

Arbeitstitel – Forum für Leipziger Promovierende // Gegründet 2009
Herausgegeben von Stephanie Garling, Enrico Thomas, Franziska Naether,
Christian Fröhlich, Felix Frey
Meine Verlag, Magdeburg

**Transformation bei Mensch und Maschine –
Parallele Umstrukturierungsprozesse in Unternehmen.
Am Beispiel der Einführung
einer Service-orientierten Architektur (SOA)**

Martin Sommer

Zitationsvorschlag: Sommer, Martin: Transformation bei Mensch und Maschine –
Pa-rallele Umstrukturierungsprozesse in Unternehmen. Am Beispiel der Einführung
einer Service-orientierten Architektur (SOA). In: Arbeitstitel – Forum für Leipziger
Promovie-rende 1 (2009). S. 1–20.

urn:nbn:de:bsz:15-qucosa2-168741

•

Abstract

– deutsch –

Die Einführung einer Service-orientierten Architektur (SOA) stellt große Anforderungen an die technischen und organisatorischen Gegebenheiten von Unternehmen. SOA wird nicht nur für ein einzelnes Projekt eingeführt, sondern soll die komplette Informationstechnologie/-system-Landschaft (IT/IS-Landschaft) nachhaltig konsolidieren. Durch die größere Komplexität gegenüber monolithischen IT-Landschaften erfordert die Einführung einen höheren Initialaufwand, der besonders in der Umstellung der Aufbau- und Ablauforganisation im Unternehmen sichtbar wird. Die SOA-Einführung muss daher strukturiert über einen Prozess gesteuert werden. Über diesen Sachverhalt herrscht in der Literatur allgemeiner Konsens. Jedoch gibt es bisher keinen Ansatz, wie der Prozess der Transformation allgemein gestaltet sein muss. Eine SOA muss auf die individuellen Bedürfnisse des Unternehmens angepasst werden. Bedingt durch diesen Aspekt existiert keine „Standard-SOA“, die als Basis für solche Überlegungen dienen könnte. In diesem Beitrag wird ein allgemeines Verständnis über diesen Prozess entwickelt, und es wird aufgezeigt, wie die durch die SOA erforderlichen Rollen in diesem Prozess agieren.

– englisch –

The introduction of a service-oriented architecture (SOA) demands great requirements for the technical and the organizational standards of enterprises. SOA is not supposed to be introduced for a single project, but shall consolidate the complete IT world with sustainable effect. Because of the greater complexity in comparison to monolithic IT-Systems, the introduction requires a bigger initial expenditure which becomes visible in particular in the conversion of the organisational structure and workflow management of the enterprise. The launch must be controlled by a process. This is agreed on in academic studies. However there is no conception, about how the transformation must be organized in general. A SOA must be designed according to the individual requirements of the business. Due to this no „Standard-SOA“ exists which could serve as a base for such considerations. In this paper I would like to give a general understanding of this process and associate the involved roles according to the transformation.

1. Einleitung

In Unternehmen werden die meisten Wertschöpfungsprozesse maßgeblich durch Informationssysteme und Informationstechnologie gestützt umgesetzt (vgl. Alisch/Winter/Arentzen 2004: 1484). Neben diesen werden aber auch nicht-wertschöpfende Prozesse wie Verwaltungs- und Administrationsprozesse durch die IT unterstützt. Die IT stellt somit eine bedeutende Rolle im Unternehmen dar. Gerade diese Abhängigkeit ist ein großes Problem, das historisch an Bedeutung gewonnen hat. Durch Fusionen und den sukzessiven Ausbau der Informationssystem-Landschaften sehen sich Unternehmen heute damit konfrontiert, aufwendig wartbare IS/IT-Landschaften zu betreiben, die darüber hinaus zu unflexibel sind, um den immer schneller auftretenden neuen Anforderungen der Unternehmen gerecht zu werden. Die IT als organisatorische Schlüsseleinheit des Unternehmens besitzt die zentrale Aufgabe, die Unternehmensstrategie in einer performanten Art und Weise durch technologische Optionen (vgl. Ferstl/Sinz 2008: 91) zu unterstützen. Die sich schnell ändernden strategischen Unternehmensziele, die hohe Marktdynamik und die Forderung nach Kostenminimierung zwingen Unternehmen dazu, die Organisation der IT-Landschaften zu überdenken, denn eine zu hohe Komplexität führt nicht nur zu einer ineffizienten Wartbarkeit, sondern auch zu einer Handlungsträgheit der IT, die sich dadurch bemerkbar gemacht hat, dass eben die neuen Anforderungen des Unternehmens nicht zeitgerecht umgesetzt werden können. Der Bedarf an Information kann nicht mehr optimal gedeckt werden (vgl. Stickel 2001: 4)..

Das zentrale Ziel der IT, die performante und reibungslose Unterstützung des Unternehmens in all seinen Abläufen, kann

nur damit erreicht werden, dass die monolithischen Systeme in transparente Komponenten aufgebrochen werden und die Verwendung dieser über anerkannte Standards erfolgt. Diese Forderung wird durch das logische Architekturkonzept der SOA verfolgt. Dadurch ist eine Wiederverwendung möglich, die zu einer effizienteren Wartbarkeit führt. Die eingesetzten Standards haben darüber hinaus den Effekt, dass neue Komponenten effizienter in die bestehende Landschaft eingebunden werden können.

Die Einführung dieser Architektur wird in der Regel durch die Reorganisation des bestehenden betrieblichen Informationssystems vollzogen. Ein betriebliches Informationssystem wird als soziotechnisches System aufgefasst (vgl. Stickel 2001: 4). Demnach wirkt sich die Reorganisation auf all seine Komponenten aus. Dies betrifft die Technologie, betriebliche Aufgaben und Abläufe und den Menschen als Benutzer.

Die Besonderheit bei der SOA-Einführung liegt in einem anspruchsvolleren Projektmanagement. Dieser Artikel verfolgt das Ziel, ein allgemein gültiges Prozessmodell zur Einführung einer SOA zu entwickeln, in dem alle Komponenten des soziotechnischen Systems gleichermaßen berücksichtigt werden. Der Fokus liegt dabei verstärkt auf der „Komponente“ Mensch. Herausgearbeitet werden soll, welche SOA-spezifischen Rollen vorhanden sein müssen und wie diese basierend auf deren Aufgabenbezogenheit den Phasen des Einführungsprozesses zugeordnet werden können. Auf Basis eines verbesserten Prozess- und Rollenverständnisses sind weitere Überlegungen in Hinsicht auf Kompetenzanalyse, -identifizierung und -adaption erst möglich.

Der Artikel sieht vor, dass zunächst unter 2. der Wandel zur SOA beschrieben wird.

Darauf aufbauend erläutert der 3. Abschnitt, wieso Kompetenzen so bedeutsam bei der Einführung von neuen Systemen und konkret SOA sind und der entstehende Bedarf durch das Projektmanagement behandelt werden muss. Im weiteren Verlauf werden die Anforderungen erörtert, die sich aus der SOA-Transformation ergeben (Abschnitt 4) und wieso ein kombiniertes Phasen- und Rollenmodell für die Transformation benötigt wird (Abschnitt 5). Im letzten Teil dieses Artikels wird dieses im Detail vorgestellt.

2. SOA –Transformation und Horizontalisierung eines betrieblichen Informationssystems

Mit Einführung einer SOA wird der Wandel von der vertikalen in die horizontale IS/IT-Architektur vollzogen. Die Anpassungszeit von informationstechnischen Komponenten auf neue Geschäftsanforderungen soll durch dieses logische Architekturkonzept minimiert werden. Zentraler Kern dieses Konzeptes ist die Bereitstellung fachlicher Dienste in Form von Services (vgl. Pulier/Taylor 2006: 51f.). Die Autoren Richter, Haller und Schrey definieren Services als Anwendungsbausteine, die lose miteinander gekoppelt sind und über einen einheitlichen Mechanismus aufgerufen werden (siehe Abbildung 1).

Dieser Kommunikationsmechanismus, auch als Enterprise Service Bus (ESB) bezeichnet (vgl. Bieberstein et al. 2006: 78), arbeitet plattformunabhängig und verbirgt die technischen Details der Kommunikation, indem der Service-Verwender den Dienst über eine anonyme Schnittstelle aufruft (vgl. Richter/Haller/Schrey 2005).

