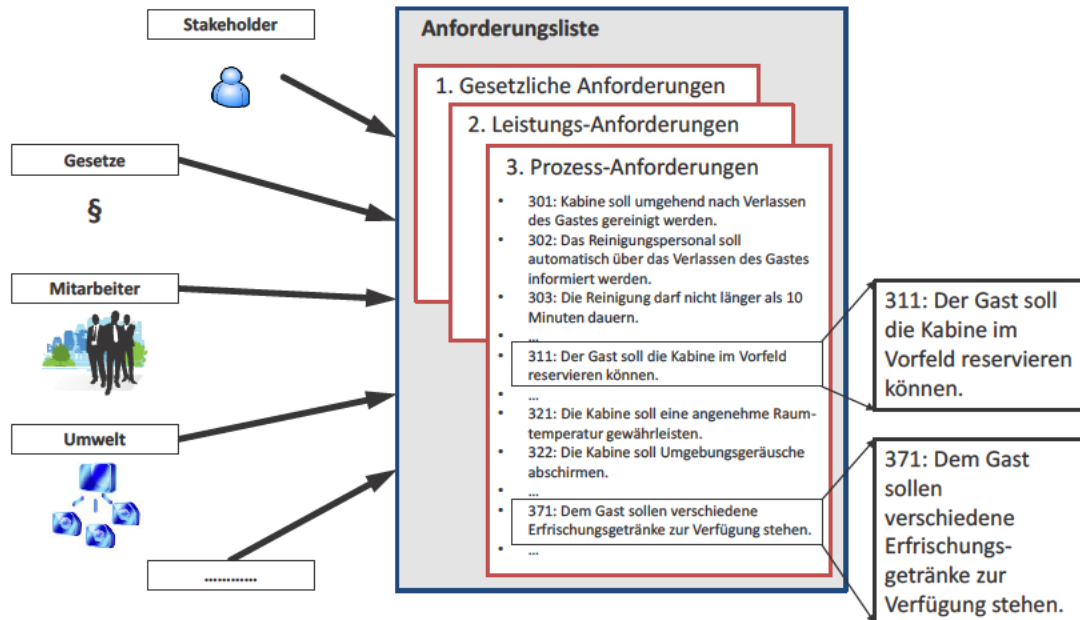


Abb. 5. Vereinfachte Anforderungsliste der ersten Iterationsschleife (nach Kortler et al. 2010)

Basierend auf dieser Anforderungsliste wurde eine erste Funktionsstruktur des PSS generiert. Wie in Abb. 6 verdeutlicht, lässt sich diese in verschiedene Kategorien clustern, in diesem Fall das Buchungssystem, die eigentliche Kabine, eine Reinigungs-/Service-Komponente usw. Die Pfeile geben hierbei eine logische Abhängigkeit zwischen den einzelnen Funktionen an: beispielsweise erfordert die Funktion „Kabine reinigen“ eine vorangehende „Reinigung/Service informieren“-Funktion. Gleichzeitig ist „Kabine reinigen“ der Funktion „Bettwäsche wechseln“ prozesstechnisch vorangestellt und bildet mit ihr die Voraussetzung für „Sauberen Schlafplatz bereitstellen“.

Analog zu Abb. 3 wurden im Anschluss Anforderungs- und Funktionsmodell in einer DMM aufeinander abgebildet, mittels Aktiv-/Passivsumme analysiert und der notwendige Verfeinerungsbedarf ermittelt. So kann unter anderem F12 „Buchung verwalten“ in fünf Teilfunktionen, wie F12a „Wunsch mit freien Kapazitäten abgleichen“, F12b „Rückmeldung an Kunden geben“, F12c „Buchung speichern“ usw. aufgespalten werden.

