

Analyse und Verbindung singulärer Vorgehensmodelle zum Software-Service-Co-Design

Marco Apitz, Kyrill Meyer, Martin Böttcher

Abteilung für Betriebliche Informationssysteme
Universität Leipzig
Augustusplatz 10-11
04109 Leipzig
uni@marcoapitz.de
{meyer|boettcher}@informatik.uni-leipzig.de

Abstract: Während der letzten Jahre konnte eine zunehmende Verbindung der Informationstechnologie mit dem Dienstleistungssektor beobachtet werden. Die Verschmelzung dieser Gebiete macht eine Konsolidierung bisher singulärer Entwicklungsprozesse notwendig. Hierfür werden im Folgenden existierende Vorgehensmodelle des Software- und des Service Engineering auf deren Ähnlichkeiten hin überprüft, Metaphasen identifiziert und eine Verbindungsvorschlag dargelegt. Der vorgestellte Ansatz für einen ganzheitlichen Entwicklungsansatz wird prototypisch umgesetzt und evaluiert.

1 Einleitung

Durch zunehmenden Wettbewerb, komplexere Kundenanforderungen und Anforderungen an Effizienz und Effektivität ist in den letzten Jahren eine zunehmende Verbindung der Teilgebiete der Dienstleistungserstellung und der Informationstechnologie zu beobachten [BS03][FF98]. Abhängig vom zugrunde liegenden Basisprodukt können diesbezüglich drei Ausprägungen identifiziert werden: *durch IT unterstützte Dienstleistungen*, *IT-begleitende Dienstleistungen* sowie *Hybridprodukte* [FH04], welche unter den Begriffen IT-basierte Dienstleistungen bzw. IT-Enabled Services zusammengefasst werden. [Bo99].

Die Entwicklung solcher IT-basierter Dienstleistungen involviert verschiedene Fachspezialisten sowie unterschiedliche Entwicklungsmethoden und -werkzeuge. Experteninterviews [FH04] und Breiterehebungen [HOBM06] in diesem Bereich zeigen auf, dass der Entwicklungsprozess Probleme beinhaltet und umfangreiche Herausforderungen stellt. Hierzu zählen Probleme unterschiedlicher Terminologien, methodische Unterschiede sowie mangelnde integrierte Ansätze für einheitliche Entwicklungsansätze zum Co-Design von Software und Services. Um einen solchen einheitlichen Entwicklungsansatz herausstellen zu können ist eine Analyse der Vorgehensmodelle beider Teilbereiche notwendig, um Gemeinsamkeiten, Unterschiede und wieder verwendbare Module herauszustellen.

Dieser Artikel legt die Analyse existierender Vorgehensmodelle beider Bereiche dar, identifiziert Vorgehensmodule und definiert Metaphasen. Darauf aufbauend wird ein Ansatz zur integrierten Entwicklung IT-basierter Dienstleistungen vorgestellt. Abschließend wird eine prototypische Umsetzung zum Tailoring und workflowgesteuerten Ablaufunterstützung aufgezeigt.

2 Analyse existierender Vorgehensmodelle des Service und Software Engineering

Ziel ist es Gemeinsamkeiten, Besonderheiten und Unterschiede zwischen den Vorgehensmodellen des Service und des Software Engineering herauszustellen. Darauf aufbauend werden Teile beider Vorgehensmodellgruppen verbunden, um einen ganzheitlichen Entwicklungsansatz für IT-basierte Dienstleistungen zu erhalten. Die Erfahrungen mit dem phasenorientierten V-Modell XT [Vm05] sowie die meist phasenorientierten Vorgehensmodelle des Service Engineering führen dazu, dass die Analyse der existierenden Vorgehensmodelle darauf ausgelegt ist zu einem ganzheitlichen, phasenorientierten und aus Modulen zusammengesetzten Vorgehensmodell für das Co-Design zu gelangen.

