

Systemkonsolidierung in fragmentierten Landschaften

Prof. Dr. Georg Herzwurm, Dipl.-Kfm. Stefan Jesse

Lehrstuhl für Allgemeine Betriebswirtschaftslehre und Wirtschaftsinformatik II

Universität Stuttgart

Breitscheidstr. 2c

70174 Stuttgart

herzwurm@wi.uni-stuttgart.de

jesse@wi.uni-stuttgart.de

Abstract: Eine der größten Herausforderungen, der sich Unternehmen heute stellen müssen ist die Integration fragmentierter Netzwerke von Systemen über unzählige regional oder weltweit verstreute Fachabteilungen oder Geschäftsbereiche hinweg. Der zunehmende Kostendruck in der IT und ineffiziente Geschäftsprozesse zwingen die Unternehmen zu einer Systemkonsolidierung. Ein möglicher Lösungsansatz zur Überwindung dieser Ineffizienzen, zur Standardisierung von Applikationen und dem Ausgangspunkt zur Einführung einer Service-orientierten Architektur ist der exklusive Einsatz eines ERP-Systems eines Anbieters, genannt „single-instance“ ERP. In dieser Arbeit wird die Systemkonsolidierung in fragmentierten Landschaften, dargestellt am Beispiel des „single-instance“ ERP-Ansatzes, als Ausgangspunkt für die unternehmensweite Einführung und den Aufbau einer Service-Orientierten Architektur diskutiert.

1 Einleitung

Eine der größten Herausforderungen, der sich Unternehmen heute stellen müssen ist die Integration fragmentierter Netzwerke von Systemen über unzählige regional oder weltweit verstreute Fachabteilungen oder Geschäftseinheiten hinweg. Der zunehmende Kostendruck in der IT und ineffiziente Geschäftsprozesse zwingen die Unternehmen zu einer Systemkonsolidierung. Darüber hinaus machen heutige Geschäftsprozesse einen viel intensiveren Gebrauch von Hardware als dies noch vor einigen Jahren der Fall war: Immer mehr Daten werden verarbeitet und gespeichert. Nebenbei setzt sich der Trend zur Verwendung von Standardsoftware immer weiter fort. Dies hat zur Folge, dass Applikationen und Systeme nie wirklich maßgeschneidert und performanceoptimiert sind, sondern immer angepasst werden müssen bzw. nur durch massiven Hardwareeinsatz schneller gemacht werden können. Diesem steigenden Ressourcenanspruch steht jedoch die Forderung nach sinkenden Investitions- und Entwicklungskosten gegenüber. Um dieser Anforderung gerecht zu werden ist es wichtig die vorhandene Hardware

optimal auszunutzen, Prozesse sinnvoll über einzelne Cluster zu verteilen und Leerlaufzeiten zu vermeiden.

Basis für eine Konsolidierung bildet häufig das „Business Alignment“, bei dem Geschäftsprozesse und IT-Systeme aufeinander abgestimmt werden. Im Zuge einer solchen Systemkonsolidierung reicht ein globales Rollout eines neuen Systems, verbunden mit der Hoffnung, dass dieses neue System alle Probleme löst, nicht aus. Es ist stets auch eine Untersuchung und teilweise Umstrukturierung von Geschäftsprozessen vorzunehmen. Zu beachten ist dabei jedoch, dass die Systemkonsolidierung in fragmentierten Landschaften zwar in erheblichem Umfang von der IT getragen und unterstützt wird, jedoch das gesamte Unternehmen von dieser Umstrukturierung betroffen ist und somit die Unterstützung der Geschäftsleitung ein zwingend notwendiger Erfolgsfaktor ist.

Weder für die erfolgreiche Systemkonsolidierung noch für die Einführung und den Aufbau einer unternehmensweiten Service-Orientierten Architektur (SOA) gibt es eine Checkliste, die ein Unternehmen befolgen bzw. abarbeiten kann. Es gibt jedoch einige grundlegende Regeln, die in erster Linie die Geschwindigkeit bei der Innovation der Geschäftsprozesse betreffen. Ein entscheidender Erfolgsfaktor für Unternehmen ist dabei die Entwicklung eines eigenen Fahrplans für die Innovation ihrer Geschäftsprozesse. Dabei sollten Unternehmen nicht in überzogene Technologie investieren, welche die Anforderungen übersteigt. Vielmehr sollte die Umsetzung spezifischer Geschäftsanforderungen den Beginn darstellen, um tatsächliche Probleme von heute anhand von neuen Technologien und Methoden zu lösen.