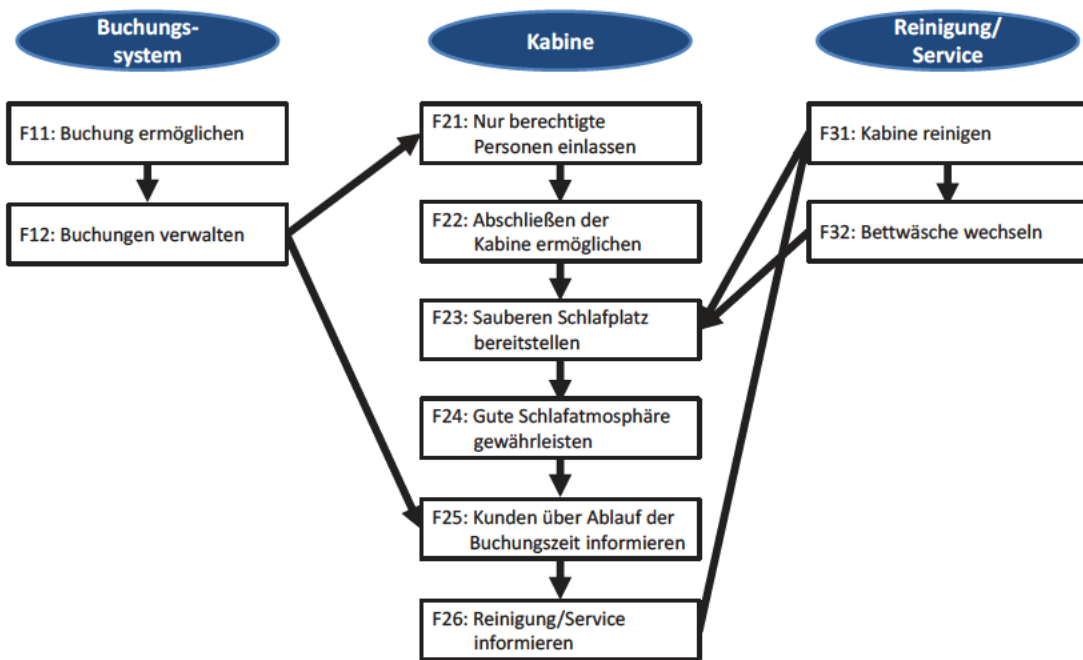


Abb. 6. Vereinfachte Funktionsstruktur der ersten Iterationsschleife

Anschließend wurden die verfeinerten Anforderungs- und Funktionsmodelle wiederum in einer DMM auf einander gemappt und analysiert. Die Iterationsschleifen wurden solange durchlaufen, bis eine homogene fein-granulare Struktur beider Modelle vorlag.

