

Vorgehensmodell zur Entwicklung von Geschäftsservices

Sebastian Stein
IDS Scheer AG
Altenkesseler Str. 17
66115 Saarbrücken
sebastian.stein@ids-scheer.com

Konstantin Ivanov
IDS Scheer AG
Altenkesseler Str. 17
66115 Saarbrücken
konstantin.ivanov@ids-scheer.com

Kurzfassung: Eine service-orientierte Architektur (SOA) sollte sich aus den Anforderungen der Fachabteilung ableiten. Die fachlichen Anforderungen werden dabei häufig als Geschäftsprozesse modelliert. Was aber genau Geschäftsprozessmodellierung umfasst, ist umstritten. Viele verstehen darunter lediglich die Orchestrierung technischer Services zur Umsetzung betriebswirtschaftlicher Abläufe. Diese IT-zentrierte Sichtweise betrachtet jedoch lediglich die technischen Aspekte einer SOA. Betriebswirtschaftliche Fragestellungen bleiben dabei unberücksichtigt.

In diesem Beitrag stellen wir ein Vorgehensmodell vor, das die betriebswirtschaftlichen Gesichtspunkte einer SOA in den Vordergrund rückt und die konsequente Ableitung der Serviceorchestrierung aus den Geschäftsprozessen fördert. Neben dem Vorgehensmodell werden die an solch einem Projekt beteiligten Personen in Form von Rollen charakterisiert.

1 Einleitung und Motivation

Service-orientierte Architekturen (SOA) [MSJL06, siehe z. B.] sind ein aktueller Trend zur Integration verschiedener Anwendungssysteme. Solch eine Integration ermöglicht es, Funktionalitäten unabhängiger Anwendungssysteme miteinander zu kombinieren. Eine ausführliche Diskussion des Integrationsbegriffs findet sich bei Thränert [Thr05].

Im Rahmen einer SOA werden die Anwendungssysteme in von einander unabhängige Services gegliedert. Die Services können anschließend miteinander kombiniert werden. Die technische Kombination wird als Orchestrierung bezeichnet und sie erfolgt mittels der Business Process Execution Language (BPEL) [ACD⁺03]. Die Beschreibung der Serviceschnittstellen erfolgt mit der Web Service Definition Language (WSDL) [CCMW01]. Neben diesen beiden Standards sind auch andere Technologien zur Umsetzung einer SOA möglich.

In diesem Beitrag zeigen wir, wie man vorgehen kann, um mittels Orchestrierung neue Geschäftsservices zu definieren. Ausgangspunkt sind die Geschäftsprozesse und somit die fachlichen Anforderungen der zukünftigen Anwender der Integrationslösung. Dieser Ansatz wird von den Analysten von Gartner [NPSI06] als *business-driven SOA* bezeichnet und ist nach ihrer Analyse ein entscheidender Fokus für SOA Projekte in den kommenden Jahren.

Im nächsten Abschnitt diskutieren wir kurz verwandte Arbeiten und grenzen den Inhalt dieses Artikels entsprechend ab. Im Abschnitt 3 beginnend auf Seite 3 stellen wir das Vorgehensmodell im Detail vor. Dabei gehen wir zunächst auf das verwendete Servicekonzept ein und charakterisieren die am Vorgehensmodell beteiligten Rollen. Anschließend werden die einzelnen Aktivitäten

sowie deren Beziehungen untereinander ausführlich dargestellt. Am Ende folgt eine kurze Zusammenfassung.

2 Verwandte Arbeiten

Ein typisches Anwendungsgebiet für Vorgehensmodelle ist das Software Engineering. So definiert z. B. Sommerville [Som04] den Begriff *software process* als eine Menge von Aktivitäten, die zur Produktion eines Softwareprodukts führen. Für das Software Engineering existiert eine Vielzahl verschiedener Vorgehensmodelle. Sommerville (ebd.) führt dies auf die großen Unterschiede bei Softwareprojekten und den daran beteiligten Menschen zurück. Je nach Produktcharakteristik und Wissen der Beteiligten können ganz unterschiedliche Vorgehensmodelle zum Einsatz kommen. Typische Beispiele für Vorgehensmodelle im Software Engineering sind z. B. der Rational Unified Process (RUP) [Kru00] und Extreme Programming (XP) [Bec99]. Vorgehensmodelle wie RUP und XP geben dabei sehr konkrete Strukturen, Artefakte, Phasen und Konzepte vor.

Im Gegensatz zu solch konkreten Vorgehensmodellen, gibt es eine Reihe von Modellen und Rahmenwerken, die abstrakt die Kriterien eines guten Prozesses definieren. Unter gut wird dabei ein Prozess verstanden, der zu hoher Prozessqualität führt. Dabei wird angenommen, dass ein qualitativ hochwertiger Entwicklungsprozess auch ein qualitativ hochwertiges Produkt hervor bringt. Ein typischer Vertreter in dieser Kategorie ist das V-Modell XT [RH05]. So definiert es Projektmanagement als einen integralen Bestandteil eines Softwareprojekts, die konkrete Umsetzung muss aber für jedes Projekt individuell definiert werden.