Der Wandel eines betrieblichen Informationssystems hin zur horizontalen Architektur wird in Abbildung 1 veranschaulicht.

Wie in dieser Abbildung verdeutlicht stellt der Ist-Zustand Anwendungssysteme als silohafte Anwendungen dar. Ihre Komponenten lassen sich in die Persistenz-, Funktions- und Benutzeroberflächen-Schicht organisieren. Die Integration der einzelnen Anwendungen in den Wertschöpfungsprozess wird nebensächlich betrachtet. Aus IT-Perspektive ergibt sich daraus, dass jedes Silo wenig Bezug zu angrenzenden Anwendungssystemen besitzt und damit Systembrüche verursacht. Anwendungssilos haben sich über die Jahre von ihrem Funktionsumfang weiterentwickelt und bilden eigenständige Segmente im betrieblichen Informationssystem. Im Laufe ihres Lebenszyklusses wurden teils unterschiedliche Architekturmuster verwendet, wie z. B. das hier gezeigte Model-View-Controller-Konzept.

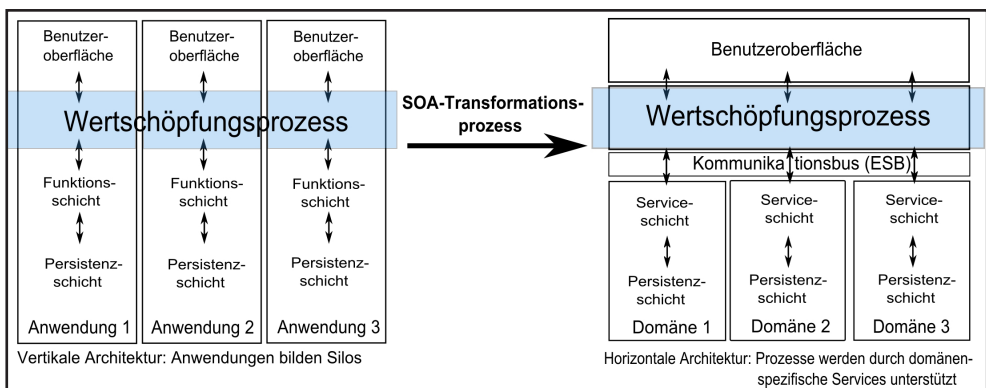


Abbildung 1: Vertikale - und Horizontale Architektur (Eigene Darstellung).

Diese unterschiedlichen Herangehensweisen, der fehlende Bezug zu angrenzenden Systemen und fehlendes Refactoring, ließen die Anwendungssysteme zunehmend intransparent werden. Ihre Gestalt nahm zunehmend die Form einer Black-Box an, deren feinste Funktionen und Abhängigkeiten unter den Subkomponenten einem Aussenstehenden nicht ersichtlich sind. Zur Folge hat dies, dass eventuell auch in anderen Bereichen des Unternehmens wieder verwendbare Funktionalitäten kaum wieder verwendet wurden. Der Wertschöpfungsprozess durchläuft in der vertikalen IT/IS-Architektur unterschiedlichste Systeme, die durch Systembrüche von einander abgegrenzt sind, bis tatsächlich ein Produkt oder eine Dienstleistung entsteht (vgl. Pleus 2008: 98).

Damit der Wertschöpfungsprozess im Mittelpunkt des unternehmerischen Handelns stehen kann, muss sich ein betriebliches Informationssystem an diesem ausrichten (siehe Abbildung 1 Soll-Architektur). Dieses Konzept wird durch die SOA verfolgt. Die Neuausrichtung des betrieblichen Informationssystems kann jedoch nicht durch vereinzelte Refaktorisierungsmaßnahmen erfolgen, sondern muss über einen Umgestaltungsprozess geschehen. In der Praxis wird dies durch Programme oder groß angelegte Projekte realisiert. Als deren Resultat entsteht eine horizontale Architektur. Wo früher Prozesse oder Teilprozesse in unterschiedlichen Programmen verarbeitet wurden und die Anwendungen im Vordergrund standen, steht nun der Wertschöpfungsprozess im Mittelpunkt. In einer SOA wird dieser durch fachliche Services aus der zuständigen Domäne bedient. Domänen stellen also für unterschiedliche Geschäftszwecke die passenden Service-Funktionalitäten bereit. Diese Services

können in anderen Domänen wieder verwendet werden.

Um die Ziel-Architektur zu erreichen gibt es zwei unterschiedliche Strategien (vgl. Müller 2006: 145–147), die gleichermaßen kombiniert werden können. Diese Ansätze haben gemein, dass als Projektergebnis die technische Infrastruktur einer SOA entsteht und die damit bisher monolithischen Systeme aufgebrochen werden. Der „leichtgewichtige“ Ansatz von beiden stellt die technischen Interessen in den Vordergrund und wird als Bottom-Up-Ansatz bezeichnet. Dahingegen ist der Umfang des Top-Down-Ansatzes wesentlich größer. Diese Vorgehensweise ist durch das Ziel geleitet ein fachlich konformes betriebliches Informationssystem zu schaffen und die IT exakt an dem Wertschöpfungsprozess auszurichten. Die Einführung einer SOA nach diesem Weg wird auch als SOA-Transformation bezeichnet (vgl. Pleus 2008: 95). Mit dieser Vorgehensweise wird die Anlehnung der IT an die Geschäftsabläufe gefördert, dieser Effekt wird auch als Business-IT-Alignment bezeichnet (vgl. Richter/Haller/Schrey 2008).

Eine SOA-Transformation hat jedoch weitreichende Konsequenzen. Die erforderlichen Umstrukturierungsmaßnahmen beziehen sich nicht nur auf die technische Ausgestaltung der Infrastruktur sondern wirken sich auf organisatorische Gegebenheiten aus. Damit entstehen mitunter erhöhte Anforderungen an das Projektmanagement. Vor allem die Veränderung der Ablauf- und Aufbauorganisation können sich negativ auf die Kultur im Unternehmen auswirken. Besonders verstärkt sich dieser Nebeneffekte, wenn im Vorfeld der SOA-Einführung die Arbeitsabläufe und die Hierarchie durch das Business-Process-Reengineering (BPR) gestrafft werden. Das Projektmanagement muss solche kulturel-

len Hürden erkennen und begegnen. Eine SOA-Transformation besitzt die typischen Projektattribute: hohe Langwierigkeit, Komplexität und Kostenintensität (vgl. Masak 2007: 2).

3. Die SOA-Transformation und der zweifache Kompetenzbedarf

Die Einführung neuer Systeme erzeugt generell den Bedarf an Kompetenzen. Blutner und Neuhaus (vgl. Blutner/Neuhaus 2007: 123) stellen die These auf, dass ein solcher Kompetenzbedarf aus zwei Gegenstandsbereichen besteht.

Der erste Gegenstandsbereich umfasst Wissen, Kenntnisse, Fertigkeiten und Fähigkeiten über das neue System. Die Abdeckung von Kompetenzen für diesen Bereich ist zwingend erforderlich, da sonst das neue System nicht eingeführt werden kann. In den SOA-Kontext übertragen bedeutet dies, dass vor allem die zu verwendenden Technologien wie auch Infrastruktur-Komponenten durch die Projektteilnehmer beherrscht werden müssen. Diese Kompetenzen sind zu diesem Zeitpunkt bereits im Unternehmen verstreut, denn bei einer SOA-Einführung wird i. d. R. auf bereits bestehende funktionale Systemkomponenten aufgebaut, in Services gekapselt und selten völlig neu implementiert. Das bedeutet das auch deren Technologien weiter verwertet werden. Das Know-How um diese liegt also bereits vor. Falls erforderlich müssen die für die Ausführung der beruflichen Tätigkeit erforderlichen Kompetenzen um SOA-spezifische Kompetenzen erweitert werden. Es gibt jedoch auch Personengruppen, deren Tätigkeiten kaum solche Kompetenz-Zusätze erfordern, beispielsweise Datenbankspezialisten. Ihre Aufgaben unterscheiden sich kaum zwischen der vertikalen und horizontalen Systemarchitektur. Ein gegensätzliches Beispiel bezieht

sich auf fachbereichsnahe Personen. Bei ihnen ist der Anteil der neu zu erwerbenden SOA-spezifischen Kompetenzen höher. Grund dafür ist eines der zentralen Anliegen der SOA-Einführung – Verbesserung des Business-IT-Alignments. Dieses Ziel wird mitunter dadurch erreicht, dass der Fachbereich zunehmend Modellierungsaufgaben übernimmt, um fachliche Dienste aus einer Domäne zu beschreiben. Damit werden die Geschäftsprozesse direkt in der IT abgebildet (vgl. Masak 2007: 11). Hier sind neben dem grundsätzlichen Verständnis über die Serviceorientierung auch konkrete fachliche Kompetenzen erforderlich. Wie beispielsweise die Beherrschung der Business Process Modeling Notation (BPMN). Mit dieser Modellierungssprache lassen sich fachliche Dienste und deren Workflows modellieren. Diese Modelle bilden später die Basis für die Serviceentwicklung und die Serviceorchestrierung. Es ist erforderlich, dass diese Modelle digital erstellt werden. Daraus ergibt sich indirekt die Anforderung über das Beherrschen von passenden Werkzeugen zur Modellierung mit der BPMN.