2 Systemkonsolidierung als Ausgangspunkt für SOA

Die Wahrnehmung, dass Service-Orientierte Architekturen nur für Großunternehmen praktikabel sei, da Sie für diese zwar eine größere Flexibilität bieten, aber auch gleichzeitig den Tribut einer höheren Komplexität fordern, ist falsch. Es gibt große Wirtschaftszweige, deren Bedarf an einer Service-Orientierten Architektur dringender ist, da Sie mit Altsystemen arbeiten, die offener und flexibler werden müssen. Jedoch ist die SOA auch für kleine und mittelständische Unternehmen gedacht, da die Einführung und der Aufbau der SOA letztendlich schnelle Innovation und flexiblen Wandel ermöglichen soll – und diese Anforderung gilt für große wie für kleine Unternehmen. Die SOA könnte sich für kleinere Unternehmen sogar als noch wichtiger erweisen, denn diese müssen in einem sehr wettbewerbsintensiven Markt überleben [ZE06]. Es stellt sich also folgendes Problem: Gibt es eine Möglichkeit mehr Flexibilität anzubieten und gleichzeitig die Total Cost of Ownership zu verringern?

Für viele Unternehmen bildet die Systemkonsolidierung den Ausgangspunkt für eine neue unternehmensweite Service-Orientierte Architektur. Unternehmen brauchen effiziente IT-Systeme, um ihre Geschäftsprozesse zu verbessern und wettbewerbsfähig zu bleiben. Viele Projekte und Anwendungen erfordern die Nutzung von Geschäftslogik über mehrere Kanäle und Nutzergruppen hinweg. Hier kommt die dienstorientierte IT-

Architektur ins Spiel. Eine SOA modelliert das gesamte Unternehmen als eine Ansammlung von Services, welche über das Unternehmen verteilt und jedem zugänglich sind. Große monolithische Systeme werden in kleinere Teilsysteme zerlegt, wobei diese Teilsysteme ihrerseits wiederum Komponentencharakter besitzen. Der Aufruf dieser Komponenten innerhalb der SOA geschieht ausschließlich über öffentlich bekannte Standardprotokolle. Eine der bisher am weitesten verbreiteten Implementierungsformen solcher SOAs sind die Web Services, andere mögliche Implementierungsformen sind Enterprise Java Beans, oder auch CORBA, es existieren jedoch auch .NET-Implementierungen. Die allgemeine SOA kann als eine 3-Tier-Architektur angesehen werden [MA06], welche in Abbildung 1 dargestellt ist.

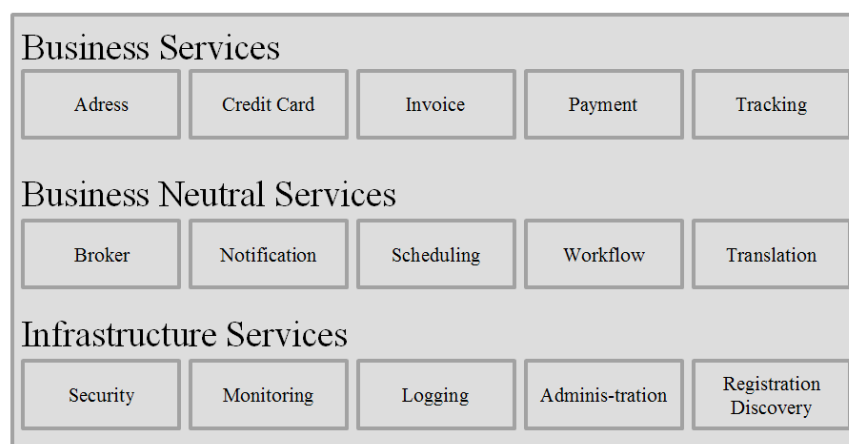


Abb. 1: Basis-SOA-Komponenten (in Anlehnung an [MA06])