Die geringsten konkreten Vorgaben für das Vorgehen in einem Softwareprojekt machen Qualitätsrichtlinien wie ISO 9000:2000ff¹ und das Capability Maturity Model Integration [Cmm02]. Allerdings gibt es für diese Richtlinien Umsetzungen und Empfehlungen zur Anwendung in Softwareprojekten, wie z. B. SPICE² und TickIT [Tic01] für ISO 9000:2000ff.

Das in diesem Artikel vorgestellte Vorgehensmodell erhebt nicht den Anspruch so vollständig und umfassend zu sein, wie z. B. RUP und XP. Die Hauptintention liegt darin, dass SOA Implementierungen konsequent aus den Geschäftsprozessen abgeleitet werden. Dabei spielt es letztendlich keine Rolle, ob dies in einem sequenziellen oder iterativ-inkrementellen Vorgehen [Som04, siehe z. B.] geschieht!

Nach Wissen der Autoren existiert noch kein vollständiges Vorgehensmodell für SOA Projekte. Entsprechend der Ausführungen am Anfang dieses Abschnittes ist es auch fraglich, ob konkrete Vorgehensmodelle für SOA Projekte aufgrund der sehr unterschiedlichen Projektparameter überhaupt möglich sind. Nichtsdestotrotz gibt es zumindest Vorschläge für Teilphasen eines SOA Projekts. So schlägt Jones [JM05] z. B. eine SOA Methodik vor, die die Phase der primären Beschreibung der Servicearchitektur regelt. Diese Methodik soll dabei unterstützen, Services zu identifizieren und zu definieren. Dazu werden die Hauptbereiche des Unternehmens zunächst als Services beschrieben und dann schrittweise in kleinere Services zerlegt. Neben konkreten Hinweisen zur Darstellung der so entwickelten Servicearchitektur³ gibt Jones (ebd.) noch Empfehlungen

¹ siehe z. B. <http://www.iso9001.qmb.info/>

² siehe z. B. <http://www.isospice.com/>

³ ein ähnlicher Vorschlag ist die Service Component Architecture (SCA) - <http://osoa.org/>

zum Erstellungsprozess. So schlägt er eine Reihe von Gruppensitzungen vor, charakterisiert den Teilnehmerkreis und benennt die notwendigen Vorbereitungsmaßnahmen.

Einen weiteren Ansatz namens *Business-driven development* (BDD) präsentiert Mitra [Mit05]. Ausgehend vom klassischen Deming Zyklus [Dem82] für kontinuierliche Prozessverbesserung definiert der Autor verschiedene Phasen und Schritte, um eine SOA entsprechend den Bedürfnissen der Fachabteilung zu gestalten. Köhler et al. [KHK⁺06] diskutieren den aktuellen Stand der Umsetzung von BDD und zeigen, dass noch intensive Forschungsarbeit notwendig ist, um BDD vollständig umzusetzen.

Es existieren weitere Vorschläge zur Strukturierung von Servicearchitekturen. Ein interessanter Ansatz ist die Übertragung von Konzepten für komponentenorientierte Softwarearchitekturen wie Herzum et al. [HS00] auf SOA. Eine mögliche Umsetzung wird sehr umfassend in McGovern et al. [MSJL06] diskutiert.

Das von uns vorgestellte Vorgehensmodell macht keine Vorschläge zur Strukturierung einer Servicearchitektur. Wir stimmen den diskutierten Autoren aber dahingehend zu, dass es mehrere verschiedene Arten von Services gibt. Zumindest eine Unterscheidung zwischen Geschäfts- und Infrastrukturservices ist notwendig. Auch weisen Services einen unterschiedlichen Abstraktionsgrad auf und differieren in ihrem Umfang. Diese Erkenntnis erübrigt auch die Frage nach der Formel für ein perfektes Servicedesign. Weiterhin können wir anhand unserer Projekterfahrung bestätigen, dass eine proaktiv verwaltete Servicearchitektur ein wesentlicher Erfolgsfaktor für eine erfolgreiche SOA Implementierung ist.

Das im folgenden Abschnitt vorgestellte Vorgehensmodell besteht aus 3 Rollenbeschreibungen sowie 10 Aktivitäten sowie deren Abhängigkeit zueinander. Auf Integrationsmöglichkeiten mit anderen Methodiken und Konzepten wird entsprechend hingewiesen. Hinsichtlich des Abstraktionsgrad ist unser Vorgehensmodell daher gut vergleichbar mit BDD [Mit05] sowie den verschiedenen Qualitätsrichtlinien. Da unser Vorgehensmodell mit der ARIS Methode nach Scheer [Sch02] integriert ist, kann eine SOA Initiative ausgehend von vorhandenen Investitionen in Geschäftsprozessmodellierung und -management gestartet werden.

