



Java™ magazin

Java • Architekturen • SOA • Agile

www.javamagazin.de

7.08

Enterprise

Komplexität meistern

Möglichkeiten und Grenzen des Java Profiling ▶ 24

GlassFish

Der Open-Source-Newcomer von Sun ▶ 36

Web

JSON

Beispiele für den Einsatz von JavaScript Object Notation ▶ 75

SOA Center

Serverside OSGi

Java EE und OSGi integrieren ▶ 81

Tools

Parser entwickeln

Eine Einführung mit SableCC ▶ 101

Getting Things Done

ThinkingRock2: So organisieren Sie Ihre Gedanken ▶ 112

CD-Inhalt

Alle CD-Infos ▶ 3

■ JAX 2008

KEYNOTE VON ROD JOHNSON:

The Future of Enterprise Java



■ Alle Quellcodes & Beispiele!

■ Tools, Tools, Tools

Spring 2.5.4, GlassFish V2, Hibernate 3.2.6, AspectJ 1.6.0, SableCC 3.2, Equinox, JUnit 4.4, IzPack 3.11.0, EasyMock 2.3, ThinkingRock 2, Statsviewer, Shoal 1.0, OpenSER, Bnd Tool

Gratwanderung zwischen Architektur und Entwicklung ▶ 60



SOA Center ▶ 95

SOA-Transformation

Die Industrie legt den Fokus hauptsächlich auf die technischen Aspekte von SOA. Aber Technik ist kein Selbstzweck: Für Projektverantwortliche stellt sich die Frage, wie eine SOA-Transformation eigentlich erfolgen soll. Wie kann der organisatorische Übergang von der klassischen zur serviceorientierten Softwareentwicklung gelingen?

Internet-Telefonie ▶ 68

VoIP selbstgemacht

Zu „skypen“ ist momentan en vogue, doch wie kann man VoIP selbst realisieren? Muss man immer einen Client installiert haben, um wirklich zu telefonieren? Warum sollte es nicht möglich sein, dies webbasiert zu tun? Mittels des WeSIP-Servlet-Containers ist es möglich, VoIP- und IMS-Applikationen selbst zu entwerfen.



SOA-Transformation

Im technologiegetriebenen Umfeld moderner Softwareentwicklung liegt der Fokus naturgemäß primär auf technischen Aspekten. Dies gilt auch für das Thema serviceorientierte Architekturen. Daneben stellt sich den Projektverantwortlichen die Frage, wie eine Transformation in Richtung serviceorientierter Entwicklung erfolgen soll.

von Wolfgang Pleus

Bevor mit der Einführung einer serviceorientierten Architektur im Unternehmen begonnen wird, ist es wichtig, dass den Beteiligten die Ziele und der Nutzen des Vorhabens für das Unternehmen möglichst klar sind. Dies klingt trivial, ist aber essenziell, da im Verlauf einer SOA-Einführung viele Entscheidungen zu treffen sind, die sich an diesen Zielen messen lassen müssen. Dabei geht es um Fragen wie die, welche Granularität die Services haben sollen, welche Business Process Management (BPM) Engine eingesetzt werden soll, oder ob ein Enterprise Service Bus

(ESB) zum Einsatz kommt. Sind die Ziele nicht ausreichend klar, besteht ein hohes Risiko, dass am Ende zwar eine Reihe von Produkten aus dem SOA-Umfeld eingesetzt wird, der Nutzen für die Organisation aber hinter den Erwartungen zurückbleibt.

Das Hauptziel eines SOA-Vorhabens ist die gesteigerte Agilität des Unternehmens in Bezug auf sich schnell verändernde Marktbedingungen. Durch eine stärkere Verzahnung von IT und Fachbereichen, auch IT/Business-Alignment genannt, soll ein Wettbewerbsvorsprung erreicht werden. Daneben kann es natürlich noch Nebenziele wie

die Konsolidierung einer bestehenden Systemlandschaft geben. Wichtig ist zu verstehen, dass Technologien kein

SOA-Transformation

Dieser erste Teil des Artikels beschreibt die typischen Schritte einer SOA-Transformation und zeigt, was bei der Planung besonders zu beachten ist. Im zweiten Teil in der nächsten Ausgabe des Java Magazins wird beschrieben, welche organisatorischen Veränderungen mit einer solchen Transformation einhergehen.

