



# Java™magazin

Java • Architekturen • SOA • Agile

www.javamagazin.de

8.08

## SOA Center

### Schöne globalisierte Servicewelt?

Servicekommunikation zwischen Java und .NET mithilfe von WSIT ▶ 76

## Enterprise

### JBoss ESB im Einsatz

So kontrollieren Sie Komponenten heterogener Systeme ▶ 40

## Web

### Automatisch Testen

Webtests mit Selenium, Groovy, TestNG und Maven ▶ 65

## Tools

### Lego-Roboter mit Java programmieren

leJOS: Java für Lego Mindstorms ▶ 103

## Architektur

### Archi-Zack!-ture

Auf Gratwanderung zwischen Architektur und Entwicklung ▶ 58

## CD-Inhalt

Alle CD-Infos ▶ 3

### ■ JAX 2008



KEYNOTE VON  
MARTIN ODERSKY:

**SCALA** –  
The Challenge of  
Scalable Languages

### ■ Alle Quellcodes & Beispiele!

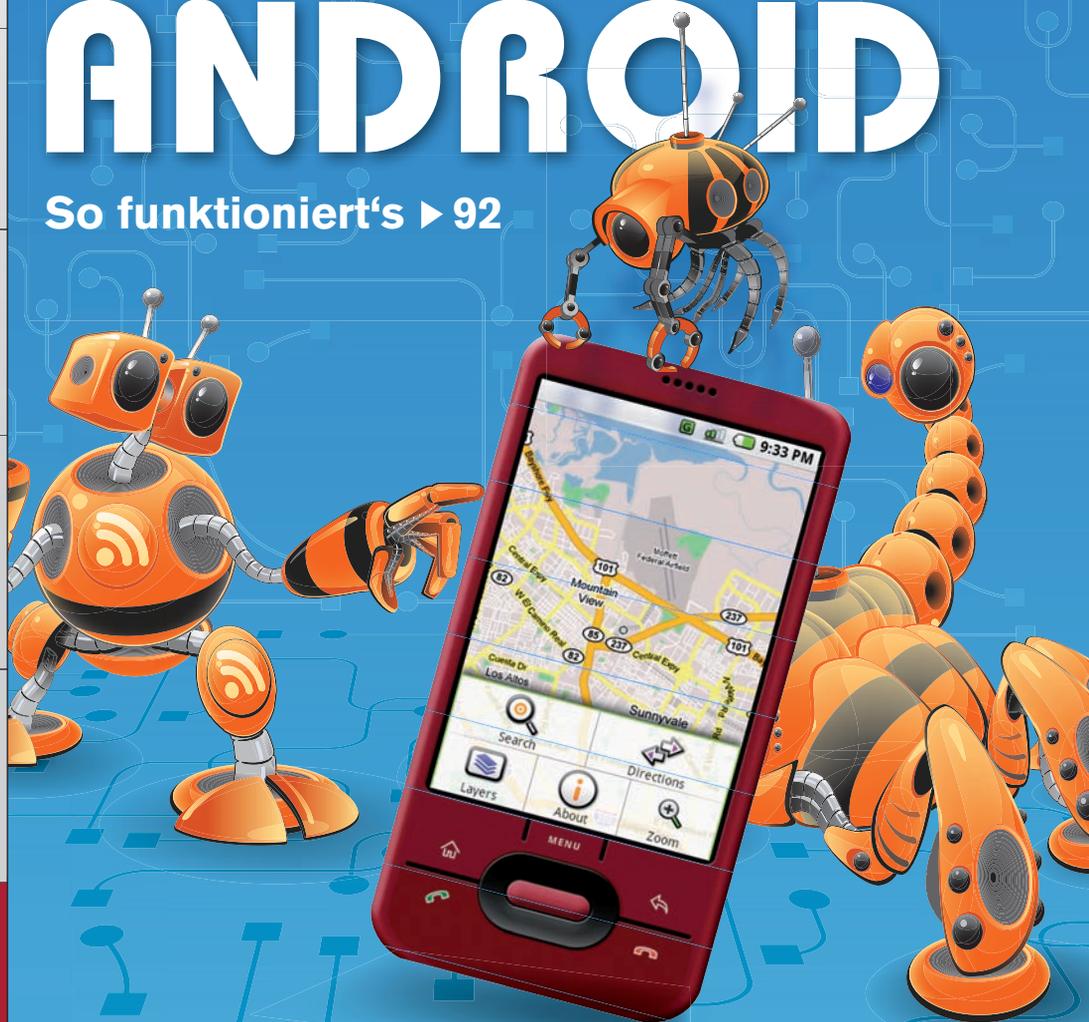
### ■ Testversionen und Tools

Apache Axis2, JAX-WS 2.1.4, Metro, Milyn Smooks, Spring Web Services, Spring Integration, Apache Felix, Maven 2.0.9, Equinox, Grails 1.0.3, Selenium, CXF

! Datenträger enthält nur Lehr- oder Infoprogramme

# GOOGLE ANDROID

So funktioniert's ▶ 92



## OSGi

▶ 52

### Zukunftsentwürfe

OSGi ist in aller Munde. Das „nächste Ding nach Spring“ adressiert Themen, die für die Anwendungsarchitektur relevant sind: Mittel zum Abhängigkeitsmanagement etwa oder das dynamische Servicemodell. Will man von OSGi profitieren, sollten beim Applikationsentwurf einige Aspekte beachtet werden.

## Neue Serie

▶ 100

### Effective Java

Klaus Kreft und Angelika Langer werden in Zukunft regelmäßig im Java Magazin unter dem Titel „Effective Java“ über Fallstricke und aktuelle Themen aus dem Bereich des Java-Sprachkerns und des JDKs berichten. In dieser Ausgabe beginnt eine Reihe über Aspekte des Java-Memory-Modells.

# SOA-Transformation – Rollen und Teams

Wie der organisatorische Übergang von der klassischen zur serviceorientierten Softwareentwicklung gelingen kann – Der zweite Teil dieser Serie beschäftigt sich mit den organisatorischen und personellen Veränderungen, die mit einer SOA-Transformation einhergehen.

von Wolfgang Pleus