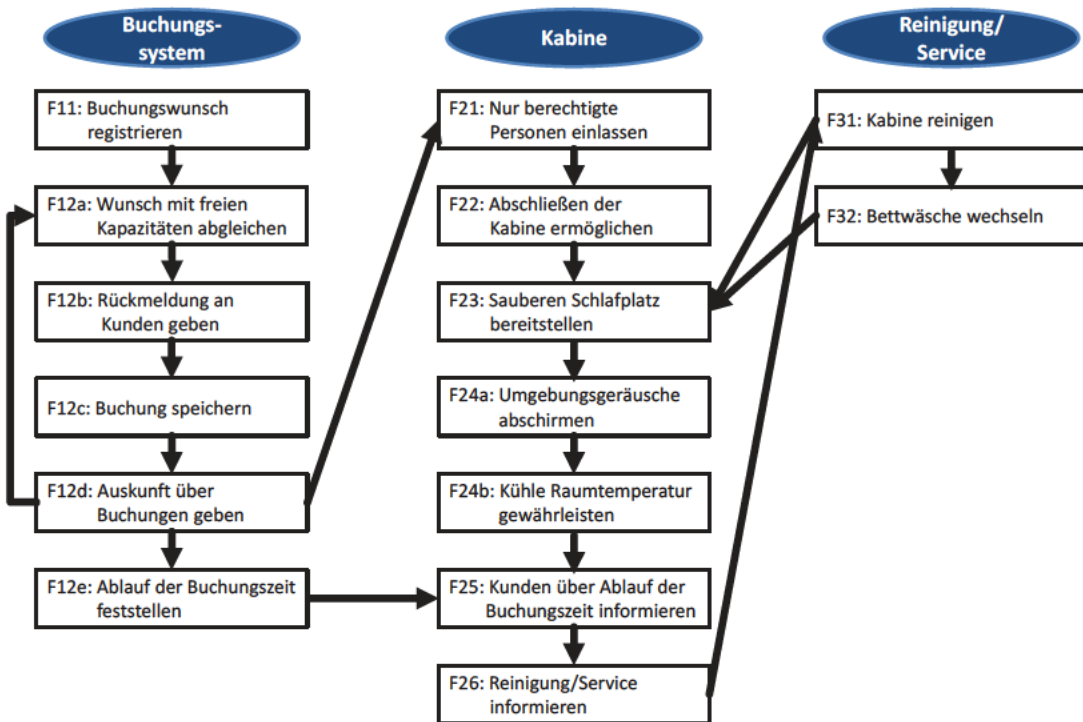


Abb. 7. Verfeinerte Funktionsstruktur der zweiten Iterationsschleife

Die folgenden Schritte der Produktentwicklung werden an dieser Stelle nicht näher betrachtet. Während der ersten Nutzungsphase des neu entwickelten PSS traten verschiedene Ereignisse auf, die einen Rückgriff bzw. Überarbeitung des PSS-Konzepts erforderten und auf die im Folgenden näher eingegangen werden soll. Im Einzelnen waren dies:

- Vergessene und veränderte Kundenanforderungen
- Neue Stakeholder mit neuen Wünschen
- Die Wiederverwendung gewonnener Prozess- und Lösungsinformationen für die Entwicklung eines anderen, unabhängigen PSS.

## **6 Implementierung neuer, vergessener oder veränderter Kundenwünsche nach Markteinführung**

Nach den ersten Wochen der Nutzung des neuen PSS zeigten sich verschiedene übersehene bzw. implizite Anforderungen, die nachträglich sowohl für das Nachfolgermodell des PSS als auch für das aktuelle PSS berücksichtigt werden mussten.

Einerseits wurde von Seiten des Besitzers der bereit gestellten Aufstellfläche eine leichte Beweglichkeit des Ruheraumes gefordert, um seine verfügbare Gesamtfläche effektiver und flexibler nutzen zu können. Dies war bis dahin nicht explizit umgesetzt worden.

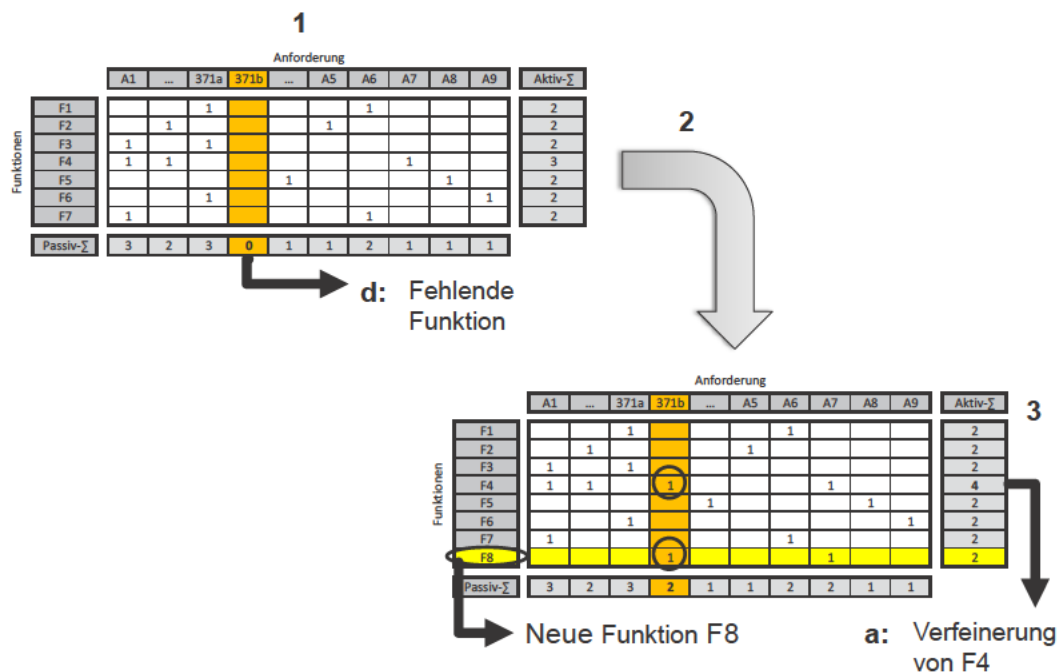
Andererseits äußerte die Mehrheit der Nutzer den Wunsch nach einer veränderten Bereitstellung der Erfrischungsgetränke. Anstelle der Möglichkeit, die Getränke an einem vor der Kabine aufgestellten Automaten eines Dritt-Unternehmens zu erwerben, wünschten sich die Nutzer die Getränke direkt neben dem Ruheplatz innerhalb der Kabine.