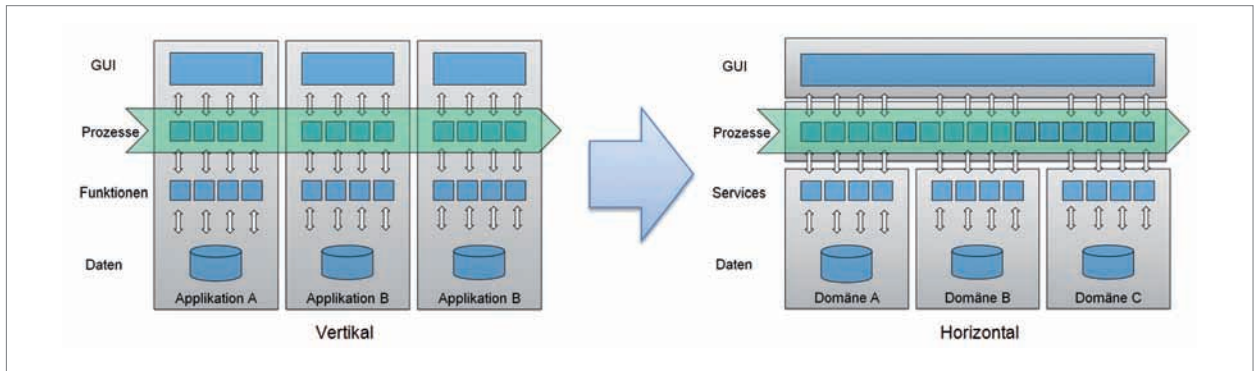


Abb. 1: Ist- und Sollarchitekturen

Selbstzweck sind, sondern sich an den geschäftlichen Zielen messen lassen müssen.

Ist- und Sollarchitekturen

Klassische Anwendungen werden in der Regel im Rahmen einzelner Projekte realisiert. Die Ausrichtung und der Umfang solcher Projekte beziehen sich dabei auf einen speziellen Problembereich, beispielsweise ein neues Portal für das Direktkundengeschäft einer Versicherung. Im Rahmen dieser Projekte werden für die Projektziele relevante Entscheidungen getroffen. Dazu gehören die Auswahl einer Datenbank oder der Einsatz von JSF und EJB für die Implementierung. Die so realisierten Systeme sind eher vertikal ausgerichtet. Das heißt, dass jedes System weite Teile von der grafischen Oberfläche über die Geschäftslogik bis zur Datenzugriffstechnologie selbst regelt. Selbst Produkte wie Applikationsserver werden oft projektbezogen lizenziert. Dieses vertikale Modell hat jedoch einige Schwächen. Jeder kennt das: Obwohl Wiederverwendung prinzipiell gewünscht ist, findet sie jenseits der Projektgrenzen kaum statt. Stattdessen wird das Rad oft neu erfunden und in Form redundanter Funktionalitäten auf allen Ebenen neu implementiert. Geschäftsprozesse überspannen nicht selten mehrere Anwendungen und Systeme. Daraus ergeben sich Systembrüche, die zu Integrationsaufwänden und unbefriedigenden Benutzererfahrungen führen. Mit jedem neuen Projekt nimmt die Komplexität des Gesamtsystems zu. Nicht zuletzt deshalb konsolidieren Unternehmen ihre Systemlandschaften mit entsprechendem Aufwand.

Serviceorientierte Architekturen folgen einem eher horizontalen Modell. Dabei stellen Services allgemeine Funktionalitäten für unterschiedliche fachliche Domänen bereit. Typische Domänen verwalten beispielsweise Kunden, Verträge oder Tarife. Idealerweise verfügen die Services über einen hohen Wiederverwendungsgrad und geringe Redundanz. Verwendet werden sie von Prozessen, die mittels eines BPM-Systems abgebildet werden. Diese Prozesse können mehrere Domänen umspannen. Die Interaktion der Benutzer mit den Prozessen wird über grafische Oberflächen ermöglicht, die ihm die zu bearbeitenden Prozessschritte präsentieren. Prozesse mit Benutzerinteraktion werden auch als Workflows bezeichnet. Die angestrebte Agilität basiert primär auf der Prozessschicht, die eine flexible Anpassung auf Basis existierender Services ermöglichen soll. Durch grafische Prozessmodellierung werden Fachbereiche und IT-Abteilungen einander angenähert.