3 Vorgehensmodell

3.1 Übersicht

Bevor die einzelnen Aktivitäten des Vorgehensmodells näher erläutert werden, muss zuvor ein kurzer Überblick über das verwendete Servicekonzept gegeben werden. Weiterhin erfolgt im Unterabschnitt 3.3 auf Seite 5 eine Beschreibung der 3 Hauptrollen. Die einzelnen Aktivitäten werden in Unterabschnitt 3.4 beginnend auf Seite 5 erläutert.

3.2 Servicekonzept

Obwohl mit dem *SOA-Reference Model* (SOA-RM) [MLM⁺06] ein sehr umfangreiches Servicekonzept allgemein verfügbar ist, haben wir ein wesentlich einfacheres Modell auf Basis des SOA-RM entworfen. Hintergrund ist, dass die vielen zusätzlichen Details im SOA-RM nicht bei der Erläuterung des Vorgehensmodells hilfreich sind, sondern eher verwirren.

Unser Servicekonzept besteht aus den folgenden drei Ebenen:

1. Ebene zur Beschreibung der Anforderungen an den Service
2. Ebene zur Beschreibung des fachlichen und technischen Designs des Service
3. Ebene zur Darstellung der Ausführungs- und Implementierungsdetails des Service

In Ebene 1 werden die funktionalen und nicht-funktionalen Anforderungen an den Service beschrieben. Diese Ebene ist nicht nur interessant während der Erstellung des Service, sondern auch zur späteren Nutzung, da die Anforderungen letztendlich auch die Fähigkeiten (Capabilities) des implementierten Service charakterisieren. Zur Darstellung der Fähigkeiten sind prinzipiell verschiedene Ansätze denkbar, so z. B. die Verwendung von speziellen Ontologien wie WSMO⁴. Wir schlagen hingegen ein wesentlich einfacheres Vorgehen vor. Für jede Anforderung wird ein aussagekräftiges Stichwort (Tag) gewählt und jeder Service wird mit all den Stichwörtern versehen (tagging), deren Anforderungen er erfüllt. Den Stichwörtern können umfangreichere Beschreibungen zugeordnet werden. Auch eine Strukturierung der Stichwörter in Hierarchien ist denkbar. Ziel muss dabei sein, für ähnliche Anforderungen bei unterschiedlichen Services gleiche Stichwörter zu verwenden. Dadurch entsteht zwischen Service und Stichwort eine n:m Beziehung. Eine Analyse sowohl von Sicht des Service mit allen zugeordneten Stichwörtern als auch aus Sicht des Stichworts mit allen zugeordneten Services ist aufschlussreich. So könnte das Stichwort *Realtime* allen Services zugeordnet werden, die in ihrem Antwortverhalten Echtzeitfähigkeit garantieren müssen. Ist einem Service dieses Stichwort zugeordnet, ergeben sich spezielle Anforderungen an das Servicedesign und die genutzte Hardwareplattform. Aus Sicht des Stichwortes kann man alle Services mit Echtzeitfähigkeit identifizieren und so z. B. Konsolidierungsmaßnahmen bei der eingesetzten Hardware ergreifen.

In Ebene 2 wird das fachliche und technische Design des Service beschrieben. Dabei muss grundsätzlich zwischen dem internen dynamischen Verhalten des Service und seiner statischen Schnittstelle zur Außenwelt unterschieden werden. Das dynamische Verhalten kann aus fachlicher Sicht mit Geschäftsprozessmodellen beschrieben werden. Ein Industriestandard dafür ist die Ereignisgesteuerte Prozesskette (EPK) [KNS92]. Das technisch dynamische Verhalten kann mit BPEL [ACD⁺03] beschrieben werden. Die statische Serviceschnittstelle kann wiederum aus fachlicher und technischer Sicht beschrieben werden. Aus fachlicher Sicht können die fachlichen Datenobjekte⁵ als Ein- und Ausgabedaten angegeben werden. Aus technischer Sicht sollte die Serviceschnittstelle mit WSDL [CCMW01] beschrieben werden.

⁴Web Service Modeling Ontology - <http://www.wsmo.org/TR/d2/>

⁵z. B. Kunde, Rechnung, Auftrag, etc.

In Ebene 3 wird dargestellt, wie der Service zur Ausführung kommt. So kann z. B. die Adresse zum Aufruf des Service angegeben werden. Auch kann angegeben werden, auf welcher Ausführungsplattform der Service platziert wird.