Der andere Gegenstandsbereich des zweifachen Kompetenzbedarfs, den Blutner und Neuhaus (vgl. Blutner/Neuhaus 2007: 123) in ihrer These erwähnen, richtet sich an das Handeln. Allgemein zählen zu dieser Dimension neben den neuen Tätigkeiten oder veränderten Arbeitsabläufen auch die Bewältigung von Problemen, die mit der Einführung von neuen Systemen verbunden sind. Ein Beispiel hierfür ist das Konflikt- und Krisenmanagement (vgl. Setzwein/Setzwein 2008: 122). Diese Kompetenzen werden in Situationen abverlangt, in denen ein geschicktes Verhalten der Projektverantwortlichen erwartet wird, um unerwartete Probleme zu lösen. Diese Situationen entstehen häufiger in IT-Projekten mit hö-

heren Aufwendungen. IT-Projekte stehen in Wechselwirkung zu Teilprojekten und anderen Bereichen des Unternehmens und werden durch sie wesentlich beeinflusst. Es entstehen Rückkopplungseffekte, die in den Aufwandschätzungen zu Beginn eines solchen Vorhabens nicht prognostiziert wurden und damit zu einer unerwarteten Aufwandssteigerung und Budgetübertretung führen können. Bei Tätigkeiten auf operativer Ebene wird der Kompetenzbedarf für Handlungen in anderen Kompetenzklassen deutlich. Dabei werden wahrscheinlich weniger die persönlichen als die fachlichen Kompetenzen abverlangt. Beispielhaft sei dies für die so wichtigen Entwurfstätigkeiten während einer SOA-Transformation demonstriert. Das Erstellen von Entwürfen ist eine wesentliche Phase für die Erstellung eines Software-Systems (vgl. Balzert 2001: 690) und dient der detaillierten Planung von technischen Einzelheiten (vgl. Abts/Mülder 2009: 304). Einer dieser bedeutenden Feinheiten ist die Sicherstellung der transaktionalen Sicherheit. Aus dem Datenbank-Bereich bekannt sorgt dieses Prinzip (auch ACID-Prinzip genannt) dafür, dass eine Transaktion als logische Einheit behandelt wird (vgl. Vossen 2000: 527). Damit wird sichergestellt, dass Änderungen in der Datenbasis stets konsistent bleiben, auch wenn die Transaktion durch einen Fehler abgebrochen wird. In einer SOA ist ein solches Prinzip ebenso wichtig. Werden die Services zur Laufzeit orchestriert, muss sichergestellt sein, dass ein unvorhergesehener Abbruch dieser Abarbeitungskette, trotz Fehler einen konsistenten Systemzustand hinterlässt. Die bedingungslose Anwendung des ACID-Prinzips in einer SOA ist jedoch nicht möglich. Zum einen können die Aufrufketten der Services sehr umfangreich sein, wodurch eine einzige Transaktionsklammer eines der elementaren Prinzipien der

Serviceorientierung (lose Kopplung (vgl. Erl 2008.: 199)) verletzen würde und zum anderen steht teilweise erst zur Laufzeit die konkrete Aufrufkette mit deren Teilnehmern und Bedingungen fest. Zu diesem konkreten Problemfall, der transaktionalen Sicherheit bei orchestrierten Services, werden von den Projektteilnehmern, die in der Entwurfsphase des SOA-Projektes mitwirken, sinnvolle Lösungen erwartet. Gefordert sind hier fachliche Kompetenzen, aber auch Kreativität. Maier et al. (vgl. Maier et al. 2008: 79) schlagen für dieses Problem die Anwendung eines Entwurfsmusters vor, das für jeden Dienst eine Undo-Funktion vorsieht, welche den ursprünglichen Zustand vor der Ausführung des Dienstes reproduzieren und somit die Konsistenz des Systemes wieder herstellen kann. Ähnlich wie dieses Beispiel lassen sich zahlreiche andere Problemfälle finden, in denen spezielle Kompetenzen aus dem SOA-Umfeld erforderlich sind.

4. Anforderungen durch die SOA-Transformation

Die SOA-Transformation beeinflusst sowohl ein bestehendes betriebliches Informationssystem, als auch die Organisation selbst. Diese kombinierten Veränderungen führen zu Wechselwirkungen die gleichermaßen hohe Anforderungen an das Projektmanagement stellt. Einerseits muss wie in jedem Softwareprojekt sichergestellt werden, dass die technischen Ziele erreicht werden. Darüber hinaus ist sicher zu stellen, dass die einhergehenden organisatorischen Restrukturierungsmaßnahmen zielführend umgesetzt werden. Dadurch, dass durch diesen Projekttypus Arbeitsabläufe und die Aufbauorganisation im Unternehmen beeinflusst werden, sind kulturelle Probleme im Unternehmen vorprogrammiert (vgl. Manes 2006: x), deren Behandlung im Projektmanagement verankert werden

muss. Wie stark die organisatorischen Veränderungen im Unternehmen tatsächlich sind kann nicht allgemein formuliert werden, sondern hängt nicht zuletzt von dem zu erzielenden Reifegrad der SOA ab. Je höher die Reife des neuen Systems, desto umfangreichere Einschnitte in die Organisation sind zu erwarten.

Die technischen Herausforderungen bei der SOA-Einführung liegen überwiegend bei der Entwicklung der Infrastruktur. Eine SOA wird wohl in den seltensten Fällen grundlegend neu entwickelt. Unternehmen, für die SOA interessant ist, sind meist mittlere und große Unternehmen (vgl. Masak 2007: 2), die bereits eine komplexe IT/IS-Landschaft besitzen. Bei diesen Unternehmensgrößen das betriebliche Informationssystem völlig neu zu entwickeln ist nicht praktikabel. Zu Beginn eines SOA-Projektes existieren daher bereits Systeme. Generell werden solche Alt-Systeme in der Wirtschaftsinformatik als Legacy-Systeme (vgl. Kurbel et al. 2008) bezeichnet. Legacy-Systeme müssen in die SOA integriert und deren Funktionalität durch Services gekapselt werden, um diese dann unternehmensweit verfügbar zu machen. Dabei existieren zwei Problemfelder. Zum einen muss verhindert werden, dass Funktionalität nicht redundant (mehrfach) im System vorliegt. Dadurch, dass die Service-Entwicklung parallel in unterschiedlichen Domänen des Unternehmens stattfinden kann, ist ein Kontrollgremium erforderlich, dass als qualitätssichernde Instanz solche Redundanzen unterbindet. Diese wird auch als SOA-Governance bezeichnet und kann durch unterschiedliche organisatorische Zusammensetzungen geformt sein. Das zweite Spannungsfeld umfaßt den optimalen Zuschnitt von Services. Die Basis für den Funktionsumfang und die Funktionsausrichtung bilden Analyseergebnisse über

die Legacy-Systeme. Für die Ermittlung des optimalen Schnitts von Services gibt es keine standardisierte Methode, jedoch gibt es Klassifizierungsversuche, wie die weit verbreitete von Erl. Er unterscheidet drei Service-Typen: die Entity-, Utiliy- und Tasks-Services (vgl. Erl 2008: 44–46). Diese Klassifikation kann als guter Anhaltspunkt dienen, um das Service-Portfolio aufzustellen und über die Domänenzugehörigkeit die Verantwortlichkeiten festzulegen. Neben dieser unternehmensweiten Sichtweise auf SOA als betriebliches Informationssystem muss auch der Servicelebenszyklus im Kleinen für den SOA-Transformationsprozess Betrachtung finden. Die Forderung nach neuer Funktionalität richtet sich nicht nur an das Problemfeld, was die passenden Services sind und wie sie bestmöglich entworfen und implementiert werden, sondern auch daran, wie sie nach Ablauf ihrer Laufzeit weiter behandelt werden. Dies ist im Sinne der Konformität der Gesamtarchitektur enorm wichtig und muss bei der Transformation auf technischer Ebene berücksichtigt werden. Sichergestellt werden kann dies nur durch eine strukturierte Vorgehensweise, die Teil des Transformationsprozesses ist.