2.1 Analyisierte Vorgehensmodelle

Zur Analyse der existierenden Vorgehensmodelle des Service und des Software Engineering wurden etablierte und an spezielle Anforderungen adaptierbare (Tailoring) Vorgehensmodelle analysiert (Tabelle 1). Die Untersuchung wurde aufgrund der Tatsache, dass das Service Engineering bisher nur Phasenorientierte Entwicklungsansätze kennt, im wesentlichen auf entsprechende Vorgehensmodelle beschränkt.

Service Engineering:	Software Engineering:
Scheuning und Johnson	Phasen- und Wasserfallmodell
Shostack	Prototypenmodelle
Edvardsson und Olsson	Evolutionäres und inkrementelles Modell
Ramaswamy	Spiralmodell von Boehm
Jaschinsky	V-Modell des Bundes
DIN	Object Modeling Technique
FhG IAO	Booch-Vorgehensmodell
	Objectory
	Extreme Programming
	Catalysis
	Unified Process

Tabelle 1: Analyisierte Vorgehensmodelle des Service und Software Engineering

2.2 Metaphasen und Module

Die analysierten Vorgehensmodelle wurden entlang einer Zeitlinie gegenübergestellt, so dass einerseits abstrakte, von den Terminologien der einzelnen Vorgehensmodelle unab-

hängige Metaphasen abgeleitet und andererseits ein Vergleich entsprechend dieser Phasen vorgenommen werden konnte (Abbildung 1).

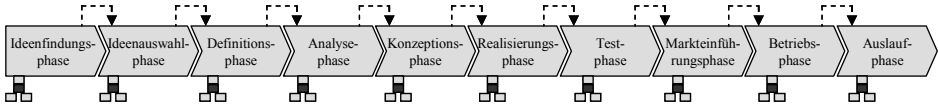


Abbildung 1: Extrahierte Metaphasen aus den Vorgehensmodellen des Service und Software Engineering

Um, ähnlich dem V-Modell XT, abstrakte Module für eine konkrete Instanziierung eines Vorgehensmodells zum Software-Service-Co-Design zu erhalten, wurden die Vorgehensmodelle einheitlich und entsprechend den identifizierten Phasen untergliedert (Tabelle 2, Tabelle 3). Hierbei konnten erste Unterschiede zwischen den Vorgehensmodellen des Service und Software Engineering festgestellt werden. So können einzelne, den Metaphasen zugeordnete Module unterschiedlich exakt abgrenzbar aus den Vorgehensmodellen extrahiert werden, da einerseits der modulare Ansatz bei einigen Modellen kaum berücksichtigt ist und andererseits einzelne Phasen in bestimmten Vorgehensmodellen zusammengefasst auftreten. Es ist beispielsweise erkennbar, dass die Phasen Definition und Analyse sowie Konzeption und Realisierung in Vorgehensmodellen des Service Engineering im Gegensatz zum Software Engineering immer zusammengefasst existieren und somit nur schwer voneinander trennbar sind. Für einen einheitlichen modularen Ansatz mit der Möglichkeit der Kombination der Module zu einem ganzheitlichen Entwicklungsvorgehen müssen für alle zu nutzenden Vorgehensmodelle die phasenorientierten Module mit gleicher Granularität identifiziert und beschrieben werden.

Modell	Phase	Ideenfindung	Auswahl	Definition	Anforderung	Konzeption
Service Engineering						
Scheuring / Johnson	Formulation of new Service Objectives & Strategies					
	Idea Generation	Idea Screening	Concept Developing			
			Concept Testing	Business Analysis		
				Project Authorisation	Service Design & Testing	
DIN	Ideenfindung und Bewertung			Anforderungen	Design	
Software Engineering						
Wasserfallmodell			Definition	Analyse	Entwurf	
X-treme Programming			User Stories			

Tabelle 2: Exemplarische Module von Vorgehensmodellen entsprechend der Metaphasen (Teil 1)

Die Analyse ergab des Weiteren, dass die Vorgehensmodelle des Service und Software Engineering unterschiedliche Gewichtung in den einzelnen Bereichen eines Entwicklungszyklus legen. So bieten die Vorgehensmodelle des Software Engineering umfangreiche Unterstützung in den Abschnitten Analyse bis Test, wohingegen bei den Vorgehensmodellen des Service Engineering ein sehr starker Fokus auf die Phasen der Ideenfindung und -auswahl existiert. Diesbezüglich ist erkennbar, dass für ein angestrebtes Software-Service-Co-Design existierende singuläre Vorgehensmodelle nicht ausreichen, um den gesamten Bereich IT-basierter Dienstleistungen abzudecken. Daher stellt die Modularisierung und Kombination einen sinnvollen Ansatz zur Nutzung der kompletten Entwicklungsbreite dar.