In der IT-Branche werden Unternehmen und Softwareentwickler permanent mit neuen Paradigmen konfrontiert. Sei es Objektorientierung, Schichtenarchitekturen oder Dependency Injection, immer wieder ändert sich die Art, wie Software entwickelt wird. Während diese Paradigmenwechsel eine ständige Anpassung und Weiterentwicklung etablierter Entwicklungsmethoden erfordern, bleibt die Rollenverteilung in den Teams weitgehend unverändert. Dies erleichtert die Umstellung auf das jeweils neue Paradigma, da sich die Organisation als solche nicht verändert. Im Gegensatz dazu handelt es sich bei dem serviceorientierten Paradigma zu etwa gleichen Teilen um ein technisches wie auch organisatorisches Thema. Damit wird eine Umstellung ungleich schwieriger, da zusätzlich zu den technischen ebenso personelle Aspekte berücksichtigt werden müssen. In der Praxis stellt die Vernachlässigung der organisatorisch-personellen Aspekte das Hauptrisiko für eine nachhaltige SOA-Einführung dar.

## Klassische Teams

In heutigen IT-Projekten finden sich bestimmte Rollen immer wieder (Abb. 1). Dazu gehören Entwickler mit unterschiedlichem Spezialisierungsgrad,

zum Beispiel für Client- und Serverprogrammierung sowie Projektleiter, die den Ablauf koordinieren und die Projektplanung übernehmen. In der Regel finden sich diese Rollen in jedem Einzelprojekt wieder. Daneben gibt es einige zentrale Rollen. Die Analysten arbeiten in den Fachabteilungen und definieren die fachlichen Anforderungen an die Anwendungen. Architekten gestalten die Konsistenz der Anwendungen und der gesamten IT-Landschaft. Daneben gibt es Tester, die sicherstellen, dass die Anwendungen vor der Inbetriebnahme die funktionalen und nichtfunktionalen Anforderungen erfüllen. Es kommt auch vor, dass die zentralen Rollen projektspezifisch besetzt sind. In diesem Fall verfügt ein Projekt über eigene Tester und/oder Architekten. Zudem können mehrere Rollen von einer Person verkörpert werden. Dies ist beispielsweise der Fall, wenn ein Architekt ebenfalls Entwicklungsaufgaben übernimmt.