Ebenso wie auf technischer Ebene stellen sich besondere Anforderungen auf organisatorischer Ebene. Der Mensch ist die bedeutendste „Komponente“ in jedem soziotechnischen System. Dabei gilt es zwei Gesichtspunkte zu berücksichtigen. Zum einen muss das neue System durch den Menschen bedient und zum anderen durch ihn implementiert werden. Veränderungen auf dieser Ebene müssen im Vorfeld des eigentlichen technischen Wandels vollzogen werden. Das Projektmanagement ist hier vor allem gefragt. Entscheidungen, wie neue Rollen im Unternehmen durch welche Stellen bzw. Mitarbeiter repräsentiert werden,

müssen getroffen werden. Auch der eigentliche Umbau der Ablauforganisation muss vor der technischen Transformation erfolgen. Der vorherige Abschnitt dieses Artikels hat verdeutlicht, dass die Einführung neuer Systeme generell einen zweifachen Kompetenzbedarf erzeugt. Dieser Bedarf muss durch die involvierten Mitarbeiter abgedeckt werden. Im Zuge eines strukturierten Projektmanagements müssen daher auch die Human Ressourcen auf ihre Kompetenzausstattung hin untersucht werden. Diese Überprüfung erfolgt nach dem die potentiellen Rollenträger feststehen.

5. SOA-Transformation – Notwendigkeit eines kombinierten Phasen- und Rollenmodells

Der Facettenreichtum einer SOA-Einführung macht deutlich, dass die SOA-Transformation ein komplexer Prozess ist. Damit deren Einführung zielgerecht erfolgt, ist eine strukturierte Vorgehensweise erforderlich. Angelehnt an die Problematik, dass keine standardisierte SOA existiert (vgl. Masak 2007: 10; Richter/Haller/Schrey 2005) ist demnach auch keine standardisierte Vorgehensweise für die SOA-Einführung vorhanden. Es besteht die Notwendigkeit, eine einheitliche abstrakte Sichtweise auf das SOA-Transformationsprojekt zu entwickeln, welche denen in Abschnitt 4 aufgezeigten Anforderungen gerecht wird.

Das zentrale Forschungsinteresse des Autors liegt dabei auf der Fragestellung, wie der zweifache Kompetenzbedarf identifiziert und abgedeckt werden kann. Dieser Bedarf ist nicht geschäftskritisch und daher werden die geforderten Kompetenzen nicht zu den Kernkompetenzen eines Unternehmens gezählt. Eine Nicht-Befriedigung dieses Bedürfnisses führt daher nicht direkt zur Verschlechterung der Marktposition eines Unternehmens. Der Bedarf

bezieht sich auf operative Tätigkeiten und liegt damit im Rahmen des Skillmanagements (vgl. Kunzmann 2005: 21). Der naive Versuch, auf Basis der Sachlage die Methoden des Skillsmanagements anzuwenden, ist nicht praktikabel. Damit die Kompetenzen operationalisiert werden können ist zunächst ein Phasenmodell erforderlich, um die notwendigen Aufgaben und Tätigkeiten in den Phasen zu organisieren. Erst auf Basis dessen lassen sich die involvierten Rollen identifizieren. Die Kombination des Phasen- und Rollenmodells bildet die Basis für die Beantwortung der weiteren Forschungsfragen. Unter anderem soll geklärt werden, welche beruflichen Handlungskompetenzen nötig sind, um die für die SOA-Transformation erforderlichen Aufgaben angemessen umzusetzen. Als Ergebnis dieser Untersuchungen kann ein Kompetenzmodell für diesen Projekttypus entwickelt werden, dessen Einsatzbereich sehr weitreichend ist. Im Sinne des Skillsmanagements können Kompetenzdefizite identifiziert und behandelt werden. Darüber hinaus sind Aussagen über die Eignung eines Mitarbeiters für die vorgesehene Rolle möglich und schlussendlich kann auch die Eignung der Teams sowie die Auswirkung der Integration eines Mitarbeiters in ein solches untersucht werden.

Ein Modell bildet stets nur auf abstrakte Art und Weise einen Teil der Realität ab (vgl. Mertens et al. 2001: 311). Dieser Informationsverlust ist auch eine Eigenschaft von Rollen- und Phasenmodellen. Eines auf alle individuellen Bedürfnisse zugeschnittenes Modell der Unternehmen, die eine SOA fachlich getrieben einführen wollen, zu entwickeln, ist nicht realistisch. Deshalb können bestimmte Varianzen bezüglich der Modell- und Realwelt auftreten. Als Beispiel dafür wäre die Reihenfolge der einzelnen Projektphasen zu nennen. Vor-

stellbar ist auch, dass Phasen in einzelnen Projekten nicht erforderlich sind. Auch kann das Verständnis der Projektrollen von Unternehmen zu Unternehmen variieren. Neben der differierenden Benennung können sich darüber hinaus Unterschiede in den zu leistenden Aufgaben ergeben.

6. Kombiniertes Phasen- und Rollenmodell

Das Phasenmodell untergliedert sich in fünf Phasen. Diese decken sowohl den SOA-Systemlebenszyklus als auch den Lebenszyklus von Services (Service-Lifecycle)

die kontinuierliche Weiterentwicklung des Systems, genauer der Services, bis der angestrebte Systemreifegrad erreicht ist, wie es die Roadmap vorsieht. Zum Erreichen des nächsten Reifegrades würde der Transformationsprozess wiederholt werden.

Die Realisierung der Aufgaben in der zweiten Hauptkomponente erfolgt parallel. Die Domänen, die an dem Projekt beteiligt sind, veranlassen gleichzeitig die Service-Realisierung, was zu weiteren Schwierigkeiten führt. Eines der obersten Gebote der Serviceorientierung verlangt die Redundanzfreiheit und die angemessene Granularität

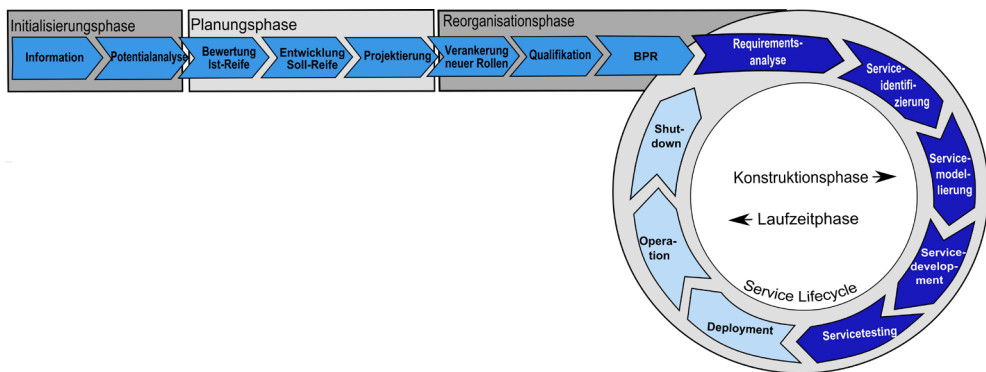


Abbildung 2: Phasenmodell des SOA-Transformationsprozesses (Eigene Darstellung)

ab. Abbildung 2 veranschaulicht schematisch den Aufbau. Die erste Hauptkomponente ist die eigentliche Vorbereitung auf ein SOA-Vorhaben. Sie besteht aus der Initialisierungsphase, der Planungsphase und der Reorganisationsphase. Am zeitlichen Ende dieser Vorbereitungen folgt die technische Realisierung. Diese stellt die zweite Hauptkomponente im Konstrukt dar, die aus der Konstruktions- und Laufzeitphase besteht. In ihr werden u. a. die Services identifiziert, entworfen, implementiert und getestet, woraufhin sie im System für ihre Verwendung verfügbar gemacht werden. Die zweite Hauptkomponente trägt die Gestalt eines Zyklus. Dieser repräsentiert

der Dienste. Diese Prinzipien müssen bereits bei der Service-Identifizierung und auch beim Service-Design berücksichtigt werden. Parallele Subprojekte bedürfen einer besonderen Überwachung. Das Kontrollorgan, die SOA-Governance, stellt genau dieses sicher und ist die Qualitätssicherungsinstanz in einem SOA-Vorhaben. Über ihre Notwendigkeit, besteht in der Literatur vollstes Einverständnis.