In letzterem Fall bedeutete dies eine von außen implizierte (in Abgrenzung zur DMM-Analyse) Ergänzung des bestehenden Anforderungsmodells. Wie Abb. 8 veranschaulicht, wurde die ursprüngliche Anforderung „371: Dem Gast sollen verschiedene Erfrischungsgetränke zur Verfügung stehen“ um die zusätzliche Anforderung ergänzt: „371b: Die Getränke sollen innerhalb der Kabine, neben dem Ruheplatz bereitgestellt werden“. Aus der DMM wird der ersichtlich, dass nach dem Einfügen der neuen Anforderung deren Passivsumme gleich 0 ist, da ihr bislang keine Funktion zugeordnet wurde. Als Konsequenz können nun sowohl bestehende Funktionen (F4) zur Erfüllung der Anforderung genutzt werden, als auch neue Funktionen (F8) erstellt werden. Die Zuordnung von Funktionen führt zu einer Veränderung der jeweiligen Aktivsummen. Im dargestellten Fall bedeutet dies, dass F4 schließlich eine vergleichsweise hohe Anzahl an Anforderungen erfüllt und gegebenenfalls verfeinert werden sollte.

Das Ergebnis im Rahmen des Fallbeispiels war eine Umstellung des vor der Kabine aufgestellten Getränkeautomaten hin zur Bereitstellung der Getränke in der Kabine durch das Reinigungs-/Service-Personal. Dies bedeutete einen Wechsel



der realisierenden Domäne, da von einer technischen zu einer dienstleistungs-basierten Teillösung gewechselt wurde.



**Abb. 8.** Implementierung neuer Anforderungen

## 7 Neue Stakeholder mit neuen Wünschen

Als Folge der zuvor erwähnten, nachgerüsteten Beweglichkeit des Ruheraumes kamen unerwartet verschiedene, neue Stakeholder mit zusätzlichen Wünschen zu den bereits existierenden hinzu. Unter einem Stakeholder wird hierbei jede Person oder Personengruppe verstanden, die mit dem PSS interagiert oder in irgendeiner Art und Weise Interesse und Wünsche an dieses hat.

Unter anderem waren dies einerseits umliegende Ladenbesitzer, die eine Lärmbelästigung und Abschreckung von Kunden durch die relativ laute Kabinenlüftung befürchteten, ebenso wie eine Verdeckung ihrer Läden durch die Kabine und eine hierdurch sinkende Kundenzahl.

Andererseits befürchteten nahegelegene Hotels eine Konkurrenz durch eine zu nahe bei ihnen platzierte Kabine, da Kunden möglicherweise den Ruheraum für eine komplette Nacht buchen könnten anstelle eines Hotelzimmers.

In beiden genannten Fällen wurden unterschiedliche Anforderungen abgeleitet. Im Falle der Läden waren dies unter anderem: Die auftretende Lärmbelästigung durch die Kabinenlüftung soll geringer als 35 dB sein, auf verdeckte Läden soll aufmerksam gemacht werden.

Für die befürchtete Konkurrenzsituation ergab sich: Die maximale Nutzungsdauer soll auf zwei Stunden begrenzt werden.

Die Implementierung der neuen Anforderungen in das bestehende Anforderungsmodell und die Ableitung der resultierenden Funktionsmodelländerungen erfolgen analog zum zuvor beschriebenen Vorgehen.