Während diese Rollen in einem projektzentrischen Umfeld adäquat sind, sind sie in einem serviceorientierten Umfeld in der Regel nicht mehr ausreichend. Der Grund ist vor allem der wesentlich größere Kontext und die damit steigende Komplexität. Um dieser Situation gerecht zu werden, ist eine stärkere Spezialisierung der einzelnen

Rollen sowie die Einführung neuer Rollen erforderlich.

## SOA-Rollen

Durch die klarere Trennung der Verantwortlichkeiten und den höheren Spezialisierungsgrad in einer SOA ändern sich die Ausprägungen der klassischen Rollen. So wird sich ein Entwickler, der sich in einem Projekt bislang mit Integrationsthemen und Oberflächenprogrammierung beschäftigt hat, wahrscheinlich als Serviceentwickler eher mit Fachlogik innerhalb einer Fachdomäne beschäftigen, oder er wird sich auf die Oberflächen für die Benutzerschnitte im Workflow spezialisieren. Für Architekten, Projektleiter und Tester ändert sich vor allem der technische Fokus durch den Einsatz von Business Process Management (BPM) oder Enterprise Service Bus (ESB).

In einer serviceorientierten Architektur richten sich die Rollen nicht mehr an Projekten, sondern an horizontalen Schichten oder fachlichen Domänen aus. Zudem werden wichtige Themen wie beispielsweise Integration, Sicherheit oder Prozesse zentralisiert. Dies führt zu neuen Rollen, die diese zentralisierten Bereiche ausfüllen. Diese neuen Rollen sind essenziell, um langfristig eine konsistente Service- und Prozess-

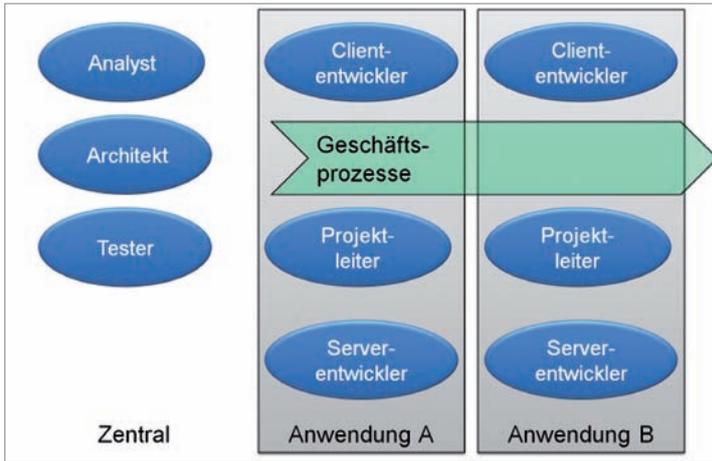


Abb. 1: Klassische Projektrollen

landschaft zu gewährleisten. Wichtige neue Rollen sind Domänenexperte, Bibliothekar, Integrationsspezialist und Prozessdesigner (Abb. 2).

Der *Domänenexperte* verfügt über detailliertes, semantisches Wissen über die fachliche Domäne, beispielsweise Kunden oder Produkte. Er bestimmt, welche Fachlichkeit in seiner Domäne abgebildet wird und wie diese von anderen genutzt werden kann. Er stellt sicher, dass die Domäne jederzeit konsistent und klar strukturiert ist, und dass es keine Überschneidungen mit anderen Domänen gibt. Zudem ist er der fachliche Ansprechpartner für potenzielle Servicenutzer. Diese Rolle ist wichtig, um Servicewildwuchs zu vermeiden, wie er oft bei rein anforderungsgetriebenen Serviceimplementierungen auftritt. Der Domänenexperte ist gewissermaßen der fachliche Architekt seiner Fachdomäne. Während in einem kleinen Unternehmen ein Experte mehrere Domänen betreuen kann, findet in größeren Unternehmen eine stärkere Spezialisierung auf einzelne Fachdomänen statt. Dies ist erforder-

lich, um auch in komplexen SOA-Landschaften Konsistenz zu gewährleisten. Zu seinen Fähigkeiten gehört vor allem eine genaue Kenntnis der jeweiligen Domäne. Daher kommen für diese Rolle beispielsweise frühere Analysten oder auch Projektleiter in Frage. Aber auch Anwender mit detailliertem Fachwissen können sich für diese Rolle qualifizieren.