Im Folgenden werden die Phasen der SOA-Transformation näher erläutert. Das zentrale Ziel ist dabei die Erzeugung von einem einheitlichen Verständnis über die Transformation und über das Zusammenspiel

von Rollen und deren Aufgaben in diesem Prozess, um in späteren Arbeiten darauf aufbauend die erforderlichen Kompetenzen zu identifizieren. Bezüglich der aufgezählten Tätigkeiten wird kein Anspruch auf Vollständigkeit erhoben.

6.1 Initialisierungsphase

In der Initialisierungsphase werden wichtige Vorüberlegungen angestellt, die im weiteren Verlauf bewertet werden. Diese Phase stiftet dem Unternehmen primär keinen Nutzen, sondern evaluiert, ob die SOA dem Unternehmen einen Nutzen erbringt. Die Initialisierungsphase besteht aus zwei Aktivitäten: der Sammlung von Information und der Durchführung der Potentialanalyse.

6.1.1 Informationsphase

Die Informationsphase umfasst die grundlegende Verdeutlichung des Themas SOA. Dies kann auf unterschiedlichen Wegen geschehen. Die Quelle für die Informationsaneignung hat dabei keine primäre Bedeutung. Ziel dieser Phase ist ein Grundverständnis aufzubauen, das in den folgenden Phasen eine solide Entscheidungsgrundlage bildet. Neben der Wissensaneignung über die technischen und konzeptionellen Aspekte von SOA, wird auch Wissen über die Vorgehensweise der Einführung und die Produkte der Hersteller gewonnen. Wichtig ist, dass die Service-orientierten Prinzipien verstanden worden sind. Diese Phase bildet gleichermaßen den Ausgangspunkt für das SOA-Programm (vgl. Pleus 2008: 98).

Im Wesentlichen sind in diesen Abschnitten die Rollen der Geschäftsführung, des Projektleiters und des SOA-Architekten involviert. Besonders der SOA-Architekt erlernt in diesem Schritt fundamentales und detailliertes Wissen, was er in den weiteren Phasen durch seine beratende Tätigkeit an-

wenden muss. Zu empfehlen ist, dass diese Rolle durch einen erfahrenen Mitarbeiter aus dem Architektur-Umfeld besetzt wird. Das Wissen des Projektmanagers und der Unternehmensführung ist dagegen nicht durch jene Detailtiefe geprägt. Jedoch müssen die grundlegenden Prinzipien der Serviceorientierung verstanden worden sein.

6.1.2 Potentialanalyse

Unter Einbeziehung der Geschäfts- und IT-Strategie sowie der Prozess- und Anwendungslandschaft werden die Potentiale und Risiken für eine SOA-Einführung bewertet. Dabei wird die vorhandene Infrastruktur zunächst evaluiert. Neben den Geschäftsprozessen werden auch die organisatorischen Strukturen untersucht, die eine SOA unterstützen soll (vgl. Kontogiannis/Lewis/Smith 2008: 4; Pleus 2008: 98). Stellvertretend wird bei diesen Analysen der Fachbereich durch den Business Process Architect vertreten, der über genügend globales Wissen aus dem Bereich der Geschäftsprozesse verfügt. Unterstützt wird er durch den SOA-Architekten. Sein kombiniertes Wissen über die vorhandene IT/IS-Landschaft und über die Serviceorientierung bildet die Basis für technische Fragestellungen. Ziel dieser Phase ist die Bewertung des Mehrwertes und der Verbesserung des Business-IT-Alignments, das wohl den größten Nutzen einer SOA darstellt. Sollte die Einführung einer SOA dem Unternehmen keinen Mehrwert stiften, wird das Projekt an dieser Stelle durch die Geschäftsführung beendet (vgl. Kontogiannis/Lewis/Smith 2008: 4; Pleus 2008: 98).

Ein Projektende bringt auch den Abbruch des Transformationsprozesses mit sich. Das kann beispielsweise der Fall sein, wenn die SOA das Business-IT-Alignment nicht nennenswert optimiert. Alternativ könnte sie als Integrationsansatz begriffen werden

und durch den weniger aufwendigen Button-Up Ansatz eingeführt werden.

6.2 Planungsphase

Die Planungsphase beinhaltet konstruktive Tätigkeiten rund um die Planung, um die SOA-Transformation voran zu treiben. Sie wird nach erfolgreichem Abschluss der Initialisierungsphase eingeleitet.

6.2.1 Bewertung der Ist-Reife

In dieser Phase wird unter Anwendung eines SOA-Reifegrad-Modells die aktuelle Reife des Unternehmens in Hinblick auf die SOA ermittelt. Der Prozess ist wichtig, damit später über einen Soll-Ist-Vergleich die konkrete Umsetzung abgeleitet werden kann. In dieser Phase sind bereits weiterführende Kenntnisse im SOA-Zusammenhang erforderlich. Neben den Entscheidungsträgern ist auch der SOA-Architekt für diese Phase enorm wichtig. Letzterer arbeitet erneut eng mit dem Business Process Engineer zusammen. Die Ermittlung des bestehenden SOA-Reifegrads beinhaltet eine Analyse der bestehenden IT-Systeme, Geschäftsprozesse und Organisationsstrukturen. Da sich betriebliche Informationssysteme und damit auch SOA an der Geschäftsstrategie ausrichten, sollten ebenfalls die strategischen Elemente mit in die Untersuchung einfließen (vgl. Hildebrandt/Wolf 2009: 31; Pleus 2008: 99).

6.2.2 Entwicklung der Soll-Reife

Die Abstufung der entsprechenden Reifegrade kann sich zwischen den verschiedenen SOA-Reifegrad-Modellen unterscheiden. Daher muss für diese Phase das selbige Modell verwendet werden wie auch bei der Bestimmung der aktuellen Reife im Teilschritt zuvor. Der gewünschte Reifegrad wird sowohl in technischer als auch in organisatorischer Sicht entwickelt (vgl.

Pleus 2008: 98). Ziel ist es, möglichst genau zu determinieren, wie der Zustand des Unternehmens nach dem Transformationsprozess gestaltet sein soll. Durch den Reifegrad wird maßgeblich der Automatisierungsgrad der SOA festgelegt. Je höher der Reifegrad, desto agiler ist die IT-Landschaft im Unternehmen. In der höchsten, teilweise fiktiv anmutenden, Stufe der SOA-Reifegradmodelle, werden Dienste dynamisch in Geschäftsprozesse eingebunden. Dabei werden sowohl interne als auch externe Dienste verwendet. Auf dieser Stufe können Dienste die Vertragsbedingungen für Kontrakte, die sie untereinander eingehen, automatisch verhandeln. Dagegen sind in niedrigeren Reifegraden nicht alle Konzepte der Serviceorientierung erforderlich. In dieser Phase finden die selben Rollen Anwendung wie bei der Bewertung der Ist-Reife.

6.2.3. Projektierungsphase

Die Projektierungsphase ist der komplexeste und bedeutungsvollste Schritt in der Planungsphase, denn ihr Resultat beeinflusst maßgeblich den Projekterfolg und die Qualität der Ziel-Architektur. In dieser Phase wird der genauere Rahmen für die SOA-tangierenden Organisationsbereiche festgelegt (Systeme, Organisationseinheiten). Die Entwicklung eines Projektplans, der die einzelnen Schritte bis zum Erreichen des Zielreifegrades festlegt, ist elementar. In diesem werden konkrete Kriterien festgelegt, anhand derer das Erreichen von Meilensteinen gemessen werden kann (vgl. Pleus 2008: 98). In dieser Phase kommen die beratenden Rollen aus IT, fachbereichsnaher IT und Fachbereich zum Einsatz. Neben dem Business Process Architect und dem SOA-Architekten tritt nun erstmalig die Rolle des IT-Architekten in Erscheinung. Der IT-Architekt ist für

technische Problemlösungen verantwortlich.

