

Wann liefert eine Serviceorientierte Architektur echten Nutzen?

Dr. Jan-Peter Richter

sd&m Research
sd&m AG, software design & management
Lübecker Straße 1
22087 Hamburg
jan-peter.richter@sdm.de

Abstract: Das Schlagwort der Serviceorientierten Architektur (SOA) beherrscht gegenwärtig die Diskussion um die Gestaltung unternehmensweiter Anwendungslandschaften. Unternehmen erwarten sich von einer SOA die Reduktion von IT-Kosten und eine höhere Flexibilität ihrer Geschäftsprozesse. Der Begriff SOA an sich ist aber zu ungenau definiert, um konkrete Hilfestellungen bei der Fortentwicklung der Unternehmens-IT geben zu können. Der Beitrag benennt Voraussetzungen, um den erwarteten Nutzen zu erzielen und beleuchtet dabei sowohl technische als auch fachliche Aspekte. Wichtiger als die Einführung einer technischen Infrastruktur ist der Aufbau einer fachlichen Architektur für die Anwendungslandschaft.

1 Einleitung

Die Unternehmens-IT steht heute vor der Herausforderung, in immer kürzeren Zyklen neue Geschäftsprozesse zu unterstützen und auf Änderungen der Unternehmensstruktur flexibel zu reagieren. Gleichzeitig sieht sie sich einem Kostendruck ausgesetzt, der eine fortwährende Weiterentwicklung oder Ablösung bestehender Systeme nach klassischem Vorgehen verbietet.

Die *Serviceorientierte Architektur (SOA)* verspricht dieses Dilemma aufzulösen, indem die bestehenden Systeme über Services lose gekoppelt werden. Neue Anwendungen werden durch die Komposition von Services bestehender Anwendungsbausteine realisiert, ohne die Entwicklungskosten herkömmlicher Softwareprojekte zu verursachen.

Betrachtet man die Diskussion um Serviceorientierte Architekturen, fällt auf, dass nur wenige Autoren eine klare Definition dieses Begriffes geben. Als charakteristische Merkmale einer SOA können folgende Punkte identifiziert werden (vgl. z. B. [CHT04, Dat04, W3C04, Woo04]):

- SOA ist ein Architekturmuster, das den Aufbau einer Anwendungslandschaft aus einzelnen Anwendungssystemen beschreibt.
- Die Anwendungssysteme sind lose miteinander gekoppelt, indem sie einander ihre Funktionalitäten in Form von *Services* anbieten.
- Ein Service ist eine feste, definierte Leistung, die als Baustein eines oder mehrerer größerer Verarbeitungsabläufe verwendet werden kann. Als solcher stellt der Service eine abstrakte Sicht auf das anbietende Anwendungssystem dar und verbirgt alle Implementationsdetails. Die Definition eines Service hat den Charakter einer vertraglichen Übereinkunft zwischen Service-Anbieter und Service-Nutzer. Services sind tendenziell grobgranular.
- Services werden über einen einheitlichen Mechanismus aufgerufen, der die Anwendungssysteme plattformunabhängig miteinander verbindet und alle technischen Details der Netzkommunikation verbirgt. Der Service-Nutzer adressiert bei seinem Aufruf eine - anonyme - Schnittstelle. Der Aufrufmechanismus beinhaltet das Auffinden eines geeigneten, konkreten Service-Anbieters.

Die dargestellte Charakterisierung ist der Minimalkonsens, der sich aus der Literatur entnehmen lässt. Einzelne Autoren betonen spezifische technische Details wie z. B. das Vorhandensein maschinenlesbarer Metadaten zur Servicebeschreibung [W3C04], die von anderen Autoren aber nicht diskutiert werden.

Allen Beschreibungen in der Literatur gemeinsam sind jedoch zwei Punkte, die unmittelbare Auswirkungen auf die praktische Anwendung der SOA-Prinzipien bei der Gestaltung von Anwendungslandschaften haben:

- Der Begriff Service ist nicht ausreichend präzise eingegrenzt. Vielmehr kann fast jede beliebige Funktionalität als Service interpretiert werden. Mit dieser Beliebigkeit wächst die Gefahr, dass unpassend gewählte Services den angestrebten Nutzen in der Praxis zunichte machen.
- Die elementaren Grundgedanken der SOA sind die Trennung der Zuständigkeiten sowie die Kapselung technischer Details. Damit überträgt die SOA altbewährte Prinzipien der Softwarearchitektur aus der Domäne des einzelnen Anwendungssystems auf die Domäne der Anwendungslandschaft.