Der *Bibliothekar* stellt eine Schlüsselrolle in Bezug auf die Wiederverwendbarkeit von Prozessen und Services dar. Wir erinnern uns, dass ein großer Teil der Agilität einer SOA aus der Kopplung allgemeiner Services zu Verbundservices oder Prozessen erwächst. Wiederverwendung beschäftigt die IT seit jeher. Erfolge in diesem Bereich sind jedoch eher bescheiden. Wie kann es also diesmal gelingen, Teile des Gesamtsystems mehrfach zu nutzen? Der Schlüssel liegt im Einsatz von Repositories, die Services, Prozesse und andere Artefakte auffindbar machen. Die Vorstellung, dass die Entwickler ein solches Werkzeug benutzen, um Services aufzufinden, ist aufgrund der oft komplexen fachlichen

Zusammenhänge jedoch naiv. Bevor ein Service verwendet werden kann, muss klar sein, ob er die erforderlichen fachlichen und technischen Anforderungen tatsächlich erfüllt. Der Bibliothekar ist der Ansprechpartner für Serviceentwickler und Prozessdesigner. Er führt im Auftrag dieser Personen Recherchen durch, um geeignete Servicekandidaten zu ermitteln. Dazu kann er das Repository einsetzen und arbeitet eng mit den Domänenexperten zusammen, um es zu pflegen. Zu seinen Fähigkeiten gehören die Kenntnis des eingesetzten Repositories sowie ein solider fachlicher Hintergrund. Technische Kenntnisse, beispielsweise über WSDL oder UDDI, sind ebenfalls erforderlich. Für diese Rolle kommen daher Analysten oder Projektleiter mit technischer Affinität in Frage.

Der *Integrationsspezialist* stellt sicher, dass die Services universell erreichbar sind. Die Integration eines Services in die SOA-Infrastruktur ist nicht mehr primär Aufgabe des Serviceentwicklers, sondern wird in enger Abstimmung mit dem Integrationsspezialisten realisiert. Dem Integrationsspezialisten kommt dabei eine beratende Rolle zu. Er bindet die Services mit den jeweils geeigneten Protokollen an und stellt eine hochverfügbare Kommunikationsinfrastruktur bereit. Zudem entwickelt er Vorgaben für Schnittstellenspezifikationen, um eine bestmögliche Integration einzelner Services zu erreichen. Da die Integration und Kommunikation von Services für den Betrieb einer SOA grundlegend ist, sollte diese Rolle möglichst nicht mit anderen Rollen geteilt werden. Zu seinen Fähigkeiten gehören eine genaue Kenntnis der eingesetzten Kommunikationsinfrastruktur und Standards wie beispielsweise Java Business Integration (JBI) [2], Erfahrungen mit den gängigen Protokollen, wie SOAP, FTP, HTTP, SMTP etc. Zudem sollte er über Kenntnisse der gängigen Integrationsmuster verfügen [3]. Als Kandidaten für diese sehr technische Rolle kommen Entwickler oder Architekten in Frage.

Der *Prozessdesigner* ist für die Abbildung der Geschäftsprozesse in die technische Laufzeitumgebung zuständig. Er definiert die Prozesse zusammen mit den Fachabteilungen und modelliert diese