Darüber hinaus wird in dieser Phase die SOA-Governance etabliert. Dieses Kontrollgremium sorgt dafür, dass stets die erforderlichen Richtlinien, Standards und Prozeduren eingehalten werden. Aufgaben aus diesem Umfeld können sehr umfangreich werden. Daher empfiehlt es sich eigens für diese Zwecke eine eigene Rolle einzuführen – den SOA-Gouverneur. Nachdem der genaue Fahrplan festgelegt worden ist, wird das Projektteam ausgewählt. Dies sollte objektiv, strukturiert und standardisiert auf Basis eines SOA-Kompetenzmodells geschehen, dass die Mitarbeiter getreu dem Skillsmanagement bewertet und mögliche Kompetenzlücken identifiziert. Diese werden in der nachfolgenden Phase geschlossen. Darüber hinaus wird in dieser Phase auch die Entscheidung über den oder die partizipierenden Dienstleister getroffen. Idealerweise sollte das Team so gewählt werden, dass die fachlichen SOA-Skills sowohl auf Seiten des Unternehmens als auch im Blick auf den Dienstleister mindestens komplementär sind. Mit Hilfe des Kompetenzmodells sollten ebenfalls Aussagen über das Teampotential herausgearbeitet werden können. Nach dieser Phase sind alle Planungsaktivitäten abgeschlossen und es wird mit der konkreten Umsetzung begonnen.

6.3 Reorganisationsphase

SOA erfordert neue Denkweisen. Das neue Rollen im Unternehmen gebraucht werden, die sowohl für die Konstruktion als auch den Betrieb verantwortlich sind, darüber besteht allgemeiner Konsens in der Fachliteratur. Die Reorganisationsphase wird durch das Management im Unternehmen getrieben. Ständiger Begleiter ist dabei die SOA-Governance durch ihre Vertreter.

6.3.1. Verankerung neuer Rollen

In dieser Phase werden den Projektkandidaten die Rollen zugewiesen. Das Management und die SOA-Governance sind die beteiligten Steuerungsgremien. Im Unternehmen werden nur jene Rollen verankert, die auch tatsächlich durch interne Ressourcen erbracht werden. Durch die Übertragung der neuen Rollen an die Mitarbeiter wird ein wichtiger Schritt in Richtung Bewältigung der kulturellen Hürden genommen. Den Mitarbeitern wird dadurch bewusst, dass die Serviceorientierung eine neue Evolutionsstufe ist und andere Aufgaben, Aktivitäten erfordert und neue Entscheidungssituationen abverlangt. Die Verankerung neuer Rollen ist der wichtigste Schritt zur Integration des Paradigmas in der Aufbauorganisation. Der Personalverantwortliche steuert diesen Prozess und bekommt von der Geschäftsführung ggf. Unterstützung.

6.3.2 Qualifikation

Nicht alle Tätigkeiten im Unternehmen werden durch die Einführung der SOA berührt. Gleichwohl müssen die Kompetenzlücken der Mitarbeiter, die an diesem Projekt beteiligt sind unbedingt geschlossen werden. Bevor die eigentliche Umstrukturierung im Unternehmen beginnt, müssen daher alle Projektbeteiligten für das Projekt ausgebildet sein. Diese Phase behandelt daher die Aufbereitung der Mitarbeiterqualifikation in Richtung SOA-Dekade. Schulungen, Zertifizierungen und andere Weiterbildungen sind erforderlich. Nicht nur die Projektmitglieder, sondern auch die Mitarbeiter, die im späteren produktiven Betrieb, für die Unterstützung sorgen oder mit dem System unter anderen Bedingungen arbeiten müssen, werden in diesem Vorgang berücksichtigt. Der Personalverantwortliche steuert diese Aktionen und

überprüft mit geeigneten Methoden den Erfolg der durchgeführten Maßnahmen.

6.3.3 Business Process Reengineering (BPR)

Nachdem in der Qualifikationsphase die Kompetenzlücken bei den Projektteilnehmern geschlossen wurden, ist die Basis geschaffen, um konform nach dem Paradigma der Serviceorientierung die vorhandenen Geschäftsprozesse zu reorganisieren. Dieser Schritt bezieht sich auf die Ablauforganisation des Unternehmens. Gemäß entsprechender Verfahren werden die Geschäftsprozesse an die veränderten Rahmenbedingungen angepaßt. Dieser Prozess lässt Spielraum für Umstrukturierungen der Geschäftsprozesse zu. Neben den wertschöpfenden Geschäftsprozessen, werden auch interne Arbeitsabläufe, wie z. B. die der IT-Abteilung, neu strukturiert. Darüber hinaus wird die Aufbauorganisation im Sinne des BPR-Ansatzes feiner geformt. Diese Phase schließt die Reorganisationsphase ab. Betreut wird dieser organisatorische Entwicklungsabschnitt durch den Business Process Architekten, den SOA-Architekten und den SOA-Gouverneur.

6.4 Konstruktionsphase

In der Konstruktionsphase wird die technische Infrastruktur für die SOA geschaffen. Sowohl die Konstruktionsphase als auch die Betriebsphase können in ihren Einzelschritten parallel durchlaufen. Dies erfolgt im Sinne der Serviceorientierung entsprechend in den teilnehmenden fachlichen Domänen.

6.4.1 Requirements-Analyse

Mit dem Auftrag einer Fachabteilung für die konkrete Umsetzung von Services wird zunächst mit dem Requirements Engineer und dem Domain Expert eine detaillierte

Beschreibung von funktionalen und nicht funktionalen Anforderungen erzeugt. Der Domain Expert wird von fachlicher Seite durch den Business Process Architect unterstützt. Neben dem Lastenheft sollten auch alle Fachbegriffe definiert und ein Prozessmodell erstellt werden (vgl. Hildebrandt/Wolf 2009: 30 f.). Zu diesem Zeitpunkt empfiehlt es sich, die konkreten Testfälle und Testszenarien ausführlich und verbindlich festzulegen. Für die in diesem Zusammenhang anfallenden Aufgaben sind der SOA-Architekt und der SOA-Gouverneur verantwortlich.

6.4.2 Serviceidentifizierung und Serviceanalyse

Das Hauptziel dieses Schrittes im Sinne der IT ist die Vermeidung von Redundanzen. Aus Sicht der Betriebswirtschaft stellen Kosteneinsparungen durch verringerte Aufwendungen im Bereich der Entwicklung einen großen Anreiz dar.

Mit dem fertiggestellten Lastenheft ist der SOA-Architekt in der Lage, Kandidaten für die Service-Operationen zu identifizieren. Dies geschieht entweder auf Basis von Legacy-Anwendungen oder einem bereits verfügbaren Service-Repository. Um Redundanzen zu vermeiden, muss das bestehende Service Repository auf eventuell bestehende Services untersucht werden, welche die gleiche Funktionalität aufweisen. Hierzu arbeitet der SOA-Architekt mit dem Repository-Verwalter zusammen. Der Repository-Verwalter ist eine administrative Rolle zur Entwicklung und Pflege der so genannten Service-Registry. Dieser Mechanismus ist ein zentraler Bestandteil einer SOA, der Services publiziert, auffindbar macht und damit dem Aufrufer ermöglicht diese zu binden (vgl. Bieberstein et al. 2006: 215). Sollte kein Service vorhanden sein oder eine Erweiterung von bereits be-

stehenden Services nicht in Frage kommen, dann wird ein neuer Service entwickelt. Dies geschieht über die folgenden Phasen. In diesem Fall wird dieser Service sofort einem Service Owner zugeordnet. Dieser trägt seitens des Fachbereiches die Verantwortung für den Service. Sein Pendant auf technischer Seite ist der Technical Service Owner. Die Zuständigkeit beider Rollen endet erst mit dem Ende des Service-Lebenszyklus. Zum Abschluss der Phase werden alle vorhandenen Artefakte freigegeben. Ein wichtiger Bestandteil ist auch die Definition der Service-Level-Agreements (SLA). Der SOA-Gouverneur prüft in dieser Phase, ob alle Governance-Richtlinien eingehalten worden sind (vgl. Hildebrandt/Wolf 2009: 31). Der Business Process Architect nimmt seitens des Fachbereiches nur eine beratende Funktion ein.