Aller Anfang ist schwer

Wie jedes neue Paradigma hat es auch die Serviceorientierung am Anfang schwer. Dies liegt daran, dass verständlicherweise niemand auf ein Paradigma setzen möchte, bevor feststeht, dass es wirklich Erfolg bringt. Bei der Einführung eines neuen Frameworks ist dieser Beweis schnell erbracht.

Die Effekte einer serviceorientierten Architektur zeigen sich dagegen eher langfristig und sind kurzfristig kaum verifizierbar. Da das Thema an sich aber sehr vielversprechend ist, beginnen viele Unternehmen, ausgewähl-

te Technologien wie BPM oder ESB im Rahmen eines Projekts anzuwenden. Die Aussagekraft jenseits der rein technischen Fragestellungen ist dabei jedoch sehr eingeschränkt. Ohne eine unternehmensweite oder zumindest abteilungsweite SOA-Strategie enden viele Projekte, noch bevor sie eigentlich begonnen haben. Daher ist es erforderlich, dass das Thema SOA als strategisches Ziel eines Unternehmens begriffen und kommuniziert wird. Dies kann nicht aus einem Projekt heraus geleistet werden, sondern muss vom Management getrieben werden. Den Projektmitarbeitern fällt dabei eine beratende Rolle zu, wenn es darum geht, die möglichen Potenziale zu vermitteln.

Big Bang oder evolutionär?

Bei der Wahl des grundsätzlichen Vorgehens bieten sich zwei Ansätze an. Beim Big Bang wird ein holistischer Ansatz verfolgt. Dieser Ansatz wird auch als strategischer oder Top-Down-Ansatz bezeichnet. Die Basis bilden die Geschäftsprozesse und fachlichen Domänen eines Unternehmens. Am Beginn steht eine ausgiebige Analysephase, um den Problembereich weitestgehend zu erfassen. Auf dieser Basis wird dann die Prozess- und Serviceschneidung vorgenommen und somit die serviceorientierte Landschaft etabliert. Der Vorteil besteht darin, dass frühzeitig eine Fokussierung auf die Geschäftsprozesse stattfindet. Daraus kann eine vernünftige Domänenschneidung und damit verbundene Servicelandschaft hervorgehen. Doch dieser Ansatz ist in der Praxis nicht unproblematisch. Zum einen ist es für

Anzeige



Abb. 2: SOA-Programm



Abb. 3: Phasen der Roadmap-Definition

große Unternehmen oft kaum möglich, eine ganzheitliche Sicht zu erreichen. Zudem entstehen hohe Initialkosten für Analyse, Evaluierung und Infrastruktur, ohne dass dem zunächst ein konkreter Nutzen gegenübersteht.

Nicht zuletzt stehen einem solchen Ansatz unter Umständen kulturelle Widerstände entgegen, da eine neue Ausrichtung der Entwicklung Verantwortlichkeiten verschiebt und die Entscheidungsfreiheit der Projekte einschränken kann.

Die Lernkurve ist ebenfalls nicht zu unterschätzen. Denn die SOA-Paradigmen müssen im Unternehmen vermittelt werden, damit sie von den Mitarbeitern unterstützt und getragen werden.

Der evolutionäre, auch taktischer oder Bottom-Up-Ansatz genannt, nähert sich der SOA in Form kleinerer Schritte. Dabei werden zunächst bestehende Systeme SOA-fähig gemacht. Dies erfolgt beispielsweise durch die Bereitstellung offener Schnittstellen,

die von anderen Projekten verwendet werden können. Der Einsatz einer BPM Engine für die Prozessabbildung ist ebenfalls ein häufig gewählter Weg. Vorteilhaft dabei ist, dass die Machbarkeit sehr schnell dargestellt werden kann. Zudem entsteht direkt ein geschäftlicher Nutzen, da konkrete fachliche Anforderungen umgesetzt werden. Dieser Weg ist zudem leicht einzuschlagen, da sich die Aktivitäten im bekannten Projektrahmen bewegen. In Bezug auf eine ganzheitliche SOA ist der Wertbeitrag jedoch gering, da der Kontext eines Projekts zu klein ist und die SOA-Effekte vor allem projektübergreifend entstehen. Zudem entspricht die Granularität der Projekte nicht notwendigerweise der Granularität der SOA-Domänen, wodurch strukturelle Verbesserungen verhindert werden.