Prinzipiell unterscheiden wir mindestens Infrastruktur- und Geschäftsservices. Infrastrukturservices werden durch die verwendeten Anwendungssysteme zur Verfügung gestellt. Unser Vorgehensmodell konzentriert sich auf die Entwicklung von Geschäftsservices. Ein Geschäftsservice kapselt dabei einen Teil eines Geschäftsprozesses.

3.3 Rollen

Im Vorgehensmodell werden drei Rollen verwendet. In diesem Abschnitt wird kurz die Hauptaufgabe jeder Rolle sowie die wichtigsten Fähigkeiten beschrieben.

Der Business Analyst entwickelt die Geschäftsprozessmodelle, ordnet die vorhandenen Services fachlich ein, baut ein fachliches Datenmodell auf und definiert die fachlichen Anforderungen an einen Service. Dabei verfügt er über umfangreiches betriebswirtschaftliches Wissen sowie über genaue Kenntnisse über das Unternehmen und seine Kunden.

Der Process Engineer unterstützt bei der fachlichen Beschreibung von Services und der Transformation des Geschäftsprozessmodells in eine entsprechende technische Orchestrierung sowie beim Deployment in der Ausführungsplattform. Dabei verfügt er sowohl über betriebswirtschaftliche Grundkenntnisse als auch über ein umfangreiches technisches Wissen im Bereich der genutzten SOA Technologie.

Der Software Developer unterstützt bei der Implementierung und Anpassung von Services. Er verfügt über Software Engineering Kenntnisse und er ist ein Experte für die eingesetzte Ausführungsplattform.

3.4 Aktivitäten

Die Aktivitäten, deren Reihenfolge sowie die die Aktivitäten ausführenden Rollen sind in Abbildung 1 auf Seite 6 dargestellt. Der dargestellte Ablauf muss für jeden zu entwickelnden Service komplett durchlaufen werden, ebenso für jede neue Version eines Service. Ausgehend von den Geschäftsprozessmodellen wird schrittweise die technische Beschreibung des Service in einem top-down Ansatz abgeleitet.

Im Schritt 1 erfolgt eine service-orientierte Geschäftsprozessmodellierung. Zunächst werden übergeordnete fachliche Prozessmodelle erstellt, die selber nicht zur Ausführung bestimmt sind. Ausgehend von diesen abstrakten Prozessen, erfolgt die service-orientierte Prozessmodellierung

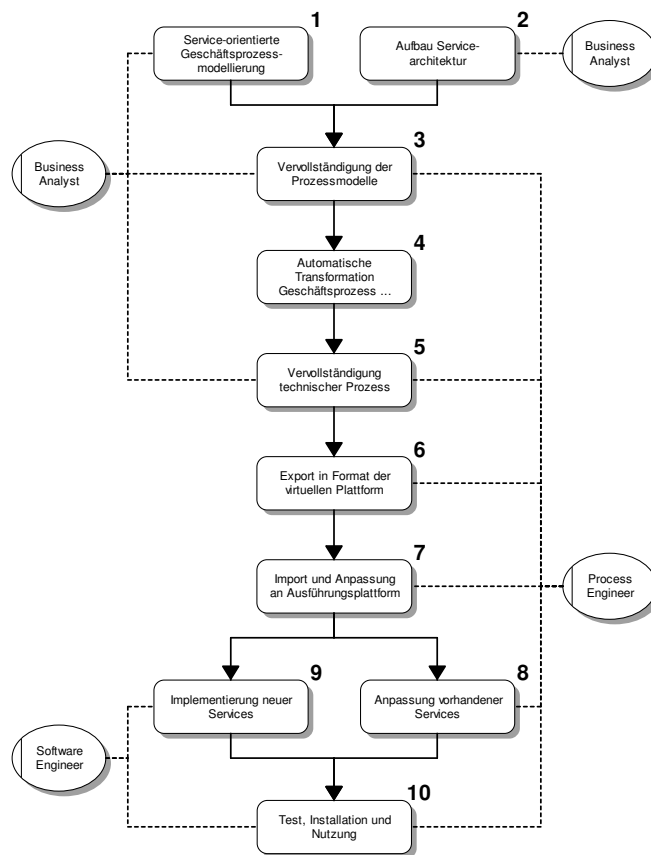


Abbildung 1: Vorgehensmodell mit 10 Schritten zur Entwicklung von Geschäftsservices

in einem top-down Ansatz. Durch die Modellierung der Fachprozesse wird ersichtlich, welche Eigenschaften Services aufweisen müssen, um die fachlichen Aktivitäten später technisch unterstützen zu können.

Diese Modellierung muss so lange durchgeführt werden, bis ein Prozess nur aus Funktionen besteht, die mit Services automatisiert werden können. Die Anforderungen an die Services müssen unter Angabe der relevanten Stichwörter (siehe 3.2) erfolgen.