6.4.3 Servicedesign und -modellierung

In dieser Phase wird eine ausführliche Service-Spezifikation und ein Service-Modell entwickelt. Die Service-Spezifikation sollte neben den funktionalen Anforderungen auch die nicht-funktionalen Anforderungen wie z. B. Service-Level-Agreements umfassen. Werden in dieser Phase bereits parallel die Service-Konsumenten entwickelt, so ist darauf zu achten, dass Änderungen entsprechend publiziert werden. Sollte die modellierte Serviceschnittstelle bereits veröffentlicht worden sein, so ist eine inkompatible Änderung nur über Versionierungsmechanismen zu lösen. Hildebrandt und Wolf (vgl. Hildebrandt/Wolf 2009: 31) schlagen in diesem Zusammenhang vor, ein pragmatisches und kein perfektes Design anzustreben. Änderungen werden weniger durch schlechtes Design sondern vielmehr durch sich ändernde fachliche Anforderungen angestoßen. Der Service-Modellierer muss bei der Modellierung der WSDL-Dokumente und XML-Schemado-

kumente darauf achten, dass vor allem die Richtlinien zur Versionierung eingehalten werden. Der SOA-Architekt prüft am Ende, ob das Design mit der Referenzarchitektur konform ist, zur IT-Strategie passt und die nicht-funktionalen Anforderungen mit in das Design eingeflossen sind. Diese Aufgaben werden in Kooperation mit dem SOA-Gouverneur durchgeführt. Für technische Detailfragen und Probleme bei dem Design wird der IT-Architekt zu Rate gezogen. Auf Seiten des Fachbereichs hingegen sind der Business Process Architect und der Domain Experte die Ansprechpartner. Weitere wesentliche Bestandteile sind Datenmodelle. Diese fließen letztendlich mit in die erforderlichen Migrationsschritte ein.

6.4.4 Serviceentwicklung

Nach der Freigabe des Designs beginnt der Service-Entwickler mit dem internen Design des Services und der Umsetzung mit der entsprechenden Technologie. Der Service-Entwickler ist ebenfalls für die Erstellung der Unit-Tests verantwortlich, um den Service auf sein technisches Verhalten hin zu überprüfen. Im Rahmen der Serviceentwicklung wird auch das Betriebs- und Notfallkonzept erstellt. Als Artefakte entstehen demnach in dieser Phase der implementierte Service, die Betriebsdokumentation, das Notfallkonzept und die entsprechenden Unit-Tests (vgl. Hildebrandt/Wolf 2009: 31). Bei der Service-Entwicklung werden bereits Teile der SOA-Infrastruktur verwendet. Rollen, mit denen der Entwickler daher zusammenarbeiten muss, sind die Verantwortlichen für SOA-Infrastruktur, Persistenz-Schicht und Registry. Probleme die bei der Entwicklung im Zusammenhang der Gesamtarchitektur entstehen, werden mit dem jeweils zuständigen Architekten gelöst.

6.4.5 Servicetesting

Nach fertiger Implementierung und Freigabe gehen die Services in die Testphase über. Die Soll- und Ist-Werte werden überprüft, ausgewertet und dokumentiert. Die Integrationstests stellen sicher, dass die vorhandenen Services mit den neu entwickelten interoperabel sind. Sie und die Komponententests sollten automatisch ablaufen. Systemtests und User-Acceptance-Tests werden manuell von einem Tester durchgeführt. Dieser muss über entsprechendes Know-How aus dem Bereich verfügen. Getestet wird auch die Orchestrierung der Services in einem Geschäftsprozess. Produktiv laufende Services mit Fehlern durchlaufen einen erneuten Konsolidierungsprozess. Der Testprozess wird durch die Abnahme des Service Owners abgeschlossen. Als Resultat wird der Service für den produktiven Betrieb freigegeben und durch den Service-Registry-Verantwortlichen im System verfügbar gemacht (Deployment)(vgl. Hildebrandt/Wolf 2009: 32). Das Servicetesting wird hauptsächlich durch den Service-Tester und in kleinen Anteilen durch den Service-Entwickler durchgeführt.

6.5 Laufzeitphase

Die Laufzeitphase beinhaltet den produktiven Betrieb der SOA. Services werden nun über die Registry gesucht, gefunden und nach Bedarf verwendet. Die Laufzeitphase stellt besondere Anforderungen an die Verwaltung und Überwachung von Services.

6.5.1 Deployment

Wurde die Testphase erfolgreich durchlaufen, erfolgt das Deployment (Veröffentlichen) des Services durch den Registry-Verantwortlichen. Der Service wird dazu in die Registry aufgenommen und die beteiligten Verantwortlichen aller betroffenen Komponenten werden durch den Registry-

Verwalter über diesen Vorgang informiert. Weitere wichtige Aufgaben sind die Konfigurierung des Monitorings und des Reportings der SLA's sowie die Überprüfung der Betriebsdokumentation und des Notfallplans (vgl. Hildebrandt/Wolf 2009: 32). Diese Aufgaben werden durch den SOA-Systemadministrator übernommen. Durch die Abnahme wird bestätigt, dass der Service korrekt im Orchestrierungsverbund integriert ist und für den produktiven Einsatz zur Verfügung steht.

6.5.2 Operating

Zeitlich betrachtet folgt nun die längste aller Phasen des Service-Lifecycles: der Betrieb. Der SOA-Administrator verfolgt den laufenden Betrieb und überwacht das korrekte Einhalten der SLAs und der Key Performance Indikatoren (KPI) (vgl. Hildebrandt/Wolf 2009: 33). Die Überschreitung von Schwellenwerten wird gemäß dem vorher definierten Notfallplan behandelt. Logging und Monitoring bilden eine solide Datenbasis, um Erkenntnisse für Optimierungen von Services zu gewinnen. Ausgewählte Erkenntnisse müssen den Fachbereich erreichen und die Grundlage für die Weiterentwicklung von Services bilden. Der SOA-Administrator überwacht neben den Services auch das Laufzeitverhalten der Plattform sowie die sicherheitsrelevanten Eigenschaften des Systems. Diese Rolle sorgt ebenfalls dafür, dass das System entsprechend den Anforderungen skaliert ist.

6.5.3 Shutdown

Es liegt in der Natur der Dinge, dass sich Anforderungen an Dienste ändern. Ausgehend von der Versionierungsstrategie werden nach unterschiedlichen Gesichtspunkten die Services versioniert. Werden durch neue Anforderungen des Fachbereichs oder durch technische Probleme neue Versio-

nen von Services erzeugt, muss sichergestellt werden, dass alte Versionen aus dem System entfernt werden. Dies ist eine unabdingbare Forderung, um die Wartbarkeit des Systems sicher zu stellen (vgl. Hildebrandt/Wolf 2009: 33). Die Verpflichtung hierfür trägt der Registry-Verantwortliche, der die für das Abschalten erforderlichen Richtlinien der SOA-Governance befolgt. Diese geben auch vor, in welcher Form die Service-Konsumenten über diesen Schritt benachrichtigt werden.

6.6 Überblick Rollen

Die nachfolgende Tabelle (S. 18) veranschaulicht zusammenfassend die Aktivitäten der Rollen in den entsprechenden Phasen des SOA-Transformationsprozesses. Jene Aktivitäten sind durch ein Kreuz gekennzeichnet.

7. Fazit und Ausblick

Das Architekturkonzept SOA stellt einen Lösungsansatz dar, wie ein heterogenes und monolithisch geformtes betriebliches Informationssystem restrukturiert werden kann, um eine wendigere IT/IS-Landschaft zu schaffen. Durch SOA wird das Business-IT-Alignment gefördert. Damit wachsen die Geschäftsprozesse und die IT näher zusammen und verleihen aufgrund dieser Tatsache dem Unternehmen eine höhere Dynamik bei der Anpassung von Geschäftsprozessen. Der Einführungsprozess ist langwierig und durch die Veränderungen auf technischer, als auch organisatorischer Ebene kompliziert. Beide Abstraktionsebenen müssen gleichsam im Einführungsprozess integrativ behandelt werden. Darüber hinaus entsteht durch das neue System ein zweifacher Kompetenzbedarf, der durch die Projektmitglieder abgedeckt werden muss. Im Fokus der Forschungsinteressen des Autors liegen

die Kompetenzen der Mitarbeiter. Damit diese operationalisiert werden können, ist ein allgemeines Verständnis über den SOA-Transformationsprozess mit seinen Aufgaben und den involvierten Rollen erforderlich. Dieses Verständnis wurde über das kombinierte Phasen- und Rollenmodell in diesem Artikel entwickelt.