In diesem Kontext werden die Schichten nicht als Schichten, sondern als Service Layers bezeichnet. Neben diesen statischen Funktionalitäten haben SOA ein zweites Charakteristikum, ihre Dynamik. Innerhalb der SOA werden alle Services nicht statisch, sondern ausschließlich dynamisch gebunden. Die Folge hiervon ist die Notwendigkeit, die Komponenten Publishing (Unternehmensweite Bekanntmachung der Fähigkeiten und Existenz eines Services), Finding/Discovery (Finden eines Services) und Binding (Nutzung eines aufgefundenen Services) zu etablieren. Dabei beschreibt die SOA eine Software-Infrastruktur, in der die wesentlichen Funktionen einer Anwendung bzw. Softwaremodule als Services organisiert sind. Services können beliebig verteilt sein und lassen sich dynamisch zu Geschäftsprozessen verbinden. SOA legt hierbei die Schnittstellen fest, über die andere Systeme via Netzwerk diese Dienste nutzen können. Services tauschen dadurch unabhängig von den zugrunde liegenden technischen Plattformen Daten aus. Zwingende Abhängigkeiten der monolithischen Architekturen und zwischen bestimmten Client-/Server-Architekturen sind damit aufgelöst [MA06].

In gewissem Sinne wird die SOA-Anwendung dabei unsichtbar sein. In einer traditionellen Anwendung gab es eine technische Ebene – die Datenmodelldefinition – und eine greifbare Ebene – die Benutzeroberfläche. Irgendwo zwischen diesen beiden Ebenen war die Applikationslogik angesiedelt. Die Benutzeroberfläche war sehr eng mit

den Business-Funktionen dieser Anwendung verwoben. Die Benutzeroberfläche war in gewisser Weise die Anwendung. Manchmal war das Datenmodell an der Oberfläche sichtbar und dadurch für den Benutzer unattraktiv. In einer echten SOA-Anwendung werden alle greifbaren, sichtbaren Teile der Anwendung von der Geschäftslogik getrennt. Die Geschäftslogik stellt Funktionen über Services bereit. Einer der ersten Punkte, an dem Services konsumiert werden, ist die Benutzeroberfläche [ZE06]. Ein klarer Vorteil liegt somit darin, dass es nicht nur eine vordefinierte Oberfläche gibt, die die Geschäftslogik darstellt.

Einer der entscheidenden Erfolgsfaktoren, aber auch gleichzeitig eine der größten Herausforderungen einer SOA ist, dass es auf allen Ebenen Unternehmen beeinflusst - sowohl technisch als auch organisatorisch. Neben den Ebenen der Strategie, der Geschäftsprozesse und der Anwendungen ist auch die IT-Basistechnik betroffen. Obwohl SOA primär ein technisches Konzept ist, stellt die Einführung größere Anforderungen an das Management, als viele andere klassische IT-Projekte. Dies liegt vor allem daran, dass SOA die Interoperabilität vieler einzelner Komponenten verlangt. Da diese Komplexität in der Planung berücksichtigt werden muss, verlangt sie in jedem Fall in einer aus Services bestehenden Architektur die enge Zusammenarbeit zwischen Fachabteilungen und der IT.

Für Unternehmen ist die Umsetzung einer SOA eine Herausforderung und Chance zugleich. Sie stehen einerseits vor der Aufgabe, ihre bestehenden Applikationen in SOA-fähige Software umzubauen, oder besser gesagt zu zerlegen. Andererseits profitieren sie davon, dass eine grundlegende Anforderung an SOA besagt, dass Software besser zu pflegen und zu warten ist, um einzelne Bausteine einfacher aktualisieren zu können und nicht mehr die gesamte Applikation redesignen zu müssen. Die Mehrzahl der führenden ERP-Hersteller ist mittlerweile auf den SOA-Zug aufgesprungen, so dass eine Systemkonsolidierung in fragmentierten ERP-Landschaften als ein möglicher Ausgangspunkt für die Einführung und den Aufbau einer unternehmensweiten SOA gesehen werden kann.

3 Systemkonsolidierung in fragmentierten ERP-Landschaften

ERP-Systeme haben sich in den vergangenen Jahren zum Backbone der Unternehmensorganisation entwickelt. In einer Vielzahl von Unternehmen sind fast alle Unternehmensbereiche und -aktivitäten miteinander verknüpft. In den Systemen werden geschäftskritische Prozesse vorgehalten und ein Ausfall der Systeme kann zu Produktionsstillständen und Umsatzausfällen in erheblichem Maße führen. Betrachtet man jedoch die Systemlandschaft der Unternehmen, so zeigt sich schnell, dass diese Erkenntnis zwar vorhanden ist, sich jedoch meist nicht in einer gut strukturierten und durchdachten Strategie hinsichtlich des Einsatzes und Betriebs von ERP-Systemen widerspiegelt. Gerade in großen Unternehmen findet sich oft ein historisch gewachsenes heterogenes und hochkomplexes Applikations-Portfolio [CO06]. Dieses Portfolio wird durch Fusionen und taktische Übernahmegeschäfts oftmals noch komplexer. So fragmentierte Systemlandschaften stehen der wettbewerbsfähigen, marktorientierten