Um die eingangs genannten Ziele zu erreichen, ist daher das Grundkonzept der Serviceorientierten Architektur in vielen Aspekten zu präzisieren. Die Gestaltung der Anwendungslandschaft nach Prinzipien der SOA verlangt einerseits nach detaillierten Vorgaben hinsichtlich der technischen und fachlichen Architektur. Andererseits ist es notwendig, die Prozesse und die Kommunikationsbeziehungen zwischen den IT- und Fachabteilungen neu auf die Geschäftsprozesse des Unternehmens hin auszurichten.

Bei dieser Aufgabe greift die sd&m AG auf die langjährigen Erfahrungen bei der Beratung zur Gestaltung von Anwendungslandschaften und die Entwicklung großer unternehmenskritischer Anwendungssysteme sowie ihrer Integration in die IT-Landschaft zurück. Dieser Beitrag stellt dar, was sd&m unter Serviceorientierter Architektur versteht und welche Ergänzungen zu den bisher diskutierten SOA-Prinzipien notwendig sind, um ihre erfolgreiche Anwendung sicherzustellen.

2 Die technischen Aspekte der Serviceorientierten Architektur

Gegenwärtige Anwendungslandschaften sind typischerweise geprägt durch große, monolithische Systeme, die jeweils eine Menge von Geschäftsprozessen vollständig abdecken. Die Flexibilität dieser Systeme bestimmt die Flexibilität der Geschäftsprozesse. Die Übergänge der Geschäftsprozesse werden durch individuell gestaltete Punkt-zu-Punkt-Verbindungen zwischen den Systemen realisiert, die sie immer enger aneinander koppeln. Eine Wiederverwendung der Schnittstellen wird häufig durch den eingeschränkten Funktionsumfang sowie durch Abhängigkeiten von technischen Systeminternas verhindert. Zu geringe Kapselung der fachlichen Funktionalität sowie unklare Hoheit über die Datenobjekte tragen dazu bei, die gegenseitige Abhängigkeit weiter zu verstärken.

Die Hersteller von Infrastrukturkomponenten zum Aufbau SOA-basierter Anwendungslandschaften versprechen häufig, diese Defizite durch den Einsatz ihrer Produkte zu beseitigen. Dazu werden die fachlichen Systeme eingebettet in eine homogene technische Kommunikationsinfrastruktur, die den Aufruf von Services realisiert. In einer idealen Serviceorientierten Architektur geschieht jede Interaktion zwischen Anwendungssystemen über den Aufruf solcher Services. Dies stellt sicher, dass die Abhängigkeitsbeziehungen zwischen den Systemen kostengünstig einheitlich administriert werden können und dass die Zuverlässigkeit der Systemkopplung nicht auf ad-hoc Lösungen bei der Systemintegration beruht.

Die Vereinheitlichung des Aufrufmechanismus ist sicherlich ein wichtiger Schritt zur Entkopplung der Anwendungssysteme und zur Vereinfachung des Betriebs der IT-Landschaft. Um zu einer tragfähigen Serviceorientierten Architektur zu gelangen, ist es jedoch notwendig, zusätzliche Prinzipien zu beachten.

Zunächst muss bei der Erstellung einer SOA klar unterschieden werden zwischen den fachlichen *Enterprise Services* und den *technischen Services* innerhalb der SOA-Infrastruktur. Enterprise Services sind diejenigen Funktionalitäten, die die Abwicklung der Geschäftsprozesse des Unternehmens unmittelbar unterstützen. Sie sind die elementaren Bausteine der fachlichen Architektur der Anwendungslandschaft. Beispiele für Enterprise Services sind die Rechnungsstellung oder die Kundenstammdatenverwaltung.

Technische Services werden dagegen nur innerhalb der Infrastruktur verwendet. Ein Beispiel für einen technischen Service ist der Verzeichnisdienst für Anbieter von Enterprise Services. Technische Services und Enterprise Services befinden sich auf getrennten Abstraktionsebenen. Anwendungssysteme nutzen die technischen Services nur implizit, indem sie die Infrastruktur benutzen, um Enterprise Services aufzurufen. Daher ist es für die Serviceorientierte Architektur der Anwendungslandschaft auch irrelevant, wie im Detail die technischen Services angesprochen werden und wie ihr Zusammenspiel funktioniert. Entscheidend ist vielmehr, welche funktionalen und nicht-funktionalen Eigenschaften der SOA-Infrastruktur durch die Gesamtheit aller technischen Services realisiert werden.