Die service-orientierte Geschäftsprozessmodellierung ist nicht trivial. Ausgangspunkt kann z. B. die ARIS Methode [Sch02] sein. Eine weitere Herausforderung besteht in einer sinnvollen Gliederung der Prozesshierarchie. Hier können Ansätze wie bei Jones [JM05] oder bei Herzum et al. [HS00] behilflich sein.

Im Schritt 2 werden bereits existierende Services in die Servicearchitektur mit aufgenommen und ihre Fähigkeiten mit Stichwörtern beschrieben. Dieser Schritt kann parallel zum ersten Schritt erfolgen. Während der ersten beiden Schritte erfolgt somit der schrittweise Aufbau der Stichworthierarchie zur Beschreibung von Anforderungen und Fähigkeiten. Diese fachliche Beschreibung der Services ergänzt eine eventuell vorhandene technische Beschreibung in Form von WSDL.

Bei der Beschreibung von Services sollte auf eine Trennung in fachliche und technische Beschreibung geachtet werden. Ein Business Analyst kann wahrscheinlich wenig mit Interfaces und zugehörigen Operationen anfangen. Deshalb ist es auch nicht sinnvoll, in fachlichen Prozessmodellen die technischen Datentypen zu verwenden, die in WSDL und zugehörigen Dateien beschrieben werden. In der Praxis hat sich gezeigt, dass zur korrekten Darstellung des verwendeten technischen Datenmodells eine fachliche Informationsarchitektur entwickelt werden muss. Das Mapping zwischen fachlicher Informationsarchitektur und technischem Datenmodell kann dabei nur in Zusammenarbeit von Business Analyst und Process Engineer erfolgen.

Im Schritt 3 werden nun den Funktionen im fachlichen Prozessmodell Services zugeordnet. Dabei können moderne Matching Algorithmen bei der Auswahl der richtigen Services anhand der Stichwörter helfen.

Wird für eine Funktion kein Service gefunden, der alle Anforderungen erfüllt, so gibt es mehrere Optionen. Der Service kann z. B. in einem Software Engineering Projekt entwickelt werden. Dieser Fall ist mit Schritt 8 in unserem Vorgehensmodell vorgesehen. Alternativ kann geprüft werden, ob externe Anbieter einen entsprechenden Service zur Verfügung stellen. Andererseits kann die fachliche Funktion eventuell in weitere Schritte zerlegt werden, für die jeweils Services zur Automatisierung vorhanden sind.

Neben der Zuordnung von Services zu Funktionen muss der Datenfluss mittels fachlicher Datenobjekte vervollständigt werden. Neben dem internen Datenfluss müssen weiterhin die fachlichen Datenobjekte angegeben werden, die zur Ausführung des Geschäftsservice benötigt werden und welche als Ergebnis der Ausführung zurück geliefert werden. Darüber hinaus müssen für Entscheidungspunkte im Fachprozess die entsprechenden Geschäftsregeln definiert werden.

Im Schritt 4 wird die fachliche Prozessdefinition in eine technische Beschreibung wie BPEL überführt. Solch eine Transformation sollte möglichst automatisch geschehen, da eine manuelle Überführung zu einer Neuinterpretation der Modellinhalte und somit zu einer Fehlerquelle führen würde. Durch eine automatische Transformation wird das technische Prozessdesign direkt aus den fachlichen Anforderungen abgeleitet.

In Stein et al. [SI07] stellen wir eine entsprechende Transformation vor. Das Ausgangsmodell ist ein service-orientierter Geschäftsprozess modelliert mit einer EPK. Das Zielmodell ist BPEL. Die von uns entwickelte Transformation überführt nicht nur die Prozessstruktur, sondern alle weiteren in der EPK modellierten Inhalte. So werden z. B. fachliche Funktionen mit zugeordnetem Service in einen entsprechenden Aufruf des Service umgewandelt. Die fachlichen Ein- und Ausgabedaten von Funktionen werden in die entsprechenden technischen Ein- und Ausgabedaten umgewandelt. Weiterhin wird automatisch eine Interfacebeschreibung in Form einer WSDL Datei für den neuen Geschäftsservice erzeugt. Andere Transformationsansätze sind in Stein et al. (ebd.) sowie in Köhler et al. [KHK⁺06] ausführlich beschrieben.

Prinzipiell ist bei der Transformation zu beachten, dass das Zielmodell eine virtuelle Plattform sein sollte, die auf industrieweiten Standards basiert. Nur dies garantiert, dass die SOA wirklich unabhängig von der eingesetzten realen Infrastruktur ist und nicht mit Wechsel der Infrastruktur alle Services angepasst werden müssen!