## **8 Effiziente Wiederverwendung von Prozess- und Lösungsinformation**

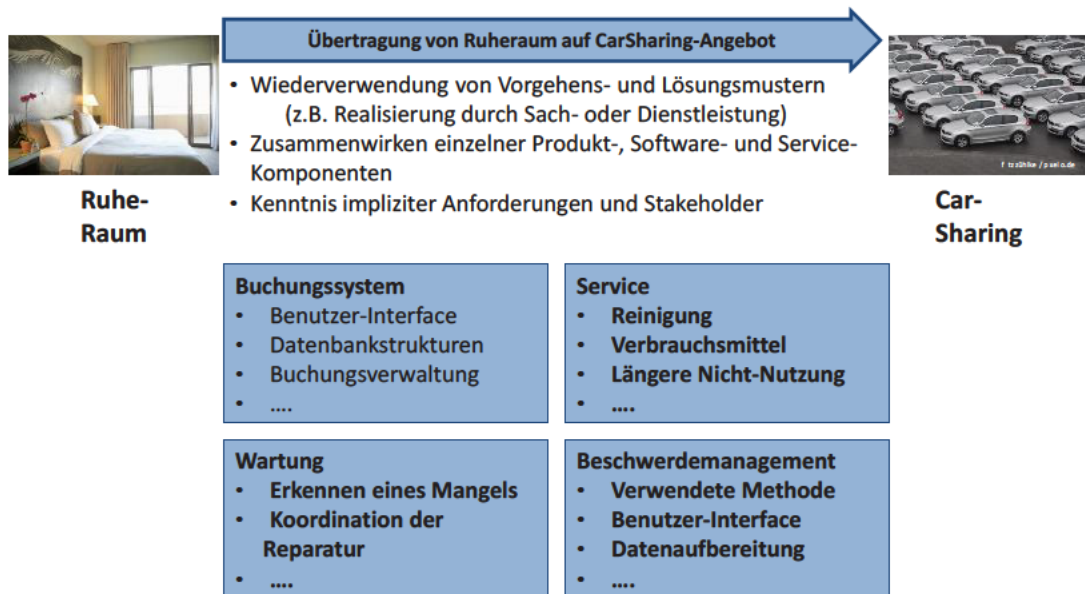
Die Neuentwicklung eines Produkts, speziell eines PSS, ist verbunden mit einem großen Zeit- und Ressourcenaufwand. So sind beispielsweise mehrere Iterationen notwendig, um eine möglichst vollständige Liste an Anforderungen zu erhalten, welche sich zudem aufgrund verändernder Kundenwünsche stetig wandelt. Zusätzlich können ebenfalls neue Stakeholder mit komplett neuen und unterschiedlichen Anforderungen Einfluss auf das Entwicklungsprojekt nehmen. Auch die Erstellung einer homogenen, fein-granularen Anforderungs- und Funktionsstruktur erfordert eine gewisse Anzahl an Iterationsschleifen, wie zuvor beschrieben wurde.

Daher ist zu prüfen, inwieweit während der Entwicklung eines PSS gewonnene Prozess-, Konzept- und Lösungsinformationen im Rahmen der Entwicklung eines anderen PSS genutzt bzw. wiederverwendet werden können. Die Prozessinformation beinhaltet in diesem Kontext das angewandte Vorgehen zur Konzeptverfeinerung, zur Aufdeckung impliziter Anforderungen und Stakeholder, und zur Lösungsfindung. Im Rahmen der Konzeptinformation gilt es zu analysieren, in wie weit sich diese für beide PSS hinsichtlich der Anforderungen decken und welche verknüpften Funktionen übernommen werden können. Entgegen der am Anfang des Beitrags erwähnten Lösungsneutralität zu Beginn einer PSS-Entwicklung kann im Hinblick auf einen effizienten, praxisnahen Entwicklungsprozess unter Umständen für einzelne Teillösungen die Festlegung der realisierenden Domänen (Produkt, Software, Dienstleistung) und bereits existierende Lösungsalternativen mit eventuell ebenfalls vorhandenen Bewertungsergebnissen übernommen werden.

Im konkreten Fall ging es um die Überprüfung der Übertragbarkeit der im Rahmen der Entwicklung des PSS „Mobiler Ruheraum“ gewonnenen Information auf ein neu zu entwickelndes Car-Sharing-Angebot, wie Abb. 9 veranschaulicht. Nach eingehender Analyse ließ sich unter anderem für die dargestellten Bereiche „Buchungssystem“, „Service“, „Wartung“ und „Beschwerdemanagement“ ein hohes Potenzial für einen Transfer feststellen.