Diese funktionalen und nicht-funktionalen Eigenschaften bilden die Kriterien, anhand derer die Auswahl eines Infrastrukturproduktes erfolgt bzw. die Eigenentwicklung einer solchen Infrastruktur projektiert wird. Diese Auswahl stellt in jedem SOA-Vorhaben eine besondere Herausforderung dar, da die Anforderungen an diese Eigenschaften von einem abstrakten Konzept des fachlichen Enterprise Service abgeleitet werden. Ein Beispiel soll dies illustrieren:

Im Vorfeld eines SOA-Vorhabens ist z. B. die Frage zu klären, ob der Aufruf von Enterprise Services gegen das Abhören des Datenverkehrs geschützt werden muss. Wird diese Frage mit „Ja“ beantwortet, so ist dieser Schutz damit eine Eigenschaft des abstrakten Enterprise Service. Die Eigenschaft muss durch geeignete technische Services für kryptographische Verfahren innerhalb der Infrastruktur realisiert werden.

Die wichtigsten Aspekte, die bei der Auswahl einer SOA-Infrastruktur berücksichtigt werden müssen, lassen sich wie folgt kategorisieren:

- *Funktionale Vollständigkeit des Aufrufmechanismus*: Die Bereitstellung und Inanspruchnahme von Enterprise Services kann auf verschiedenen Interaktionsmustern wie request/reply, point-to-point-message, publish/subscribe-message und ihren Varianten basieren. Die fachliche Natur des Service bestimmt dabei, welches Interaktionsmuster am besten geeignet ist. Besonders wichtig ist, dass die Infrastruktur für alle Services auch die asynchrone Nutzung ermöglicht. Dies trägt zu einer besseren Entkopplung der Systeme bei.
- *Sicherheit (Security)*: Services müssen gegen versehentliche oder böswillige Nutzung durch unberechtigte Service-Nutzer geschützt werden. Der Nachrichtenaustausch muss ggf. gegen Abhören gesichert werden. Je nach Einsatzszenario sind daher angemessene Konzepte der Authentifizierung, der Autorisierung und der Verschlüsselung umzusetzen.

- *Betriebssicherheit (Safety)*: Die Infrastruktur muss sicherstellen, dass auch im Falle von technischen Störungen eine aktive Interaktion in jedem Fall geregelt abgeschlossen wird. Abhängig von der Art des Interaktionsmusters kann sie z. B. für alle Seiten eindeutig erkennbar abgebrochen oder nach Behebung der Störung transparent fortgeführt werden. Bei idempotenten Services ist die automatische Wiederholung des Service-Aufrufs zur Behebung des Fehlers möglich.
- *Binding*: Da ein Service-Nutzer nur die Schnittstelle des verwendeten Enterprise Service kennt, muss er sich auf einen technischen Service abstützen, um in die Interaktion mit einem konkreten Service-Anbieter eintreten zu können. Beim Aufruf eines Enterprise Service wird daher zunächst eine Suchanfrage an den technischen Service gestellt. Dies geschieht für den Service-Nutzer transparent innerhalb der Infrastruktur jedoch mit einer Suchanfrage, die vom Service-Nutzer angegeben wurde. Die Suchanfrage kann neben dem Namen bzw. der ID des Service ggf. auch eine Auswahl zulässiger Service-Versionen, die erwartete Dienstgüte (*Quality of Service*) oder weitere Kriterien enthalten.

Individuelle Randbedingungen können zu unterschiedlichen Gewichtungen dieser und anderer Anforderungen führen. Einer der ersten Schritte zur Einführung einer Serviceorientierten Architektur ist daher die Definition eines abstrakten Begriffs „Enterprise Service“, der geeignet ist, die Anforderungen des Unternehmens zu tragen. Diese Aufgabe ist besonders schwierig, da die konkreten Anwendungsszenarien in der Regel noch nicht bekannt sind und viele technische Abwägungen zwischen Komplexität, funktionaler Vollständigkeit und Effizienz getroffen werden müssen. Universell „richtige“ Lösungen existieren hier nicht. Vielmehr muss anhand der individuellen Bedingungen im Unternehmen entschieden werden.