Im Schritt 5 wird das erzeugte technische Prozessmodell vervollständigt. Dabei dürfen nur technische Änderungen vorgenommen werden, die keine fachlichen Auswirkungen haben. So kann es z. B. notwendig sein, zwischen verschiedenen Datentypen zu konvertieren, wenn die orchestrierten Services unterschiedliche Datenformate verwenden.

Eine weitere Aufgabe ist es, das technische Modell robust zu machen. Dazu zählt z. B. die Verarbeitung von technischen Ausnahmezuständen (Exceptions). Dabei ist darauf zu achten, dass nicht versucht wird, fachliche Ausnahmen mit diesen Mitteln zu behandeln. Besteht z. B. eine mögliche Ausnahme darin, dass in einem Bestellprozess ein Kunde seine Bestellung storniert, so handelt es sich dabei eindeutig um eine fachliche Ausnahme, die im Geschäftsprozessmodell berücksichtigt werden muss. Eine technische Ausnahme hingegen liegt vor, wenn während eines Serviceaufrufs die Kommunikation abbricht und der Aufruf zu einem späteren Zeitpunkt erneut erfolgen muss.

Im Schritt 6 wird der erzeugte Geschäftsservice im von der virtuellen Plattform unterstützten Format exportiert. Im Rahmen einer aktuellen SOA Implementierung kommen hier meist BPEL, WSDL sowie XML Schema zur Datenbeschreibung zum Einsatz. Darüber hinaus sind noch viele weitere Standards für verschiedene Aspekte verfügbar.

Im Schritt 7 wird der Geschäftsservice in die reale Ausführungsplattform⁶ importiert (Deployment). Falls diese Plattform die in der virtuellen Plattform verwendeten Standards nicht oder nicht vollständig unterstützt, muss während des Imports eine entsprechende Anpassung stattfinden. Diese Anpassungen sind unseren Erfahrungen nach immer notwendig, aber sie können mittels Generatoren und Skripten weitestgehend automatisiert werden. Wird zu einem späteren Zeitpunkt die reale Ausführungsplattform gewechselt oder soll der Geschäftsservice auf einer anderen Plattform ausgeführt werden, müssen lediglich die Generatoren und Skripte neu entwickelt werden. Der Service hingegen bleibt unberührt und ist damit plattformunabhängig.

Im Schritt 8 werden bereits existierende Services noch vervollständigt bzw. erweitert, falls dies zur Umsetzung des Geschäftsservice notwendig ist. Ein typisches Szenario für diesen Fall ist, dass die verwendeten Nachrichtenformate von Geschäftsservice und aufgerufenem Service nicht übereinstimmen. Ein weiteres Problem liegt vor, wenn der aufzurufende Service nur synchron kommunizieren kann, der Geschäftsservice aber eine asynchrone Kommunikation voraussetzt. Einige Middlewareprodukte bieten bereits Komponenten zur Lösung von solchen Konflikten. In der wissenschaftlichen Diskussion werden auch Mediatoren⁷ vorgeschlagen, die solche Konflikte automatisch erkennen und lösen.

⁶wie z. B. IBM Websphere, Oracle BPEL Process Server, SAP Netweaver, etc.

⁷z. B. WSMO Mediators - <http://www.wsmo.org/TR/d29/>

Besteht hingegen kein technischer sondern ein fachlicher Konflikt, so muss eine neue Instanz dieses Vorgehensmodells gebildet werden. Als Ergebnis erhält man eine neue Version des aufzurufenden Service, der die neuen fachlichen Anforderungen unterstützt.

Im Schritt 9 werden die Services implementiert, die bis jetzt noch gar nicht existieren. Der vorliegende Geschäftsservice inklusive seiner fachlichen und technischen Beschreibung dient dabei als Spezifikation für den neu zu entwickelnden Service. Handelt es sich bei dem zu entwickelnden Service um einen Infrastrukturservice, so wird dieser häufig mit Methoden und Vorgehensmodellen des Software Engineerings entwickelt. Handelt es sich hingegen bei dem neuen Service um einen neuen Geschäftsservice auf Basis einer Serviceorchestrierung, so muss eine neue Instanz dieses hier diskutierten Vorgehensmodells gebildet und ein entsprechendes Projekt durchgeführt werden.

Es kann davon ausgegangen werden, dass zu Beginn einer SOA Initiative die meisten Services neu entwickelt werden müssen bzw. durch bereits vorhandene Anwendungssysteme zur Verfügung gestellt werden müssen. Dies führt zu einem hohen Aufwand zu Beginn einer SOA Initiative. Später werden bei der Entwicklung neuer Geschäftsservices die bereits entwickelten Services wieder verwendet, was zu einer wesentlichen Beschleunigung des Vorgehens führt. Aus diesem Grund ist es wichtig, am Anfang einer SOA Initiative mit der Entwicklung von Geschäftsservices zu beginnen, die gleiche oder zumindest verwandte Services verwenden, damit der Wiederverwendungseffekt möglichst früh eintritt.