Im Falle des Buchungssystems können sowohl das Benutzer-Interface als auch die dahinterliegende Datenbankstruktur mit der auf ihr aufbauenden Buchungsverwaltung und andere Elemente übernommen werden.

Zur Gestaltung der Service-Komponente können organisatorische Elemente der Reinigung, Auffüllung der jeweiligen Verbrauchsmittel als auch Maßnahmen im Falle einer längeren Nichtnutzung des Ruheraums (z. B. Bett, Getränke) bzw. des Autos (z. B. Batterie) adaptiert werden.



**Abb. 9.** Übertragung von Prozess-, Konzept- und Lösungsinformation zwischen PSS

Die Wartung beinhaltet Aspekte, wie die Erkennung einer Fehlfunktion, die anschließende effektive Koordination der Reparatur, usw.

Das Beschwerdemanagement umfasst beispielsweise die verwendete Methode zur Erfassung der Beschwerden, das Nutzer-Interface im Falle einer webbasierten Eingabeplattform und die anschließende Datenaufbereitung für alle betroffenen Abteilungen im Unternehmen.

## 9 Zusammenfassung und Ausblick

Die präsentierte Matrix-Methode unterstützt ein effizientes Zyklenmanagement im Rahmen der Konzept- bzw. Architekturerstellung. Letztere ist ein bedeutender Bestandteil einer PSS-Entwicklung, da diese die Schnittstelle zwischen Anforderungs- und Lösungselementen darstellt. Die essentielle Bedeutung der Anforderungen beruht auf ihrer Ableitung aus Markt- und Kundenwünschen, deren Erfüllung über Erfolg und Misserfolg des PSS entscheiden. Ein effektiver Umgang mit ihnen gestaltet sich oft herausfordernd, da sie kontinuierlichen Veränderungen ausgesetzt sind, die besonders in späten Phasen der Entwicklung einen hohen Änderungsaufwand mit sich bringen können. An dieser Stelle bietet die vorgestellte Matrix-Methode großes Unterstützungspotenzial. Sie erlaubt die Verknüpfung von Anforderungs- und Funktionsmodell auf Basis einer DMM, welche eine schnelle Analyse mittels Aktiv- und Passivsumme ermöglicht. Auf diese Weise können sowohl der notwendige Verfeinerungsbedarf der Modellierung als auch fehlende Funktionen und übersehene/implizite Anforderungen zuverlässig identifiziert werden. Letztere lassen sich ebenso wie neue oder veränderte Kundenanforderungen problemlos in das bestehende Anforderungsmodell integrieren und zeigen den Be-

darf an neu benötigten oder zu verfeinernden Funktionen auf. Das Ergebnis ist eine homogene, fein-granulare Modellstruktur. Die feine Granularität bewirkt hierbei eine Minimierung sowohl zu abstrakter Anforderungen, deren Erfüllung nur schlecht kontrollierbar ist, als auch zu abstrakter Funktionen, die nur mit großem Aufwand in eine realisierbare Lösung umgesetzt werden können. Die Homogenität der Modellstruktur bedeutet einen einheitlichen Abstraktionslevel, der die Entwicklung des PSS unterstützt, da er unter anderem die Ableitung alternativer Teillösungen ermöglicht. Dies ist umso bedeutender, je unterschiedlicher die Ausprägungen der jeweils an der Realisierung beteiligten Domänen (Produkt, Software, Dienstleistung) sind.