3 Prinzipien für die Anwendungsarchitektur

Die Einführung einer einheitlichen technischen Basis für die Interaktion der Anwendungssysteme und damit die Festlegung des abstrakten Enterprise Service ist eine Voraussetzung für eine Serviceorientierte Architektur. Sie ist jedoch bei weitem nicht hinreichend, um Anwendungslandschaften erfolgreich zu gestalten. Die wesentliche Aufgabe liegt vielmehr in der Entwicklung einer *Anwendungsarchitektur* für die IT-Landschaft, die den Rahmen für die Einordnung der Enterprise Services darstellt. Diese Architektur gibt sowohl die Leitlinien vor, nach denen die einzelnen Enterprise Services gestaltet werden, als auch den jeweiligen Kontext, in dem die Enterprise Services sinnvoll miteinander verknüpft werden.

Strukturierung der Anwendungslandschaft

Ein guter Ansatz für die Ausrichtung der Anwendungslandschaft an einer solchen Architektur ist die Unterscheidung zwischen einer vertikalen und einer horizontalen Strukturierungsdimension.

Vertikale Strukturierung

Die vertikale Dimension der Anwendungslandschaft charakterisiert die Anwendungssysteme hinsichtlich ihrer primären Aufgaben innerhalb der Datenverarbeitungskette (Abbildung 1).

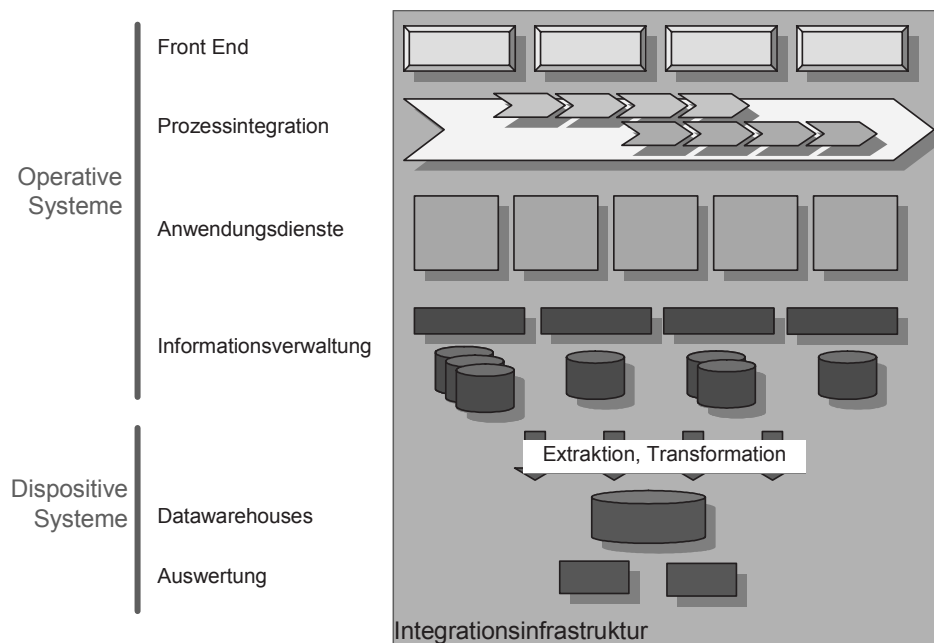


Abbildung 1: Idealisierte, vertikale Strukturierung der Anwendungslandschaft

- *Front End Systeme* dienen der visuellen Aufbereitung und Integration für Endnutzer sowie zur Unterstützung verschiedener Zugangsmedien z. B. durch Verwendung von Portal-Technologie.
- *Prozessintegrationssysteme* dienen der Steuerung von Geschäftsprozessen unter Rückgriff auf Funktionalitäten anderer Systeme.

- Systeme im Bereich der *Anwendungsdienste* realisieren in sich abgeschlossene Abläufe wie z. B. Auftragsverwaltung oder Abrechnung.
- Die Verwaltung operativer Datenbestände wie z. B. Partner, Produkt, Vertrag, und Auftrag wird in dedizierten Systemen der *Informationsverwaltung* konzentriert.
- *Dispositive Systeme* dienen nicht unmittelbar der Unterstützung wertschöpfender Geschäftsprozesse. Vielmehr bieten sie die Werkzeuge für die Entscheidungsfindung im Rahmen der Unternehmensführung. Die Aufgabe der dispositiven Systeme kann noch einmal unterteilt werden nach *Sammeln und Bereitstellen von Informationen* auf der einen Seite sowie *Auswertung* auf der anderen Seite.