Kompetenzen erstrecken sich auf fachliche, methodische, soziale und persönliche Eigenschaften eines Menschen (vgl. North/Reinhardt 2005: 2). Welche der Eigenschaften durch die jeweilige Rolle im SOA-Transformationsprozess abgedeckt werden muss, ist zentrales Ziel der nachgelagerten Forschungstätigkeit. Inwieweit in der Praxis die erforderliche SOA-Kompetenzausstattung berücksichtigt wird und wie ggf. die Kompetenzen strukturiert behandelt werden, wird durch eine empirische Studie geklärt. In darauf folgenden Arbeiten muss als Basis der Operationalisierung ein Kompetenzkatalog für diesen Kontext ausgearbeitet werden, mit dessen Hilfe Methoden des Skillsmanagement angewendet werden können.

	Information	Potentialanalyse	Bewertung Ist-Reife	Entwicklung Soll-Reife	Projektierungsphase	Verankerung neuer Rollen	Qualifikation	Business Process Reengineering	Requirementsanalyse	Serviceidentifizierung und -Analyse	Service design und -modellierung	Serviceentwicklung	Service Testing	Service Deployment	Operation, Manage	Shutdown
Domain Expert									X		X					
Service Owner										X	X	X	X	X	X	X
Business Process Architect		X	X	X	X			X	X	X	X					
Requirements Engineer									X							
SOA-Architekt	X	X	X	X	X			X	X	X	X	X	X			
SOA-Gouverneur					X	X	X	X	X	X	X	X	X	X	X	X
Service-Registry-Administrator										X		X	X	X	X	X
Datenbank-administrator												X	X	X	X	X
IT-Architekt					X						X	X				
Service-Entwickler												X	X			
Technical Service Owner										X	X	X	X	X	X	X
Service-Tester													X			
Service-Designer/Modellierer											X					
SOA-Systemadministrator												X	X	X	X	X
Personalverantwortlicher						X	X									
Projektmanager	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Unternehmensführung	X	X				X		X								

Tabelle 1: Rollen-Phasen-Mapping für die SOA-Transformation (Eigene Darstellung)

Quellen:

Abts, Dietmar/Mülder, Wilhelm (2009): Grundkurs Wirtschaftsinformatik. Wiesbaden: Vieweg Verlag.

Alisch, Katrin/Winter, Eggert/Arentzen, Ute (Hg.) (2004): Wirtschaftslexikon. 16. akt. Aufl., Wiesbaden: Gabler Verlag.

Balzert, Helmut (2001): Lehrbuch der Software-Technik. Heidelberg/Berlin: Spektrum Akademischer Verlag.

Bieberstein, Norbert/Bose, Sanjay/Fiamante, Marc/Jones, Keith/Shah, Rawn (2006): Service-Oriented Architecture (SOA) Compass. Uppder Saddle River, N. J.: IBM Press.

Blutner, Doris/Neuhaus, Herbert M. (2007): Mensch-Maschine-Schnittstellen unter Bewährungsdruck: Kompetenzerwerb bei Einführung neuer IT-Systeme. In: Kompetenzentwicklung in realen und virtuellen Arbeitswelten. Dortmund: GfA-Press. S. 123–126.

Crasemann, Christoph/Krasemann, Hartmut/Vorwerk, Raymund (2008): Grosse IT-Projekte und ihre Erfolgsfaktoren (Teil III): Passende Projektbesetzung als Schlüssel zum Erfolg. In: Objektspektrum 2008, Nr. 2, S.44–52.

Erl, Thomas (2008): SOA Principles of Service Design. Boston, Mass.: Prentice Hall.

Ferstl, Otto. K./Sinz, Elmar. J. (2008): Grundlagen der Wirtschaftsinformatik. 6. überarb. Aufl., München: Oldenbourg Verlag.

Gu, Qing/Lago, Patricia (2007): A stakeholder-driven Service Life Cycle Model for SOA. Dubrovnik: Konferenzbeitrag IW-

SOSWE'07. (Verfügbar unter ACM Digital Library).

Hildebrandt, Eduard/Wolf, Andy (2009): Service-Lifecycle. In: Javasppektrum 2009, Nr. 1, S.30–33.

Kontogiannis, Kostas/Lewis, Grace A./Smith, Dennis B. (2008): A Research Agenda for Service-Oriented Architecture. Konferenzbeitrag SDSOA'08. Leipzig. (Verfügbar unter ACM Digital Library).

Kunzmann, Christiane (2005): Konzeption von Skills-Management-Instrumenten für die Bildungsbedarfsplanung in der Pflege am Städtischen Klinikum Karlsruhe. Diplomarbeit. Hochschule Pforzheim.

Kurbel, Karl/Becker,Jörg/Gronau,Norbert/Sinz,Elmar/Suhl, Leena(2008): Ablösung von Alsystemen. <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/is-management/Integration-und-Migration-von-IT-Systemen/Ablosung-von-Altsystemen>. Zuletzt aufgerufen am 25.10.2009.

Liebhart, Daniel (2007): SOA goes real. München/Wien: Hanser Verlag.

Maier, Berthold/Normann, Hajo/Trops, Bernd/Utschig-Utschig, Clemens/Winterberg, Thorsten (2008): SOA Design Pattern: Compensation. In: Javamagazin 2008, Nr. 7, S. 78–82.

Manes, Anne Thomas (2006): Vorwort. SOA ist eine Lebensweise. In: Gernot Starke/Stefan Tilkov(Hg.): SOA-Expertenwissen. Heidelberg: dpunkt Verlag, ix–xii.

Marks, Aric A./Bell, Michael (2006): Service-Oriented Architecture. A Planning and Implementation Guide for Business and Technology. Hoboken, N. J.: Wiley.

Masak, Dieter (2007): SOA? Serviceorientierung in Business und Software. Heidelberg: Springer Verlag.

Mertens, Peter et al. (2001): Lexikon der Wirtschaftsinformatik. Berlin/Heidelberg: Springer Verlag.

Müller, Thomas (2006): Der Weg zum guten Service. In: Gernot Starke/Stefan Tilkov (Hg.): SOA-Expertenwissen. Heidelberg: dpunkt Verlag, S. 141–159.

North, Klaus/Reinhardt, Kai (2005): Kompetenzmanagement in der Praxis. Wiesbaden: Gabler Verlag.

Pleus, Wolfgang (2008): SOA-Transformation. In: Javamagazin 2008, Nr. 7, S. 95–100.

Pulier, Eric/Taylor, Hugh (2006): Understanding Enterprise SOA. Greenwich: Manning.

Richter, Jan-Peter/Haller, Harald/Schrey, Peter (2008): Serviceorientierte Architektur. http://www.gi-ev.de/no_cache/service/informatiklexikon/informatiklexikon-detailansicht/meldung/serviceorientierte-architektur-118.html. Zuletzt aufgerufen am 25.10.2009.

Setzwein, Christian/Setzwein, Monika (2008): Turnaro und-Management von IT-Projekten. Heidelberg: dpunkt.

Stickel,Eberhard (2001): Informationsmanagement. München/Wien: Oldenbourg Verlag.

Vossen, Gottfried (2000): Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme. München: Oldenbourg Verlag.

Über die/den Autor_in

Dipl.-Wirtsch.-Inf. Martin Sommer hat sein Studium der Wirtschaftsinformatik an der Universität Leipzig mit Spezialisierung auf Management großer Softwaresysteme und Personalwirtschaftslehre im März 2006 erfolgreich abgeschlossen. Anschließend nahm er seine Tätigkeit als IT-Consultant bei der Firma evodion Information Technologies in Hamburg auf. Seit Ende August 2007 ist er Doktorand der Wirtschaftswissenschaftlichen Fakultät der Universität Leipzig im Bereich Projektmanagement für die Einführung von Service-orientierten Architekturen (SOA). Im Zentrum seiner Forschungsinteressen steht hierbei das Management von Skills. Betreut wird er durch Prof. Dr. Bogdan Franczyk (Institut für Wirtschaftsinformatik, Professur Informationsmanagement).