Es ist sicherlich ein sinnvoller erster Schritt, die SOA-Paradigmen innerhalb eines Projekts zu erproben. Wenn es danach aber nicht gelingt, diese Pa-

radigmen auf einer höheren Ebene wie einer Abteilung zu etablieren, kann von einer serviceorientierten Architektur, die sich nicht alleine über technische Modebegriffe definiert, kaum gesprochen werden.

Der evolutionäre Ansatz hat sich in der Praxis bewährt, sofern er Teil einer SOA Roadmap ist, die vom Management aktiv unterstützt wird. So sind die Aktivitäten immer in einen größeren Kontext eingeordnet, mit dem Ziel, die SOA schrittweise einzuführen. Zeitgleich kann im Rahmen eines strategischen Ansatzes mit der Prozess- und Domänenanalyse begonnen werden. Durch regelmäßige Abstimmungen kann so mit der Zeit eine stabile Unternehmens-SOA entstehen.

Das SOA-Programm

Der erste konkrete Schritt in Richtung einer SOA ist das Aufsetzen eines SOA-Programms. Das SOA-Programm beschreibt die grundsätzlichen Phasen der Transformation. Diese Phasen können variieren, beinhalten jedoch prinzipiell immer die in Abbildung 2 gezeigten Phasen. Das Aufsetzen des SOA-Programms bedeutet nicht notwendigerweise, dass die SOA auch tatsächlich umgesetzt wird. Es ist lediglich der erste Schritt im Rahmen eines strukturierten Vorgehens und stellt ein grundsätzliches Bekenntnis zum Thema SOA dar.

Die Phase der Information beinhaltet eine erste Auseinandersetzung mit dem Thema, die in beliebiger Form erfolgen kann – beispielsweise durch Fachvorträge, Workshops, Konferenzbesuche, oder Literaturstudium. Ziel ist es, ein Grundverständnis zu erlangen, das als Basis für die weiteren Phasen erforderlich ist.

Ob eine SOA für ein Unternehmen sinnvoll ist, hängt vor allem davon ab, ob es von einem Agilitätswachstum und einem besseren IT/Business-Alignment profitieren kann. Dies ist zum Beispiel der Fall, wenn die Geschäftsprozesse maßgeblich durch die IT unterstützt werden und sich das Unternehmen in einem dynamischen Marktumfeld bewegt. Während der Potenzialanalyse wird die Eignung



Abb. 4: Aktivitäten einer Iteration

einer SOA unter Einbeziehung von Geschäfts- und IT-Strategie sowie der Prozess- und Anwendungslandschaft ermittelt und dargestellt. Je nach Ergebnis der Analyse wird das SOA-Programm an dieser Stelle beendet oder fortgeführt.

Die SOA Roadmap

Die SOA Roadmap stellt den Masterplan für die SOA-Umsetzung dar. Dieser Plan ist für jedes Unternehmen individuell, weil er die bestehende Projekt- und Anwendungslandschaft einbezieht. Im Rahmen der Roadmap-Definitionsphase wird die konkrete, individuelle Roadmap entwickelt. Die Roadmap-Methodik variiert je nach Hersteller oder Beratungshaus [3], [4], folgt aber grundsätzlich den Phasen aus Abbildung 3. Die während der verschiedenen Phasen erforderlichen Aktivitäten sind in Tabelle 1 dargestellt.