Systeme in bestehenden Anwendungslandschaften integrieren typischerweise einen vertikalen Ausschnitt dieser Aufgaben für einen Bereich von Geschäftsprozessen. Eine enge Integration unterschiedlicher Aufgaben in einem Anwendungssystem ist für die Flexibilität des Unternehmens aber von Nachteil: Die Struktur der operativen Datenbestände ändert sich in der Regel nur sehr langsam. Die Steuerung der Geschäftsprozesse sowie die Gestaltung der Kommunikationskanäle zu den Anwendern müssen dagegen heutzutage schnell und kostengünstig an neue Anforderungen angepasst werden.

Im Zuge der Ausrichtung der Unternehmens-IT an den Prinzipien der Serviceorientierten Architektur sind Wartungs-, Entwicklungs- und Einführungsprojekte daher an der Leitidee der *Aufgabentrennung* auszurichten. Für bestehende und neue Systeme ist zu entscheiden, welches ihre primäre Aufgabe entlang der vertikalen Dimension ist. Die primäre Aufgabe kann beispielsweise die Informationsverwaltung für eine bestimmte Menge von Geschäftsobjekten sein. Die Variabilität bzw. Langlebigkeit der identifizierten Aufgabe bestimmt dann die folgenden Architekturentscheidungen:

- Jedes System bietet nur solche Funktionalitäten nach außen als Enterprise Services an, die seiner primären Aufgabe entsprechen.
- Die Softwarearchitektur des Systems wird so (um-)gestaltet, dass die sekundären Aufgaben im Prinzip auch von anderen Systemen erfüllt werden könnten. Wo dies bereits möglich ist, werden externe Enterprise Services genutzt.
- Falls mehrere Kandidaten für die Zuordnung einer primären Aufgabe bestehen, ist die Softwarearchitektur des Anwendungssystems so zu gestalten, dass es in lose gekoppelte Teilsysteme aufgetrennt werden kann.
- Aufwand und Kosten des Projektes werden auf die primäre Aufgabe konzentriert. Insbesondere wird hoher technologischer Aufwand nur dort aufgebracht, wo sein Nutzen durch Export der Funktionalität als primäre Anwendungsaufgabe gerechtfertigt ist.

Auf diese Weise entsteht eine Trennung von Innensicht und Außensicht der Anwendungssysteme bezüglich ihrer Einbettung in die IT-Landschaft. Die „Stärke“ jedes Systems wird Nutzen stiftend exportiert. Die sekundären Funktionalitäten bleiben dagegen gekapselt. Es gibt keine Abhängigkeiten zwischen anderen Systemen und den sekundären Funktionalitäten des betrachteten Anwendungssystems.

Die dargestellte Aufgabentrennung ist keine Vorbedingung, die vollständig umgesetzt werden muss, um Nutzen zu stiften. Je besser aber die Aufgabentrennung in der dargestellten Weise gelingt, desto übersichtlicher werden Abhängigkeitsbeziehungen in der IT-Landschaft und desto größer ihre Flexibilität. Muss z. B. ein System abgelöst werden, dessen primäre Aufgabe die Bereitstellung von Anwendungsdiensten ist, so bestehen keine Abhängigkeiten zur internen Informationsverwaltung oder zur Prozesssteuerung, da diese sekundären Aufgaben nicht exportiert wurden. Das Altsystem kann ersetzt werden durch ein neues, wenn dieses die geforderten Anwendungsdienste erbringt. Alle anderen, internen Aspekte können für das neue System frei gewählt werden. Dies reduziert die Komplexität des Ablösungsprojektes merklich.

Ein weiteres Beispiel ist die Verwendung eines Prozessintegrationssystems. In einer SOA steuert es verschiedene Geschäftsprozesse durch den Aufruf von geeigneten Enterprise Services. Ein solches System wird auch als *Orchestrierungs-* oder *Choreographiekomponente* bezeichnet. Die Geschäftsprozesse sind per Konfiguration im Prozessintegrationssystem verankert und damit ohne Austausch des eigentlichen Systems flexibel änderbar. Die damit einhergehende Vereinheitlichung der Prozesssteuerung in Entwicklung und Betrieb rechtfertigen den relativ hohen technologischen Aufwand. Eine notwendige Änderung von z. B. Anwendungsdiensten oder von Front End Funktionen anderer Systeme hat keine Auswirkungen auf das Prozessintegrationssystem.