Realisierung	Test	Einführung	Betrieb	Auslauf	Phase Modell
Service Engineering					
Process & System Design & Testing					Scheuring / Johnson
Marketing Program Design & Testing					
Personal Training	Service Testing & Pilot Run				
	Test Marketing	Full-Scale Launch			
		Post-Launch Review			
		Einführung	Erbringung	Ablösung	DIN
Software Engineering					
Implementierung	Test	Installation	Einsatz		Wasserfallmodell
Iteration	Acceptance Tests	Small Releases			X-treme Programming

Tabelle 3: Exemplarische Module von Vorgehensmodellen entsprechend der Metaphasen (Teil 2)

Eine weitere entscheidende Problematik für die Umsetzung des Software-Service-Co-Design wurde bereits mit dem Hinweis auf die Kommunikationsdifferenzen zwischen den Spezialisten beider Bereiche hervorgehoben und konnte durch die durchgeführte Analyse bestätigt werden, da insbesondere zwischen den Bereichen des Software und Service Engineering elementare terminologische Unstimmigkeiten festgestellt wurden. Eines von vielen Beispielen ist der Begriff „Konzepte“, welcher im Software Engineering erst in der Konzeptionsphase, bei der Dienstleistungsentwicklung schon zuvor in der Definitionsphase genutzt wird.

Die Analyse der Vorgehensmodelle ergab, dass selbst innerhalb eines der beiden Teilgebiete IT-basierter Dienstleistungen die Vorgehen mit unterschiedlichem Umfang und Granularität beschrieben werden. So geben beispielsweise das DIN- wie auch das allgemeine Wasserfallmodell nur einen groben abstrakten Entwicklungsablauf vor, wohingegen das V-Modell sehr pedantisch die durchzuführenden Aktivitäten definiert. Auch in den Definitionen der Rollen, welche für die einzelnen Vorgehensschritte verantwortlich sind können starke Unterschiede festgestellt werden. Während die Vorgehensmodelle des Service Engineering zwischen Entwicklern, Kunden und Management unterschei-

den, differenzieren einzelne Modelle des Software Engineering wesentlich stärker (z.B. das V-Modell mit 25 Rollen).

Aus der Analyse der Vorgehensmodelle des Service und Software Engineering unter dem Aspekt der Modularisierung und Wiederverwendung im Rahmen eines Software-Service-Co-Design ging hervor, dass eine Modularisierung und Weiterverwendung existierender Vorgehensmodelle grundlegend möglich ist. Des Weiteren konnte festgestellt werden, dass keines der betrachteten Vorgehensmodelle die vollständige Bandbreite beider Entwicklungsbereiche abdeckt, so dass eine Verbindung notwendig wird. Hierbei ist eine phasenweise Ausrichtung des zu kombinierenden Modells empfohlen, da eine solche von den meisten analysierten Vorgehensmodellen angewandt und eine bessere Verständlichkeit seitens der Nutzer gewährleistet wird.

3 Ansatz für die integrierte Entwicklung IT-basierter Dienstleistungen

Anhand der identifizierten Metaphasen wird die Entwicklung IT-basierter Dienstleistungen in zehn Phasen untergliedert, wobei jede Phase aus beliebig vielen Prozessschritten bestehen kann, welche wiederum einen jeweils abgeschlossenen Komplex von Aktivitäten darstellen. Prozessschritte wurden aus den einzelnen Vorgehensmodellen extrahiert und sind lediglich durch den sequentiellen Ablauf der Phasen in ihrer Reihenfolge beschränkt.