Aufgrund der einfachen Dokumentierbarkeit der iterativen, DMM-basierten Konzeptverfeinerung liegen nach Abschluss des Entwicklungsprojektes verschiedene Informationen vor, wie beispielsweise Lösungs-, Konzept- und Prozessinformationen. Letztere beschreiben den Ablauf der durchgeführten Verfeinerungen und zeigen prozesstechnische Verbesserungspotenziale der einzelnen Iterationsschritte auf. Zugleich liefern die gesammelten Informationen Hinweise auf implizite Anforderungen und mögliche Stakeholder, die sich zu Beginn eines Projektes nicht im Fokus befinden, jedoch später hinzukommen und ihre unter Umständen komplett eigenen Wünsche mit einbringen. Im Sinne eines effizienten und praxistauglichen Entwicklungsvorgehens sollten diese Informationen effizient für die Entwicklung folgender, neuer PSS genutzt werden. Neben den zuvor genannten Prozessinformationen können auch – entgegen des eingangs genannten Prinzips der Lösungsneutralität – die Festlegungen einzelner Teillösungen auf bestimmte Domänen und Domänenkombinationen übernommen werden. Dies schränkt zwar den Lösungsraum ein, beschleunigt jedoch den PSS-Entwurf durch die Verwendung einer bewährten Teillösung. Dies ist in der Augen des Autoren insofern gerechtfertigt, da in der Praxis im Gegensatz zur Forschung die Entwicklung eines Produktes/PSS nicht auf der „grünen Wiese“ sondern ausgehend von einem bestehenden Produkt-/Leistungsportfolio beginnt.

Da sich die vorgestellte Matrix-Methode bislang auf die Erstellung der PSS-Architektur beschränkt, gilt es im nächsten Forschungsschritt zusätzlich auch nachgelagerte Entwicklungsschritte und die Einschleusung von veränderten bzw. neuen Anforderungen über alle Ebenen des Lösungsraumes in die Betrachtung mit einzubeziehen. Hierbei ist wie ebenfalls in der Konzeptphase eine effiziente Ermittlung und Behandlung von Zielkonflikten speziell zwischen den einzelnen PSS-Domänen von Bedeutung. In enger Verbindung hierzu steht ein effektives Management von Varianten, deren Teillösungen auf verschiedenen Kombinationen der drei Domänen basieren.

Der Ansatz der Product-Service Systems bietet großes Potenzial, unter anderem aufgrund der gegen über einer Sachleistung erhöhten Flexibilität des Dienstleistungsparts. Diese ermöglicht eine schnelle Reaktion auf späte, auch während der Nutzungsphase auftretende Markt- und Kundenwünsche. Jedoch ist es essentiell, seinen Mehrwert gegenüber rein dienstleistungs- und rein sachleistungsbasierten Ansätzen, wie z.B. Design-for-Adaptability (DfA), abzugrenzen. DfA beschäftigt

sich mit Methoden, die es ermöglichen, ein Produkt effizient an sich verändernde Anforderungen, neue Technologien und Überalterung von Komponenten anzupassen. Hierdurch kann sowohl der Wert für den Kunden über den Produktlebenszyklus (Life-Cycle-Value) als auch die Produktlebenszeit an sich erhöht werden.

Zudem ist zu untersuchen, auf welchem Wege Stakeholder effektiver und effizienter in den Produktentwicklungsprozess eingebunden werden können, um durch ein kontinuierliches Feedback späte und dadurch aufwendige Änderungen zu vermeiden. Denkbar wären hierbei Methoden aus dem Feld der Open Innovation, mit welchen sich neben Bedarfs- auch Lösungsinformationen gewinnen lassen. Auf diese Weise können ebenfalls frühzeitig kundenseitige Bevorzugungen oder Ausschlüsse einzelner Domänen für einzelne Teillösungen bestimmt werden.

**Danksagung** Wir danken der Deutschen Forschungsgesellschaft (DFG) für die Förderung unserer Forschung im Rahmen des Sonderforschungsbereichs (SFB) 768 „Zyklusmanagement von Innovationsprozessen – Verzahnte Entwicklung von Leistungsbündeln auf Basis technischer Produkte“.