In der Planungsphase geht es darum, den Umfang klar abzugrenzen. Dabei ist es einerseits wichtig zu beschreiben, was Teil (Systeme, Organisationseinheiten) des SOA-Vorhabens ist. Andererseits ist klar zu beschreiben, welche Teile nicht einbezogen werden. Dies hilft, eine realistische Erwartungshaltung bei allen Beteiligten aufzubauen und spätere Enttäuschungen zu vermeiden. In dieser Phase sollten zudem die Teams für die weiteren Phasen benannt werden. Zudem sollte ein Steuerungsgremium mit Entscheidungskompetenz benannt werden, das auftretende Fragestellungen während der Roadmap-Definition verbindlich klären kann. Beispielsweise kann es erforderlich sein, den Umfang des SOA-

Vorhabens aufgrund neuer Erkenntnisse neu auszurichten. Da dies unter Umständen weitreichende Konsequenzen haben kann, sollte eine solche Fragestellung vom Steuerungsgremium entschieden werden.

Die Bewertung der Reife beinhaltet eine Analyse der bestehenden Landschaft in Bezug auf IT-Systeme, Geschäftsprozesse und Organisationsstrukturen. Da sich eine SOA an der bestehenden IT- und Geschäftsstrategie ausrichten muss, sollten diese ebenfalls mit in die Analyse einbezogen werden. Nicht selten ist es der Fall, dass die Zusammenhänge der bestehenden Landschaft in der Organisation nur unzureichend bekannt sind. Daher empfiehlt es sich, einen oder mehrere Systematlanten und Organigramme zu erstellen. Dieser Aufwand zahlt sich schnell aus, da sie Kommunikation in den weiteren Phasen unterstützt. Falls eine Potenzialanalyse durchgeführt wurde, können die Ergebnisse in die Bewertung der Reife einbezogen werden. Wenn eine externe Sicht erforderlich ist, können auch SOA-Maturity-Modelle herangezogen werden.

Bei der Entwicklung der Zieldefinition geht es darum, das Soll in technischer und organisatorischer Hinsicht zu beschreiben. Ziel ist es, möglichst plastisch darzustellen, wie das Unternehmen in technischer und organisatorischer Hinsicht nach der Transformation aussehen soll. Dabei ist es hilfreich, eine Referenzarchitektur zu erstellen. Diese enthält die elementaren Bestandteile wie Business Process Management, Enterprise Service Bus, Business Activity Monitoring oder Repository. Erste Domänen- und

Anzeige

| Phase | Ergebnis |
|--------------------------------|---|
| Planung | Umfang und Abgrenzung festlegen Steuerungsgremium benennen Teams für die weiteren Phasen benennen |
| Bewertung der Reife | Analyse bestehender Systeme und Prozesse Analyse der Geschäfts- und IT-Strategie Erstellung Organigramm und Systematlas |
| Entwicklung der Zieldefinition | Technisch (Domänen- und Servicekandidaten, Geschäftsprozesse) Organisatorisch (Teamstrukturen, Finanzierung) Erstellung Referenzarchitektur |
| Definition der Roadmap | Technisch (Domänen- und Servicekandidaten, Geschäftsprozesse) Organisatorisch (Teamstrukturen, Finanzierung) Erstellung Referenzarchitektur |

Tabelle 1: Aktivitäten der Roadmap-Definition

Servicekandidaten vermitteln recht konkret, in welche Richtung sich die SOA entwickeln wird. Da noch keine detaillierte Analyse stattgefunden hat, kann dies natürlich nur grob dargestellt werden.

In der letzten Phase wird der Weg vom Ist zum Soll beschrieben. Hier geht es darum, wichtige Meilensteine festzulegen. Dies könnte beispielsweise die Verfügbarkeit der Domänenkunden oder die Abbildung ausgewählter Verkaufsprozesse sein. Zudem werden

ist wichtig, da sich jede Entwicklung an den Zielen des SOA-Vorhabens messen lassen muss. Nur wenn es kontinuierlich gelingt, zu zeigen, welchen Nutzen die aktuelle Entwicklung bringt, lässt sich ein SOA-Vorhaben über einen langen Zeitraum umsetzen.