In Anlehnung an die vier Gestaltungsdimensionen einer Dienstleistung (Potential-, Prozess-, Ergebnis- und Marktdimension nach [Bul03]) sieht dieser Entwicklungsansatz vier Schichten vor: Modulschicht, Prozessschicht, Ergebnisschicht und Marktschicht (Abbildung 2).

Die Modulschicht reflektiert Potentiale und Fähigkeiten der beteiligten Personen- und Wirtschaftseinheiten, welche in Form von Prozessmodulen abstrahiert wurden. Neben der Modulbibliothek existiert in der Modulschicht auch eine so genannte Metamodulbibliothek, in der zeitunabhängige oder sich in mehreren Phasen wiederholende Prozessschritte eingepflegt sein können.

Die Prozessschicht gibt durch eine Komposition von Prozessmodulen die zeitliche Abfolge der einzelnen Prozessschritte an. Ein Prozessschritt ist eine zeitlich logische Abfolge von Aktivitäten, die einer oder mehreren Ressourcen bedürfen, um ein Ergebnis zu erzeugen. Jedes Prozessmodul besitzt Schnittstellen in Form von Eingabe- und Ausgabeinformationen sowie Rollenzuordnungen. Des Weiteren werden für jedes Modul spezifische Eigenschaften erklärt, welche eine Entscheidungsgrundlage für Tailoring und Modulauswahl stellen. Demzufolge werden Prozessschritte den einzelnen Metaphasen entsprechend einem Projekt zugeordnet.

Die Ergebnisschicht enthält Informationen und Erkenntnisse, welche durch bereits durchgeführte Prozessschritte entstanden sind. Der Markt wird durch die Unternehmensbedingungen und Projekteigenschaften symbolisiert, die den Ausgangspunkt für das

Tailoring des Vorgehensmodells bilden, aber auch Einfluss auf den Prozessablauf und die erzielten Ergebnisse nehmen.

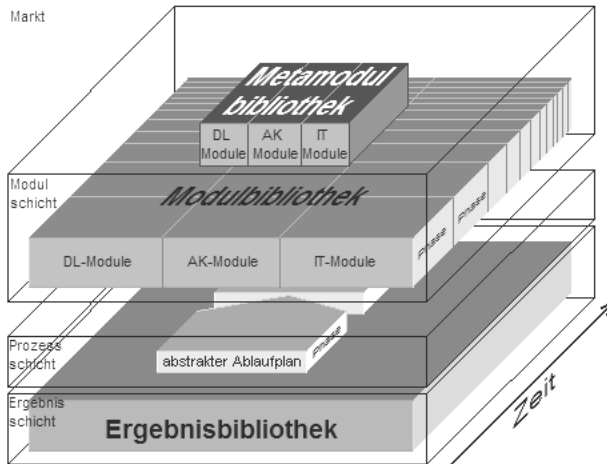


Abbildung 2: Aufbau des Frameworks

4 Prototypische Umsetzung und praktischer Einsatz

Um die Anwendung und Handhabung eines solchen Vorgehensmodells in der alltäglichen Arbeitspraxis zu gewährleisten, ist aufgrund der Komplexität des Modells eine Softwareunterstützung notwendig. Da für die Entwicklung IT-basierter Dienstleistungen bisher keine einheitlichen und durchgängigen Unterstützungswerkzeuge existieren wurde ein webbasierter Prototyp entwickelt, welcher die dargelegten Vorgehensmodellansätze integriert.

Der Prototyp kann genutzt werden, um mit ARIS modellierte Vorgehensmodelle zu laden und diese mittels einer Tailoring-Komponente dem entsprechenden Projekt anzupassen. Dieses adaptierte Vorgehensmodell zum Software-Service-Co-Design wird anschließend genutzt, um eine Ablaufsteuerung zu unterstützen. Diese Ablaufsteuerung nutzt die Workflow-Engine YAWL. Des Weiteren werden angefertigte und genutzte Artefakte hinterlegt, so dass diese in späteren Phasen wieder verwendet werden können.