In einer Anwendungslandschaft, die historisch um ein hochintegriertes ERP-System organisiert ist, müssen bezüglich der Aufgabentrennung Kompromisse eingegangen werden. Die Funktionalitäten des ERP-Systems in den Bereichen *Informationsverwaltung*, *Anwendungsdienste* und *Prozesssteuerung* sind dort in der Regel von so großer Bedeutung, dass keine dieser Aufgaben als primär identifiziert werden kann. Entsprechend schwieriger ist es, einzelne Aufgaben auf andere Anwendungssysteme zu verlagern oder gar das gesamte ERP-System abzulösen.

Bildung fachlicher Komponenten

Die Strukturierung der Anwendungslandschaft in der horizontalen Dimension richtet sich nach den Geschäftsprozessen des jeweiligen Unternehmens und ist daher individuell zu bestimmen. Die Geschäftsprozesse von Banken, Produktherstellern oder Logistik-Unternehmen unterscheiden sich so weit voneinander, dass eine allgemeine Darstellung hier nicht möglich ist. Abbildung 2 zeigt als vereinfachtes Beispiel die wesentlichen Geschäftsprozesse einer Kapitalanlagegesellschaft.

Die Strukturierung der Anwendungslandschaft in passende fachliche Komponenten ist eine besonders wichtige wie auch schwierige Aufgabe bei der Umsetzung einer Serviceorientierten Architektur. Umso verwunderlicher ist es, dass in der Literatur dieser Gedanke nicht näher diskutiert wird.

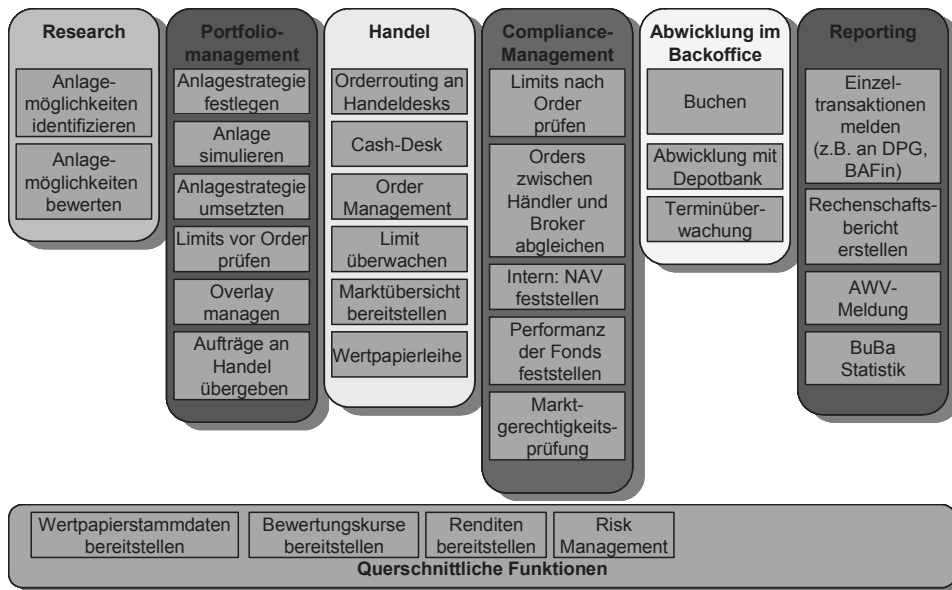


Abbildung 2: Wesentliche Geschäftsprozesse einer Kapitalanlagegesellschaft

Die Analyse der Geschäftsprozesslandschaft bildet nun den Ausgangspunkt für eine Strukturierung der IT-Landschaft in der horizontalen Dimension. Die IT-Landschaft wird in *fachliche Komponenten* aufgeteilt. Idealerweise sollte jedes technische Anwendungssystem nur einer fachlichen Komponente angehören. Dies entspricht dem Grundsatz der Trennung der Zuständigkeiten, wie es sich für die fachliche Architektur großer Anwendungssysteme bewährt hat [Sie04].

Neben den fachlichen Komponenten, deren Abgrenzung sich an den Geschäftsprozessen orientiert, ist es in der Regel notwendig, auch fachliche Querschnittskomponenten zu etablieren. Solche fachlichen Querschnittskomponenten werden z. B. für die Informationsverwaltung geschäftsprozessübergreifender Informationsobjekte benötigt. Diese fachlichen Querschnittskomponenten sind nicht mit technischen Querschnittskomponenten wie z. B. der SOA-Infrastruktur zu verwechseln.