Weiterentwicklung eines Unternehmens im Weg. Aber auch in kleinen und mittelständischen Unternehmen gibt es breit gefächerte Applikations-Portfolios, welche auch hier die wettbewerbsfähige, marktorientierte Weiterentwicklung hemmen.

Betrachtet man die Kosten des Einsatzes und Betriebs dieser heterogenen Applikations-Portfolios, so wird offensichtlich, dass zahlreiche Lizenzverträge mit suboptimalen Anwenderzahlen die laufenden Ausgaben belasten. Auch die Tätigkeiten der IT-Firewehrleute, welche ausschließlich mit der Beseitigung von Inkonsistenzen und der Schnittstellengenerierung –oder Wartung der heterogenen Systeme beschäftigt sind, belasten das IT-Budget. Gerade die erheblichen Schnittstellen-Aufwände in laufenden und anstehenden Projekten bei der Abbildung der neu zu implementierenden Prozesse sind ein immenser Kostenpunkt und ein Treiber hinsichtlich einer einheitlichen ERP-Strategie [ZR03], welche sich primär dem Managementkonzept und sekundär dem Systemarchitekturkonzept unterordnet. Ein mögliches Management- und Systemarchitekturkonzept stellt eine SOA dar, in welcher Funktionen in Form von wiederverwendbaren, technisch voneinander unabhängigen und fachlich lose gekoppelten Services implementiert werden.

Ein möglicher Lösungsansatz innerhalb der Systemkonsolidierung zur Überwindung dieser Ineffizienzen, zur Standardisierung von Applikationen und dem Ausgangspunkt zur Einführung und dem Aufbau einer unternehmensweiten SOA ist der (meist) exklusive Einsatz eines ERP-Systems eines Anbieters, genannt „single-instance“ ERP. Der single-instance ERP-Ansatz bedeutet, dass die Installation der Applikation direkt an eine einzige Datenbank gebunden ist. Dadurch kann vor allem die Komplexität der IT-Landschaft reduziert, aber auch operationale Kosteneinsparungen erzielt werden. Dabei kann die Datenbank selbstverständlich geclustert und auf verschiedenen technisch konsolidierten Infrastrukturkonfigurationen, aber auch in einer virtualisierten Infrastruktur, betrieben werden.

3.1 Systemkonsolidierung mittels Single-Instance ERP

Durch den single-instance ERP-Ansatz können operationale Kosteneinsparungen in den Bereichen Integration, Interface, Training sowie Support und Hardware realisiert werden. Darüber hinaus kann ein potentieller Nutzengewinn durch Prozessverbesserungen, bessere Datenkonsistenz und eine bessere Nutzung von vorhandenen Informationen erreicht werden [ZR03].

Ein single-instance ERP-System einzuführen ist jedoch alles andere als einfach und stößt relativ schnell an organisatorische und technische Grenzen. Die Einführung eines solchen Systems führt zu erheblichen Störungen und Unterbrechungen der laufenden Prozesse. Weiterhin stößt die Einführung an organisatorische Grenzen, da von den Benutzern geschätzte Systeme ersetzt werden und es so zu einer Ablehnungshaltung seitens der Mitarbeiter kommen kann. Selbst wenn der Business Case augenscheinlich die Einführung eines solchen single-instance ERP-Systems favorisiert, sind einige Hürden zu nehmen, die in manchen Organisationen zu Mauern werden, die aus unterschiedlichsten Gründen nicht übersprungen werden können [DO05]. Grundsätzlich kann man die Ziele bei der Einführung eines single-instance ERP-Systems mit der Einführung einer unternehmensweiten SOA vergleichen. Primärziel ist, die historisch gewachsene, heterogene Systemlandschaft effizient an Änderungen im Geschäftsprozess anpassen zu können. Sekundärziele sind die Erreichung von Kostenvorteilen durch schnelle Optimierung, die Möglichkeit einer schnellen Reaktion auf neue Herausforderungen, mittelfristige Einsparungen und die schrittweise Restrukturierung komplexer Anwendungssysteme [ER04].