5 Fazit

Motivation dieser Arbeit war die Notwendigkeit nach einem einheitlichen Entwicklungsvorgehen für IT-basierte Dienstleistungen, welche eine Verbindung der Teilgebiete des Software und Service Engineering notwendig macht. Somit wurden existierende Vorgehensmodelle beider Bereiche auf deren Verbindungsmöglichkeiten hin analysiert und ein Vorschlag für ein gemeinsames Software-Service-Co-Design dargelegt. Dieser wurde prototypisch implementiert, um deren Anwendbarkeit zu überprüfen. Somit wur-

de erstmals ein einheitlicher Entwicklungsansatz vorgestellt, der von der Praxis gefordert, jedoch bisher noch nicht umgesetzt wird.

Der vorgeschlagene Ansatz bietet eine erste Möglichkeit eines gemeinsamen Designs. Zukünftig ist jedoch die Betrachtung neuerer Entwicklungsansätze notwendig, welche über die phasenweise Betrachtung hinausgehen. Während es im Software Engineering agile Ansätze bereits gibt, werden solche Vorgehen bisher im Bereich des Service Engineering noch nicht eingesetzt. Daher wird die Analyse der Übertragbarkeit solcher agiler Vorgehen auf den Bereich des Service Engineering eine nächste Aufgabe darstellen. An solche modernen Ansätze muss dann ebenfalls die Werkzeugunterstützung entsprechend angepasst werden.

Literaturverzeichnis

- [Ba98] Balzert, Helmut: Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung, Spektrum, Akad. Verl., Heidelberg, Berlin, 1998.
- [BK02] Bunse, C.; von Knethen, A.: Vorgehensmodelle kompakt. Spektrum Akademischer Verlag, Heidelberg, Berlin. 2002.
- [Bo99] Bodendorf, F.: Wirtschaftsinformatik im Dienstleistungsbereich. Springer: Berlin et al., 1999.
- [Br98] Bremer, G.: Genealogie von Entwicklungsschemata. In: Kneuper, R.; Müller-Luschat, G; Oberweis, A. (Hrsg.): Vorgehensmodelle für die betriebliche Anwendungsentwicklung. Teubner, Stuttgart, Leipzig. 1998.
- [BS03] Bullinger, Hans-Jörg; Scheer, August-Wilhelm (Hrsg): Service Engineering. Entwicklung und Gestaltung innovativer Dienstleistungen. Springer, Berlin et al. 2003.
- [FF98] Fitzsimmons, J. A.; Fitzsimmons, M. J.: Service management: operations, strategy, and information technology. Irwin McGraw-Hill: Boston et al, 1998.
- [FH04] Fähnrich, Klaus-Peter; van Husen, Christian (Hrsg.): Entwicklung IT-basierter Dienstleistungen in der Praxis. Kurzstudie zum Co-Design von Software und Services in deutschen Unternehmen. Fraunhofer IRB Verlag, Stuttgart, 2004.
- [HOBM05] van Husen, C.; Opitz, M.; Böttcher, M.; Meyer, K.: Co-Design von Software und Services – Studie zur Entwicklung IT-basierter Dienstleistungen in deutschen Unternehmen. Fraunhofer IRB Verlag, Stuttgart, 2005.
- [Ja98] Jaschinski, C.: Qualitätsorientiertes Redesign von Dienstleistungen. Shaker. Aachen. 1998.
- [MB03] Meiren, T.; Barth, T.: Service Engineering in Unternehmen umsetzen. Leitfaden für die Entwicklung von Dienstleistungen. Fraunhofer IRB. Stuttgart. 2003.
- [Ra96] Ramaswamy, R.: Design and Management of Service Processes. Keeping Customers for Life. Reading-Messachusetts: Addison-Wesley. 1996.
- [So04] Sommerville, Ian: Software Engineering. Pearson Studium. München. 2004.
- [Vm05] Bundesrepublik Deutschland (Hrsg.): V-Modell XT. <http://www.kbst.bund.de/V-Modell/-,293/V-Modell-XT.htm>, Abruf am 21.09.2005.
- [Ya05] YAWL – Homepage. <http://www.yawl.fit.qut.edu.au/>, Abruf am: 21.09.2005.