Im Schritt 10 erfolgt zunächst der Test, dann die Installation und anschließend die Nutzung des Geschäftsservice. Diese Schlagworte zeigen, dass es sich hier um eine hoch komplexe Aktivität handelt. So muss ein Test sowohl die korrekte Umsetzung der funktionalen wie auch der nicht-funktionalen Anforderungen überprüfen. Generell sollte versucht werden, so früh als möglich zu testen, selbst wenn noch nicht alle Anforderungen umgesetzt sind. So wird verhindert, dass schwerwiegende Fehler erst am Ende gefunden werden, deren Behebung dann zu enormen Aufwänden führt [Boe81, siehe].

Die Nutzung des erstellten Geschäftsservice ist entweder direkt oder indirekt möglich. Bei einer direkten Nutzung wird durch Auslösung des zugehörigen Geschäftsprozesses eine neue Instanz des Geschäftsservice gebildet. Erhält ein Unternehmen z. B. eine neue Bestellung, so wird mit Aufnahme der Bestellung der Geschäftsservice angestoßen. Eine indirekte Nutzung erfolgt, wenn der Geschäftsservice teil einer übergeordneten Serviceorchestrierung ist. Wird dieser übergeordnete Geschäftsservice ausgeführt, wird eine neue Instanz des Geschäftsservice gebildet, sobald der Geschäftsservice in der übergeordneten Orchestrierung aufgerufen wird.

Die Ausführung von Geschäftsservices muss überwacht werden. Durch das Monitoring kann ein Operator sich z. B. einen Überblick über alle aktuell aktiven Instanzen eines Geschäftsservice verschaffen. Dadurch können gestörte Instanzen identifiziert und entsprechende Gegenmaßnahmen ergriffen werden. Bei Ausfall von Teilen der physikalischen Infrastruktur muss ein aktives Management erfolgen, um eine geregelte Abwicklung der Geschäftsservice zu gewährleisten. Die Überwachung der Geschäftsservice kann weiterhin wertvolle Hinweise auf mögliche Verbesserungen und Optimierungspotenzial geben. Diese Information kann eine Überarbeitung des Geschäftsservice im Rahmen einer neuen Instanz dieses Vorgehensmodells erfordern.

4 Zusammenfassung

In diesem Artikel haben wir ein Vorgehensmodell für die Entwicklung von Geschäftsservicen vorgestellt. Der Hauptzweck des Vorgehensmodells liegt in der Betonung, dass ein Geschäftsservicen sich immer aus Geschäftsprozessen ableiten muss, damit er fachliche Anforderungen umsetzt. Unsere Erfahrung aus SOA Initiativen bei verschiedenen Kunden zeigt, dass SOA häufig rein technisch wahrgenommen wird. Dabei entsteht der Glaube, dass SOA ein fertiges Produkt ist, das gekauft und installiert werden kann. Unser Vorgehensmodell zeigt hingegen, dass SOA zunächst ein Managementkonzept ist. Die ersten drei Schritte unseres Vorgehensmodells können nur in Zusammenarbeit von Fachmodellierer und technischem SOA Experten geschehen. Der Aufwand zum Aufbau eines service-orientierten Prozessmodells darf dabei nicht unterschätzt werden.

In unseren Praxisprojekten hat sich das Vorgehensmodell in der Vermittlung der Idee *business-driven SOA* bewährt. Kunden haben es als Vorlage genommen, um daraus ihre individuellen Vorgehensmodelle für ihre SOA Initiativen zu bilden.

Ein zukünftiger Schwerpunkt muss auf der Integration mit anderen Methodiken im Rahmen einer SOA liegen. So ist die Beachtung einer Servicearchitekturmethodik wünschenswert. Darüber hinaus muss geprüft werden, ob weitere Eigenschaften und Erwartungen von Servicen explizit dargestellt werden können.