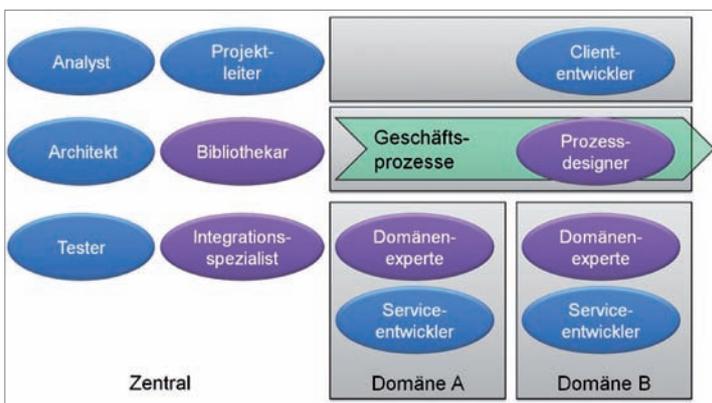
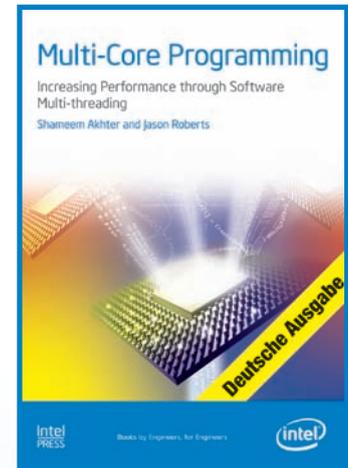


Abb. 2: SOA-Rollen

**Jetzt NEU:**  
**Das deutsche Programm!**



Shameem Akhter, Jason Roberts

## Multi-Core Programmierung

Intel hat ein neues Zeitalter der Prozessorarchitektur eingeleitet - Multi-core-Computing. Bei der Intel Multi-core-Architektur sind zwei oder mehr Prozessorkerne bzw. Recheneinheiten in einem einzigen Prozessorgehäuse untergebracht. Mit der entsprechenden Software können diese Systeme mehrere Software-Threads parallel ausführen. Die autorisierte Übersetzung des Originalbuchs von Intel Press USA zeigt die Grundlagen und erläutert Software-Entwicklern, wie sie ihre Programme optimal an die Intel-Prozessoren anpassen.

ca. 350 Seiten, Softcover, 54,90 €  
ISBN 978-3-939084-70-9

Powered by: **entwickler.press**

Weitere Informationen und Bestellmöglichkeiten finden Sie unter [www.entwickler.press.de](http://www.entwickler.press.de). Unsere Bücher erhalten Sie auch in jeder gut sortierten Buchhandlung.

Klassische Rollen / SOA Rollen	Analyst	Projektleiter	Architekt	Tester	Client-entwickler	Server-entwickler	Anwender
Analyst	X						
Projektleiter		X					
Architekt			X				
Tester				X			
Cliententwickler					X		
Serviceentwickler						X	
Anwender							X
Bibliothekar	X	X					X
Domänenexperte	X	X					X
Integrationspezialist			X		X	X	
Prozessdesigner	X	X			X	X	

Abb. 3: Rollenmatrix

zunächst in grober Form. Zudem berät er die Fachabteilungen und versetzt sie so in die Lage, eigenständig Prozesse zu definieren. Diese Prozesse werden danach vom Prozessdesigner in eine technische Form überführt. Der Prozessdesigner stellt somit die Schnittstelle zwischen der fachlichen Sicht und der technischen Abbildung dar. Dies ist eine besondere Herausforderung, da sie Kenntnisse in beiden Bereichen erfordert. Dennoch ist es sinnvoll, diese Rolle nicht in eine fachliche und eine technische aufzuteilen. Dadurch wird sichergestellt, dass die Prozesse tatsächlich technisch abgebildet werden können, Reibungsverluste werden so vermieden. Als Kommunikationsmittel werden primär grafische Modellierungswerkzeuge eingesetzt. Ein Prozessdesigner sollte über Kenntnisse in Business in den Bereichen Business Process Modeling Notation (BPMN), Unified Modeling Language (UML), Business Process Execution Language (BPEL), Web Service Description Language (WSDL) und Extensible Markup Language (XML) verfügen. Damit kommen Personen sowohl aus dem fachlichen als auch aus dem technischen Bereich in Frage, sofern Sie über Kenntnisse in beiden Bereichen verfügen. Da dies eher selten der Fall ist, ist diese Rolle erfahrungsgemäß schwer zu besetzen. Eine praktikable Möglichkeit ist es, dass zwei Personen mit Fach- und Technikwissen für eine Übergangszeit sehr eng zusammenarbeiten und sich somit das Wissen des jeweils anderen aneignen. Voraussetzung ist eine ausgeprägte Lernbereitschaft.