Die fachliche Komponentenbildung ist besonders wichtig, um Fragen der Datenhoheit oder der Zuständigkeit für bestimmte fachliche Funktionalitäten zu klären. Die Entwicklung und Definition von Enterprise Services erfolgt immer vor dem Hintergrund der fachlichen Komponentenbildung. Jeder Enterprise Service ist genau einer fachlichen Komponente zugeordnet. Die Enterprise Services, die von einer fachlichen Komponente angeboten werden, werden an der Semantik ihrer Geschäftsobjekte ausgerichtet, die ebenfalls in ihren Zuständigkeitsbereich fällt.

Kriterien für gute fachliche Services

Das richtige Design fachlicher Services ist entscheidend für die Qualität einer Serviceorientierten Architektur. Wichtig dafür sind die fachliche Expertise und eine hohe methodische Erfahrung des Designers. Aus unserer Erfahrung sollte ein fachlicher Service wenn möglich die folgenden Eigenschaften haben: *grobgranular*, *technikneutral*, *idempotent*, *einzeltransaktional*, und *frei von Session-Zustand*.

- *Grobgranular*: Wenige grobgranulare, allgemeine fachliche Services sind in der Regel besser als viele spezielle, feingranulare. So ist zum Beispiel ein einziger Service „liefere alle Kundendaten zu einer Kundennummer“ vielen Services zur Lieferung einzelner Kundendaten (zum Beispiel Adresse, Bonität, etc.) vorzuziehen.
- *Technikneutral*: Ein fachlicher Service sollte nie Interna des implementierenden Systems an der Schnittstelle nach außen zeigen, zum Beispiel interne IDs eines ERP-Systems.
- *Idempotent*: Ein Service wird als *idempotent* bezeichnet, wenn ein mehrmaliger Aufruf des Service mit denselben Parametern dieselben Effekte hat wie ein einmaliger. Idempotenz wird beispielsweise in Finanztransaktionssystemen durch Einführung einer Transaktionsnummer (TAN) erzwungen. Wird eine Überweisung mit derselben TAN fälschlicherweise mehrmals an das System gesendet, so wird sie dennoch nur einmal durchgeführt.
- *Einzeltransaktional*: Jeder Aufruf eines Service ist transaktional und genügt somit den ACID-Eigenschaften. Mehrere Aufrufe eines Service sollten aber möglichst nicht mittels geschachtelter oder verteilter Transaktionen gekoppelt werden. Es ist zum Beispiel nicht empfehlenswert, einen Reservierungs- und Bezahlvorgang über eine technische Transaktion eng aneinander zu binden. Vorzuziehen ist eine lose Kopplung durch *Compensating Actions*, im Beispiel eine Stornierung der Reservierung bei nicht erfolgter Zahlung.
- *Frei von Session-Zustand*: Ein Service sollte kein Konzept einer User Session haben. Ein solcher Service kann idealerweise sowohl von einer Dialog- wie von einer Batchanwendung gerufen werden.

Diese Kriterien dienen als Leitfaden für das Design fachlicher Services. In der konkreten Aufgabe sind sie jedoch häufig im Konflikt untereinander oder mit anderen Kriterien wie Performance. So ist zum Beispiel ein grobgranularer Serviceaufruf zum Lesen aller Kundendaten inperformant, wenn nur wenige dieser Daten wirklich benötigt werden. Auch die Verwendung von Services sowohl in Online- wie in Batch-Systemen bleibt häufig ein hehres Design-Ziel, welches an mangelnder Performance scheitert.

Die Erfahrung aus dem Projektalltag zeigt, dass aufgrund von Zeit- und Budgetrestriktionen häufig vom Ziel des Designs wiederverwendbarer Services abgewichen wird. Stattdessen entstehen spezielle Services, die nur Punkt-zu-Punkt-Verbindungen zwischen Systemen erlauben. Dies ist kurzfristig kostengünstiger, verbaut aber die Möglichkeit der Wiederverwendung von Services. Diese Erfahrungen zeigen, dass die Einführung einer Serviceorientierten Architektur nicht als ein IT-Infrastrukturprojekt missverstanden werden darf. Vielmehr handelt es sich um eine strategische Entscheidung, die von der Unternehmensführung getragen werden muss. Der Aufbau der SOA involviert sowohl die IT- als auch die Fachabteilungen, die das notwendige Fachwissen bereitstellen müssen.