## 10 Literaturverzeichnis

- Andreasen MM (1995) System Modelling. PhD Course on Design Theory and Re-search, Technical University of Denmark, Lyngby
- Baines TS, Lightfoot HW, Evans S, Neely A, Greenough R, Peppard J, Roy R, Shehab E, Braganza A, Tiwari A, Alcock JR, Angus JP, Bastl M, Cousens A, Irving P, Hohnson M, Kingston J, Lockett H, Martinez V, Michele P, Tranfield D, Walton IM, Wilson H (2007) State-of-the-art in product-service systems. *Journal of Engineering Manufacture*, 221(B):1543–52
- Berkovich M, Leimeister JM, Krcmar H (2009) An empirical exploration of requirements engineering for hybrid products. XVIIth European Conference on Information Systems (ECIS), Verona
- Böhm T, Krcmar H (2007) Hybride Produkte: Merkmale und Herausforderungen. *Wertschöpfungsprozesse bei Dienstleistungen: Forum Dienstleistungsmanagement*
- Böhm T, Langer P, Schermann M (2008) Systematische Überführung von kundenspezifischen IT-Lösungen in integrierte Produkt-Dienstleistungsbausteine mit der SCORE-Methode. *Wirtschaftsinformatik* 50(3):196–207
- Byrd TA, Cossick KL, Zmud RW (1992) A synthesis of research requirements analysis and knowledge acquisition techniques. *MIS Quarterly* 16(1):117–138
- Danilovic M, Browning T (2004) A formal approach for domain mapping matrices (DMM) to complement design structuring matrices (DSM). In: *Proceedings of the Sixth Design Structure Matrix (DSM) International Workshop*, 12.-14.09.2004, Trinity Hall College, University of Cambridge
- Danilovic M, Browning TR (2007) Managing Complex Product Development Projects with Design Structure Matrices and Domain Mapping Matrices. *International Journal of Project Management* 25(3):300–314
- Gräßle M, Thomas O, Fellmann M, Krumeich J (2010) Vorgehensmodelle des Product-Service System Engineering – Überblick, Klassifikation und Vergleich. In Böhm T, Leimeister JM (Hrsg) *Proceedings der Teilkonferenz Integration von Produkt &*



- Dienstleistung – Hybride Wertschöpfung, im Rahmen der Multikonferenz Wirtschaftsinformatik, 23.-25.02.2010, Göttingen. Norderstedt: Books on Demand, 31–42
- INCOSE (2006) Systems Engineering Handbook: A guideline for systems life cycle processes and activities. <http://www.incose.org>
- Knackstedt R, Pöppelbuß J, Winkelmann A (2008) Integration von Sach- und Dienstleistungen – ausgewählte Internetquellen zur hybriden Wertschöpfung. *Wirtschaftsinformatik* 50(3):235–247
- Leimeister JM (2012) Dienstleistungsengineering und -management. Springer, Berlin
- Leimeister JM, Glauner C (2008) Hybride Produkte – Einordnung und Herausforderungen für die Wirtschaftsinformatik. *Wirtschaftsinformatik* 50(3):248–251
- Lindemann U (2009) Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden. Springer, Heidelberg
- Menschner P, Prinz A, Koene P, Köbler F, Altmann M, Krcmar H, Leimeister JM (2011) Reaching into patients' homes – participatory designed AAL services. *Electronic Markets* 21(1):63-76
- Pohl K (2007) Requirements Engineering. Grundlagen, Prinzipien, Techniken. dpunkt. Verlag, Heidelberg
- Ponn J, Lindemann U (2008) Konzeptentwicklung und Gestaltung technischer Produkte: Optimierte Produkte – systematisch von Anforderungen zu Konzepten. Springer, Berlin
- Sadek T (2009) Ein modellorientierter Ansatz zur Konzeptentwicklung industrieller Produkt-Service Systeme. Dissertation, Ruhr-Universität Bochum, Institut für Product-Service Engineering
- Spath D, Demuß L (2003) Entwicklung hybrider Produkte – Gestaltung materieller und immaterieller Leistungsbündel. In Bullinger JH, Scheer A-W (Hrsg) *Service Engineering – Entwicklung und Gestaltung innovativer Dienstleistungen*. Springer, Berlin
- Steward DV (1981) The design structure system: A method for managing the design of complex systems. *IEEE Transactions on Engineering Management* 28:71–74
- Thomas O, Walter P, Loos P (2008) Product-Service Systems: Konstruktion und Anwendung einer Entwicklungsmethodik. *Wirtschaftsinformatik* 50(3):208–219
- Tukker A (2004) Eight types of Product Service Systems: Eight ways to sustainability? Experiences from Suspronet. *Business Strategy and the Environment* 13:246–260
- Weber C, Steinbach M, Botta C, Deubel T (2004) Modelling of Product-Service Systems (PSS) – Based on the PDD Approach. In Marjanovic D (Hrsg) *Design 2004: Proceedings of the 8th International Design Conference, Dubrovnik/Croatia, 18.-21.05.2004*. Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb: 547–554