Literatur

- [ACD⁺03] Andrews, T., Curbera, F., Dholakia, H., Golland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., und Weerawarana, S. Business process execution language for web services (BPEL4WS) 1.1. Technischer Bericht, May 2003. [Http://www-128.ibm.com/developerworks/library/ws-bpel/](http://www-128.ibm.com/developerworks/library/ws-bpel/).
- [Bec99] Beck, K. Extreme Programming Explained: Embracing change. Addison-Wesley, 1999.
- [Boe81] Boehm, B. W. Software Engineering Economics. Prentice-Hall, 1981.
- [CCMW01] Christensen, E., Curbera, F., Meredith, G., und Weerawarana, S. Web service description language (WSDL) 1.1. Technischer Bericht, W3 Consortium, March 2001. [Http://www.w3.org/TR/wsdl](http://www.w3.org/TR/wsdl).
- [Cmm02] Capability maturity model integration version 1.1 staged representation. Technischer Bericht, CMMI Product Team, 2002. CMU/SEI-2002-TR-029.
- [Dem82] Deming, W. E. Out of the Crisis. MIT Press, 1982.
- [HS00] Herzum, P. und Sims, O. Business Component Factory. John Wiley & Sons, 2000.
- [JM05] Jones, S. und Morris, M. A methodology for service architectures. Technischer Bericht, Capgemini UK plc, London, UK, October 2005. Submitted to OASIS.
- [KHK⁺06] Köhler, J., Hauser, R., Küster, J., Ryndina, K., Vanhatalio, J., und Wahler, M. The role of visual modeling and model transformations in business-driven development. In Graph Transformation and Visual Modeling Techniques Conference (GT-VMT), Seiten 1–10. Vienna, Austria, 2006.

- [KNS92] Keller, G., Nüttgens, M., und Scheer, A. W. Semantische Prozessmodellierung auf der Grundlage Ereignisgesteuerter Prozessketten (EPK). Technischer Bericht Heft 89, Universität des Saarlandes, Saarbrücken, Germany, 1992.
- [Kru00] Kruchten, P. The Rational Unified Process: An Introduction. Addison-Wesley, Boston, USA, 2000.
- [Mit05] Mitra, T. Business-driven development. Technischer Bericht, December 2005. [Http://www-128.ibm.com/developerworks/webservices/library/ws-bdd/index.html](http://www-128.ibm.com/developerworks/webservices/library/ws-bdd/index.html).
- [MLM⁺06] MacKenzie, C. M., Laskey, K., McCabe, F., Brown, P. F., und Metz, R. Reference model for service oriented architecture 1.0. Technischer Bericht, OASIS, July 2006. [Http://www.oasis-open.org/committees/download.php/19361/soa-rm-cs.pdf](http://www.oasis-open.org/committees/download.php/19361/soa-rm-cs.pdf).
- [MSJL06] McGovern, J., Sims, O., Jain, A., und Little, M. Enterprise Service Oriented Architectures. Springer, Dordrecht, The Netherlands, 2006.
- [NPSI06] Natis, Y. V., Pezzini, M., Schulte, R. W., und Iijima, K. Predicts 2007: SOA advances. Technischer Bericht, Gartner, November 2006.
- [RH05] Rausch, A. und Höppner, S. V-Modell XT - eine Einführung. In R. Petrasch und S. Höppner, Herausgeber, Schriften zum Software-Qualitätsmanagement. Logos, Berlin, Germany, 2005.
- [Sch02] Scheer, A.-W. ARIS - Vom Geschäftsprozeß zum Anwendungssystem. Springer, Berlin, Germany, 4., durchges. Aufl. Auflage, 2002.
- [SI07] Stein, S. und Ivanov, K. EPK nach BPEL Transformation als Voraussetzung für praktische Umsetzung einer SOA. In Software Engineering Conference (SE2007). Hamburg, Germany, 2007.
- [Som04] Sommerville, I. Software Engineering. Pearson, Boston, USA, 7th Auflage, 2004.
- [Thr05] Thränert, M. Integration - Eine Begriffsbestimmung. In K.-P. Fähnrich, M. Thränert, und P. Wetzel, Herausgeber, Umsetzung von kooperativen Geschäftsprozessen auf eine internetbasierte IT-Struktur, Leipziger Beiträge zur Informatik: Band III, Seiten 11–22. Leipzig, Germany, 2005.
- [Tic01] The TickIT guide: Using ISO 9001:2000 for software quality management system. Technischer Bericht, British Standards Institute, January 2001. Version 5.0; <http://www.tickit.org/>.

”ARIS“, ”IDS“ und das Symbol ”Y“ sind eingetragene Marken der IDS Scheer AG, Saarbrücken. SAP NetWeaver ist eine Marke der SAP AG, Walldorf. Alle anderen Marken sind Eigentum ihrer jeweiligen Inhaber. Alle Rechte vorbehalten. Der Inhalt dieses Dokuments unterliegt dem Urheberrecht. Veränderungen, Kürzungen, Erweiterungen und Ergänzungen bedürfen der vorherigen schriftlichen Einwilligung durch IDS Scheer AG, Saarbrücken. Jede Vervielfältigung ist nur gestattet unter der Bedingung, dass dieser Urheberrechtsvermerk beim Vervielfältigen auf dem Dokument selbst erhalten bleibt. Jede Veröffentlichung oder jede Übersetzung bedarf der vorherigen schriftlichen Einwilligung durch IDS Scheer AG, Saarbrücken.

Copyright IDS Scheer AG, Saarbrücken, 2007

Bestandsnummer ASOA0706-D-FS