3.2 Single-Instance-ERP Erfolgsfaktoren und Herausforderungen

Aus einer Untersuchung von Gartner im Jahr 2005 gehen die in der Tabelle dargestellten single-instance ERP Erfolgsfaktoren und Herausforderungen hervor:

	Erfolgsfaktor	Herausforderung
Geschäftsmodell	Einheitliches Kerngeschäft mit übereinstimmenden Prozessen	Verteilte Unternehmensgruppe mit variierenden Geschäftsfeldern
Unternehmenskultur	Zentralisiert mit starkem Leitungsorgan, politisches Diktat in zentraler Ebene	Dezentralisiert, autonome Entscheidungen in den Geschäftseinheiten
Geographische Konzentration	Unternehmen tritt als globale Einheit auf. Operations, Sales und Marketing sind in einzelnen geographischen Einheiten konzentriert	Unternehmen tritt als multinationaler Konzern auf. Operations, Sales und Marketing sind weltweit verteilt
Geschäftsumfeld	Stabil; dramatische Einbrüche oder überdurchschnittliches Wachstum unwahrscheinlich	Dynamisch, schnelles Wachstum oder signifikanter Geschäftsrückgang möglich
Übernahme und Fusionen	Nur für kleinere Einheiten vorgesehen, die sich schnell in die bestehende Infra-	Zusammenschluss mit Unternehmen gleicher Größenordnung wahrschein-

	struktur eingliedern lassen	lich, Verkauf der „Nicht-Kerngeschäftsbereiche“ möglich
Etablierung bestehender Systeme	Wenige unterschiedliche Systeme, überwiegend von einem Hersteller	Viele unterschiedliche Systeme mit mehreren Anbietern potenzieller „Single-Instance Anbieter“
Technologisches Umfeld	Enterprise Level Architektur, Entscheidung an zentraler Stelle	Architektur und Technologie werden in den Geschäftsbereichen entschieden

Tabelle 1: Single-instance ERP Erfolgsfaktoren und Herausforderungen [RA05]

Eine andere Möglichkeit der Entscheidungsfindung kann nach dem Grad der Zentralisierung bzw. Dezentralisierung vorgenommen werden. Der Einsatz von single-instance ERP-Systemen wird eher dort Erfolg versprechend sein, wo ein zentralisiertes Management in einem stabilen Marktumfeld mit standardisierten Prozessen in allen Unternehmensteilen und standardisierten Auswertungsanforderungen agiert, die sich auf einen gemeinsamen Satz von Masterdaten beziehen. Weniger Erfolg versprechend ist der Einsatz einer single-instance ERP-Strategie dort, wo eine autonome Steuerung der Niederlassungen in einem dynamischen Marktumfeld vorherrscht [MI03]. Kennzeichnend für diese Szenarien sind darüber hinaus individuelle Prozesse und individuelle Auswertungsanforderungen in einzelnen Unternehmensteilen, die sich auf mehrere Sätze von Masterdaten oder individualisiertes Datenmanagement stützen.

Welche Möglichkeiten und Optionen nun für die Unternehmen bestehen, bei welchen die Herausforderungen die Erfolgsfaktoren überwiegen, beziehungsweise die zu überspringenden technischen oder organisatorischen Hürden zu hoch sind werden im folgenden dargestellt [MI03].

1. Einsatz einer begrenzten ERP-Anbieter-Kombination und gleichzeitiger Infrastrukturkonsolidierungs-Strategie ohne den ganzheitlichen Transfer hin zu einer single-instance Konsolidierung. Beispielsweise könnten regionale oder spartenspezifische Einheiten konsolidiert werden. Ein mögliches Einsatzszenario wäre hier der Hub&Spoke-Ansatz bei welchem das Unternehmen in der Konzernzentrale SAP R/3 und in den regionalen Niederlassungen Produkte der Microsoft Navision/Dynamics-Reihe einsetzt.
2. Einsatz einer „tiered-business“ Applikations-Strategie, d.h. einer mehrschichtigen Applikations-Strategie. Hier wird versucht, dem Bedürfnis nach zentraler Kontrolle und dem Wunsch nach dezentraler Autonomie bzw. Selbstständigkeit zu entsprechen. Das Ausmaß bzw. die Balance zwischen zentraler Kontrolle und dezentraler Selbstständigkeit wird dabei durch das strategische und organisatorische Unternehmensmodell bestimmt.