4 Organisatorischer Wandel im Zuge der Serviceorientierung

Die Fortentwicklung einer Unternehmens-IT in Richtung auf eine Serviceorientierte Architektur verlangt auch einen Wandel bei der IT-Organisation. So erhält die IT-Organisation die Aufgabe, das Portfolio angebotener Enterprise Services aktiv zu verwalten:

- Enterprise Services müssen umfassend und einheitlich dokumentiert werden.
- Es muss sichergestellt werden, dass angebotene Enterprise Services an den Anforderungen der Unternehmensprozesse ausgerichtet sind und unabhängig von Spezifika der Anwendungssysteme bleiben.
- Die Fortentwicklung des realisierten Portfolios muss umfassend geplant und abgestimmt werden, um Parallelentwicklungen zu vermeiden.
- Nutzungsbeziehungen und Lastvereinbarungen zwischen den Anwendungssystemen müssen dokumentiert und überwacht werden.
- Sich entwickelnde Standards sind in der Portfolioentwicklung zu berücksichtigen. Dies erleichtert in der Regel deren Nutzung und bietet die Chance, preiswerte Standardsoftware zu verwenden.

Diese Aufgaben kann die IT-Organisation nur in engster Zusammenarbeit mit den Fachabteilungen des Unternehmens bewältigen. Speziell die Spezifikation neuer Enterprise Services ist eine gemeinsame Aufgabe von IT- und Fachorganisation. Die IT-

Organisation bringt dabei als Dienstleister das Wissen um das richtige Vorgehen, die Anwendungsarchitektur sowie den Überblick über das Portfolio mit ein. Die Fachabteilung verfügt über die Expertise bezüglich des betrachteten Geschäftsprozesses und der involvierten Geschäftsobjekte.

5 Fazit

SOA ist ein derzeit viel diskutiertes Schlagwort, sowohl in der Forschung als auch in der Industrie. Dies liegt nicht zuletzt an den hohen Erwartungen, die mit der Einführung einer SOA verbunden sind: Die flexible und schnelle Einführung neuer Produkte und Dienstleistungen und die deutliche Reduktion von IT-Kosten. Dabei wird die Diskussion häufig reduziert auf die Einführung einer technischen SOA-Infrastruktur – teilweise getrieben durch Versprechungen von Infrastruktur-Anbietern.

Aus unserer langjährigen Erfahrung in der Gestaltung von IT-Landschaften großer Unternehmen und der Einbettung zentraler Systeme haben wir gelernt, dass die Einführung einer technischen SOA-Infrastruktur zwar ein notwendiger, aber nur der kleinste Schritt auf dem langen Weg zu einer Serviceorientierten Architektur ist. Entscheidend für die Einlösung der hohen Erwartungen ist der Aufbau einer Anwendungsarchitektur nach den SOA-Prinzipien. Fachliche Komponenten und Services richtig zu konzipieren, verlangt eine hohe Expertise. In diesem Beitrag werden dazu einige Kriterien beschrieben: Die horizontale und vertikale Strukturierung von Anwendungslandschaften sowie Kriterien für gute fachliche Services. Bei gewachsenen IT-Landschaften – und das ist bei den meisten großen Unternehmen der Fall – ist dies mit enormen Kosten verbunden. Notwendig sind dafür auch angemessene, häufig neue Formen der Organisation und Zusammenarbeit zwischen Fachbereichen und IT-Abteilungen.

Die Einführung einer SOA daher eine strategische Initiative, die Fachbereiche und IT-Abteilungen umfasst. Nur so kann schrittweise und langfristig der erwartete Nutzen einer SOA erzielt werden.

Literaturverzeichnis

- [CHT04] Channabasavaiah, K., Holley, K., und Tuggle, E. M.: Migrating to a service-oriented architecture, Part 1, IBM Developer Works
[<http://www-106.ibm.com/developerworks/webservices/library/ws-migratesoa>]
- [Dat04] Datz, T.: What You Need to Know About Service-Oriented Architecture, CIO Magazine, Jan. 2004 [<http://www.cio.com/archive/011504/soa.html>]
- [Sie04] Siedersleben, J.: Moderne Softwarearchitektur, dpunkt Verlag, 2004
- [W3C04] Booth, D., et. al. [Eds.]: Web Services Architecture, W3C Working Group Note, p. 61, [<http://www.w3.org/TR/ws-arch>]
- [Woo04] Woods, D.: Enterprise Services Architecture, SAP Press, 2004