Danach folgen die obligatorischen Aktivitäten der Planung, der Implementierung und des Betriebs. Unter Umständen wird nicht jede Iteration produktiv in Betrieb gehen, jedoch

Transformation permanent an den Geschäftszielen ausrichtet, kann sie den gewünschten Nutzen bringen. In der Praxis wird dies nicht selten übersehen und es entstehen komplexe Lösungen, die vor allem durch technische Modethemen getrieben werden. Zudem sollte eine SOA nicht im Elfenbeinturm entstehen und/oder von außen verordnet werden. Nur wenn das Paradigma von den Entwicklern akzeptiert und getragen wird, kann es erfolgreich umgesetzt werden. Dies kann wiederum nur gelingen, wenn möglichst früh ein einheitliches Verständnis für das Thema SOA im Unternehmen etabliert wird.

Nicht zuletzt deshalb spielt das SOA-Programm und die SOA Roadmap eine herausragende Rolle bei der Umsetzung einer serviceorientierten Architektur.

Fazit

Im Zuge der SOA-Transformation wird der Übergang von einer projekt- und anwendungszentrierten zu einer serviceorientierten Struktur vollzogen. Im ersten Teil dieser Serie wurde das grundsätzliche Vorgehen bei einer solchen Transformation vorgestellt. Mit dem Abschluss der Transformation ist das SOA-Vorhaben nicht zu Ende, sondern wird im Rahmen des normalen Tagesgeschäfts weiter gelebt. Daraus ergeben sich Konsequenzen für die organisatorischen Strukturen, die im zweiten Teil dieser Serie behandelt werden. ■

Es ist wichtig, die Ziele des SOA-Vorhabens immer vor Augen zu haben.

Kriterien festgelegt, anhand derer das Erreichen eines Meilensteins geprüft werden kann. Es empfiehlt sich ebenfalls, den jeweils angestrebten Nutzen eines Meilensteins zu definieren. Dies hilft, um die permanent auftretende Frage nach der Motivation des SOA-Programms zu beantworten. Das Ergebnis der letzten Phase ist die konkrete Roadmap, die den groben Ablauf der Transformation beschreibt.

Umsetzung

Bisher dienten alle Aktivitäten der Vorbereitung der SOA-Umsetzung. Im Rahmen des SOA-Programms wurde die SOA Roadmap entwickelt. Wenn diese freigegeben wird, beginnt die Phase der Umsetzung. Im Vergleich zu den anderen Phasen ist diese ungleich länger, da hier die eigentliche Implementierung erfolgt. Die Umsetzungsphase ist inkrementell und orientiert sich an den Meilensteinen der Roadmap. Jeder Meilenstein kann mehrere Iterationen erfordern. Diese Iterationen beinhalten prinzipiell die Aktivitäten aus Abbildung 4.

Zu Beginn einer jeden Iteration steht die Bewertung des bisher Erreichten. Dabei werden die bisher gemachten Erfahrungen in die neue Iteration einbezogen und gegebenenfalls Kurskorrekturen vorgenommen. Danach folgt eine Potenzialanalyse im Kontext der Iteration. Es wird die Frage beantwortet, welchen Wertbeitrag die Iteration zum Meilenstein beziehungsweise zum SOA-Programm insgesamt leisten kann. Dies

sind kleine Iterationen mit häufiger Produktivsetzung gegenüber langen Iterationen vorzuziehen, da so der geschäftliche Nutzen besser dargestellt werden kann. Mit dem Fortschreiten der Umsetzung entlang der Meilensteine wird die SOA schrittweise umgesetzt. Idealerweise nimmt die Reife der SOA und die Agilität des Unternehmens dabei immer weiter zu.

Die Phase der Umsetzung endet, wenn der letzte Meilenstein der Roadmap erreicht ist. Spätestens zu diesem Zeitpunkt stellt SOA das führende Paradigma im Unternehmen dar.

Erfolgsfaktoren

Es ist wichtig, sich die Ziele des SOA-Vorhabens immer wieder zu vergegenwärtigen. Denn nur wenn sich



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] www.sonicsoftware.com/solutions/service_oriented_architecture/soa_maturity_model/
- [2] www.ibm.com/developerworks/webservices/library/ws-soa-simm/
- [3] dev2dev.bea.com/images/2005/11/maturitymatrix.jpg
- [4] dev2dev.bea.com/pub/a/2005/12/soa-roadmap.html
- [5] www.1.webmethods.com/PDF/datasheets/SOA_Roadmap_datasheet.pdf