4 Fazit

Die tiered-business Strategie ist gut geeignet, trotz des Einsatzes mehrerer Geschäftsapplikationen, ein gemeinsames Vorratslager (Repository) an Geschäftsprozessen über die gesamte Unternehmung hinweg aufzubauen. Weiterhin erlaubt diese Strategie den Unternehmen einige der Vorteile einer gemeinsamen Geschäftsapplikation oder single-instance Strategie zu realisieren und gleichzeitig ein angemessenes Verhältnis der Total Cost of Ownership einzuhalten [DOR05]. Eine solche tiered-business Strategie kann als Ausgangspunkt für eine unternehmensweite SOA dienen, da Kernelemente der SOA bereits durch den single-instance ERP-Ansatz realisiert werden. Bevor sich Unternehmen jedoch für eine der Strategien entscheiden, sollten diese eine umfassende Analyse und Machbarkeitsstudie hinsichtlich der Erfolgsfaktoren und der Herausforderungen vornehmen, da die Entscheidung für eine falsche Strategie erhebliche Kosten und Zeit beansprucht. Darüber hinaus kann die Entscheidung für eine falsche Strategie unter Umständen unumkehrbar sein.

Durch eine Systemkonsolidierung in fragmentierten Landschaften lässt sich in vielen Fällen mehr Flexibilität durch schnellere Reaktionen auf organisatorische und technische Herausforderungen erreichen. Der single-instance ERP-Ansatz stellt eine Möglichkeit der nahezu perfekten Systemkonsolidierung dar und ist somit ein idealer Ausgangspunkt für die Einführung und den Aufbau einer unternehmensweiten SOA, welche bereits während der Systemkonsolidierung im Ansatz aufgebaut wird. Es gibt gute Gründe für die Verfolgung des single-instance ERP-Ansatzes, wie gesetzliche Regelungen durch Sarbanes-Oxley und Basel II, oder kostenreduzierende Initiativen wie die Konsolidierung der IT-Ausgaben. Single-Instance ERP ist jedoch nicht für jedes Unternehmen die richtige Wahl. Die Vorteile des single-instance ERP-Ansatzes, wie Reduzierung der Kosten, standardisierte Geschäftsprozesse und tiefgehende Analysemöglichkeiten müssen sorgfältig gegenüber den Herausforderungen, wie Unabhängigkeit von Geschäftsbereichen, Prozessstandardisierung und existierenden Beschränkungen hinsichtlich des Datenmodells, abgewogen werden. Für große multinationale Konzerne ist eher die Konsolidierung auf einige wenige Systeme ein realistisches Ziel, wohingegen kleine und mittelständische Unternehmen durch den Einsatz eines single-instance ERP-Systems die größtmögliche Flexibilität in Übereinstimmung mit bestehenden Kostenzielen erhalten, ohne dabei die Geschäftsbereiche zu sehr zu behindern oder zu beschneiden.

Literaturverzeichnis

- [CO06] Conrad, S./Hasselbring, W./Koschel, A./Tritsch, R.: Enterprise Application Integration: Grundlagen – Konzepte – Entwurfsmuster – Praxisbeispiele; Spektrum Akademischer Verlag, 2006
- [DO05] Dostal, W./Jeckle, M./Melzer, I./Zengler, B.: Service-orientierte Architekturen mit Web Services: Konzepte – Standards – Praxis; Spektrum Akademischer Verlag, 2005.
- [DOR05] Dorr, E.: How to Approach Tiered Business Applications; erschienen bei Gartner Research, 04/2005
- [ER04] Erl, T.: Service-Oriented Architecture. A field guide to integrating XML and Web services; Prentice Hall PTR, Upper Saddle River, 2004
- [MA06] Masak, D.: Legacysoftware; Springer-Verlag, 2006
- [MI03] Miller, B.: One Global Enterprise System: When and Why; erschienen bei Forrester Research, 06/2003
- [RA05] Rayner, N./Dorr, E.: When to Consider a Single-Instance ERP Strategy; erschienen bei Gartner Research, 09/2005
- [ZE06] Zencke, P.: Architektur für Geschäftsanforderungen; in Interview mit Uta Springer, erschienen in SAP INFO Nr. 139, S. 27-29.
- [ZR03] Zrimsek, B./Prior, D.: Comparing the TCO of Centralized vs. Decentralized ERP; erschienen bei Gartner Research, 01/2003