### Und die Anwender?

Von einer SOA-Transformation sind nicht nur die Entwicklungsteams und

Fachabteilungen, sondern auch die Anwender betroffen. Sie kennen die Geschäftsprozesse und Funktionalitäten, die durch die SOA unterstützt werden sollen. Aufgrund fehlender Dokumentation sind sie oft die einzigen, die über dieses Wissen verfügen. Eine SOA-Einführung hat in der Regel deutliche Auswirkungen auf den Arbeitsbereich der Anwender. Manche Rollen fallen unter Umständen weg, da Prozesse automatisiert werden und sich die Benutzerinteraktion auf ein Minimum beschränkt. Andere Rollen ändern sich durch die Einführung neuer Werkzeuge. Daraus ergibt sich ein Spannungsfeld, das nicht zu unterschätzen ist. Die Einbeziehung der Anwender ist daher im Rahmen einer SOA-Transformation essentiell. Abbildung 3 zeigt sinnvolle Entwicklungsperspektiven für Teammitglieder auf.

### SOA-Teams

Die hier vorgestellten Rollen sollen vor allem ein Bewusstsein für die neuen Aufgaben schaffen, die mit Einführung einer serviceorientierten Architektur wahrgenommen werden müssen. Damit stellt sich die Frage, wie Teamstrukturen auf Basis dieser Rollen gebildet werden können. Grundsätzlich orientieren sich die Teams an den horizontalen Schichten Oberflächen, Prozesse und Domänen [1]. Folglich gibt es ein oder mehrere Prozess- und Oberflächenteams, sowie je ein Team pro Fachdomäne. Bei der Rollenverteilung empfiehlt sich ein pragmatischer Ansatz. So sind bei kleinen Vorhaben manche Rollen noch optional. Aufgrund der geringen Komplexität kann in der Regel auf den Bibliothekar und Integrationspezialisten zunächst verzichtet werden. Bei

mittleren Vorhaben haben einzelne Personen in der Regel mehrere Rollen inne. So können zum Beispiel die Rollen Bibliothekar und Domänenexperte zusammengefasst werden. So lassen sich alle Rollen auch mit wenigen Personen abbilden. Bei großen Vorhaben findet eine weitere Spezialisierung statt. Die Folge ist eine explizite Rollenbesetzung durch jeweils eine Person.

### Governance

Als SOA-Governance bezeichnet man die Überwachung der Einhaltung wichtiger Standards mit dem Ziel, langfristig eine stabile und wartbare SOA-Landschaft zu schaffen. Die Produkthersteller bezeichnen ihre Repositories oft auch als Governance-Lösungen. Dabei geht es jedoch meist eher um das Thema Servicemanagement, das heißt die Verwaltung von Informationen zu bestehenden Services und Prozessen. SOA-Governance ist ein rein organisatorisches Thema, bei dem die Zusammenarbeit und Ausrichtung der SOA-Teams geregelt wird. Bei den zu überwachenden Standards kann es sich um offene Standards wie WSDL- oder BPMN-Modellierung oder auch unternehmensspezifische Standards, wie beispielsweise zur Servicegranularität oder Dokumentation handeln. Je größer ein SOA-Vorhaben ist, umso wichtiger wird die Einrichtung von Governance Boards, in dem Repräsentanten der SOA-Teams vertreten sind. In der Regel finden sich in einem solchen Board die Rollen mit übergeordneten Verantwortlichkeiten, wie Architekten, Domänenexperten oder Bibliothekare, wieder. Grundsätzlich wird zwischen zentralistischen, föderativen und hierarchischen Modellen unterschieden.

Beim zentralistischen Modell gibt es nur ein Board, in dem alle Entscheidungen gebündelt werden. Dies erlaubt eine sehr gute Kontrolle. Allerdings skaliert es nicht und kommt damit nur zu Beginn einer SOA-Transformation oder für kleine SOA-Vorhaben in Frage.

Das föderative Modell umfasst mehrere, gleichberechtigte Boards, die sich regelmäßig abstimmen und Entscheidungen gemeinsam treffen. Das Modell skaliert besser als das zentralistische

Modell, hat aber den Nachteil des zunehmenden Kontrollverlustes bei steigender Komplexität des SOA-Vorhabens. Es eignet sich besonders für mittelgroße Vorhaben, in denen es beispielsweise mehrere gleichberechtigte Abteilungen gibt.

Beim hierarchischen Modell erfolgt die Entscheidungsfindung gemäß der hierarchischen Struktur. Hierbei gibt es mehrere Boards, bei denen die übergeordneten Boards Vorgaben für die untergeordneten entwickeln. Dieses Modell kommt beispielsweise bei internationalen Organisationen zum Einsatz, wobei es für jede Region (zum Beispiel US und EMEA) ein eigenes Board gibt, das sich mit dem zentralen Board abstimmt. Es eignet sich vor allem für große, regional verteilte SOA-Vorhaben.

### Erfolgsfaktoren

Die Erfahrung zeigt, dass Verzögerungen oder Fehlschläge häufig darauf zurückzuführen sind, dass die betroffenen Personen nicht ausreichend einbezogen werden. Dies gilt gleichermaßen für die Entwicklungsteams als auch die Anwender der Systeme. Dabei ist es leicht nachvollziehbar. Schließlich geht es bei einer SOA nicht selten darum, bestehende Geschäftsprozesse ganz oder zumindest teilweise zu automatisieren. Das Prozesswissen ist oft zum großen Teil nur bei den Anwendern bekannt, die die Prozesse einfach leben. Der Transfer des Prozesswissens in ein IT-System setzt die unbedingte Mitarbeit dieser Prozesskenner voraus. Bei den Entwicklungsteams ist es ähnlich. Ein Serverentwickler, der bisher weitgehende Hoheit

über seinen technischen Bereich hatte, wird diese als Serviceentwickler wahrscheinlich verlieren, wenn er fortan die Vorgaben der Architekten beziehungsweise der Governance Boards umsetzen soll. Wenn unklar ist, welche Rolle diese Personen im Rahmen einer serviceorientierten Organisation haben, können sie nur schwer zur Mitarbeit bewegt werden. Da die Einführung einer SOA mit personellen Umstellungen verbunden ist, empfiehlt es sich, einen Change-Management-Prozess aufzusetzen, wie er beispielsweise bei einer Firmenübernahme zum Einsatz kommt. So wie eine Referenzarchitektur im Rahmen der SOA-Roadmap beschrieben wird, sollte ebenfalls ein Referenzorganisationsplan mit den zu besetzenden Rollen erstellt werden. Anhand dieses Plans können frühzeitig Entwicklungsmöglichkeiten für die Teammitglieder erläutert und mit den Beteiligten abgesprochen werden. Dies ermöglicht einen hohen Grad der Kooperation und trägt somit entscheidend zum Erfolg bei.

### Fazit

Eine SOA-Transformation beinhaltet weit mehr als den Einsatz bestimmter Technologien. Dazu gehört vor allem das richtige Vorgehen im Rahmen eines SOA-Programms, unter Einbeziehung personeller Veränderungen. Dies unterscheidet eine SOA grundsätzlich von anderen Paradigmen in der Softwareentwicklung. Wenn diesen Aspekten genügend Aufmerksamkeit gewidmet wird, kann eine serviceorientierte Architektur ihre Potenziale für ein Unternehmen entfalten. ■



**Wolfgang Pleus** arbeitet als Technologieberater, Autor und Trainer im Bereich serviceorientierter Architekturen. Seit über 15 Jahren unterstützt er internationale Unternehmen bei der Realisierung komplexer Geschäftslösungen auf der Basis von J2EE und .NET. Er ist Mitglied des Accelsis SOA-Competence-Teams.

### Links & Literatur

- [1] Wolfgang Pleus: SOA-Transformation, Teil 1, in: Java Magazin 07.2008, S. 95–100
- [2] Java Business Integration: [jcp.org/en/jsr/detail?id=208](http://jcp.org/en/jsr/detail?id=208)
- [3] Enterprise Integration Patterns: [www.enterpriseintegrationpatterns.com](http://www.enterpriseintegrationpatterns.com)