

Software as a Service

DOI 10.1007/s11576-008-0095-0

Die Autoren

Prof. Dr. Peter Buxmann
Dipl.-Wirtsch.-Ing. Sonja Lehmann

TU Darmstadt
 Information Systems /
 Wirtschaftsinformatik
 Hochschulstr. 1
 64289 Darmstadt
 Deutschland
 {buxmann | lehmann}@
 is.tu-darmstadt.de

Prof. Dr. Thomas Hess

LMU München
 Institut für Wirtschaftsinformatik
 und Neue Medien
 Ludwigstraße 28
 80539 München
 Deutschland
 thess@bwl.lmu.de

Eingereicht am 2008-02-16, nach zwei
 Überarbeitungen angenommen am
 2008-08-27 durch Prof. Dr. Sinz.

1 Software as a Service: eine Einordnung

Der Bezug von Software als Dienstleistung, kurz „Software as a Service“ (SaaS), gilt derzeit als wichtiger IT-Trend, und Branchenanalysten prognostizieren steigende Umsätze sowie ein anhaltendes Wachstum (Cloer 2007, S. 1). Bei der Nutzung von SaaS wird den Kunden eine Standardsoftwarelösung als Dienstleistung über das Internet zur Verfügung gestellt. Der SaaS-Anbieter ist für Betrieb und Wartung der mehrmandantenfähigen Software verantwortlich. Dabei erzielen die Anbieter keine Lizenzeinnahmen. Vielmehr zahlen die Anwender für die gemieteten Softwarekomponenten und Serviceleistungen Nutzungsgebühren, die in der Regel monatlich, quartalsbasiert oder jährlich anfallen. Darüber hinaus sind aus Sicht der Software- und Serviceanbieter auch andere Erlösmodelle, wie z. B. Werbeeinnahmen oder eine nutzungsabhängige Abrechnung, denkbar.

Die Idee, die hinter SaaS steht, ist nicht grundlegend neu. Bereits in den 1990er-Jahren wurde ein ähnlicher Ansatz unter der Bezeichnung Application Service Providing (ASP) verfolgt (Günther et al. 2001). Bei SaaS handelt es sich um eine Erweiterung von ASP, die insbesondere aufgrund der Entwicklung und Verbreitung innovativer Internet-Technologien und -Standards viel Potenzial besitzt und neue Möglichkeiten für Anwender und Anbieter eröffnet. So benötigen die Anwender heute in den meisten Fällen lediglich einen Internet-Zugang und einen Webbrowser, um SaaS-Lösungen nutzen zu können. Neue Technologien, wie etwa AJAX (Asynchronous Javascript and XML), sorgen dafür, dass Webseiten nicht bei jedem Aufruf komplett neu aufgebaut, sondern lediglich die aktualisierten Teile nachgeladen werden. Darüber hinaus erleichtern serviceorientierte Architekturen und offene Standards, wie etwa Webservice-Protokolle, grundsätzlich die Integration von SaaS-Lösungen in Inhouse-Systeme sowie eine Zusammenarbeit mit anderen Services.

Der Artikel basiert u. a. auf strukturierten Experteninterviews, die im Zeitraum vom 2007-11-08 bis 2007-12-18 mit zehn Anwenderunternehmen aus verschiedenen Branchen durchgeführt wurden.

2 SaaS: Einsatzgebiete und Beispiele

Grundsätzlich ist SaaS für eine Vielzahl von Einsatzgebieten anwendbar. Besonders geeignet scheint dieses Geschäftsmodell jedoch für Funktionen und Prozesse zu sein, die sich zu einem hohen Grad standardisieren lassen. Gemäß einer Studie von Lünendonk bieten sich insbesondere die Bereiche CRM- sowie ERP-Software für SaaS (Software as a Service) an (Lünendonk 2007, S. 132). Ein weiteres Ergebnis dieser Studie besteht darin, dass dieses Geschäftsmodell insbesondere für mittelständische Kunden interessant ist. Dies gilt gerade, weil kleinere und mittelständische Unternehmen sich im Regelfall sehr stark auf ihre Kernkompetenzen konzentrieren und typischerweise

im Vergleich zu Großunternehmen über ein geringeres IT-Know-how verfügen (Buxmann et al. 2008, S. 92–93; Zencke 2007, S. 122–123).

Im Folgenden werden zwei Beispiele für SaaS dargestellt: Salesforce.com sowie SAP Business ByDesign.

Salesforce.com

Der amerikanische Anbieter salesforce.com bietet SaaS-Lösungen für den Bereich CRM an. Umsätze generiert er aus so genannten „Subscription-Fees“, d. h. monatlichen Nutzungsentgelten, die die Kunden für die Komponenten des CRM-Systems zahlen. Die Höhe des Entgelts richtet sich nach der Anzahl der Anwender, die auf das System Zugriff haben sollen. Dabei haben die Verträge von salesforce.com in der Regel eine Mindestlaufzeit von einem Jahr. Das monatliche Nutzungsentgelt schließt neben den Softwarekomponenten auch die gesamte Infrastruktur sowie den Support und Service über das Internet mit ein. Der Kunde kann zwischen verschiedenen Versionen der CRM-Anwendung wählen, die sich hinsichtlich Funktionsumfang, Grad der Anpassung und Integration sowie Speicherplatz pro Anwender unterscheiden (Buxmann et al. 2008, S. 9–10). Zurzeit erweitert salesforce.com die CRM-Lösung um so genannte AppExchange-Anwendungen. Das Ziel besteht darin, anderen Softwareanbietern die Möglichkeit zu geben, eigene Geschäftsanwendungen in die salesforce.com-Umgebung zu integrieren.

SAP Business ByDesign

Auch die SAP AG bietet unter der Bezeichnung SAP Business ByDesign SaaS-Anwendungen an (Illgner 2008; Zencke 2007). Diese Lösungen basieren auf einer serviceorientierten Architektur, deren Grundlagen die Integrationsplattform SAP NetWeaver und das Enterprise Service Repository (ESR) bilden. Zielgruppe von SAP Business ByDesign sind insbesondere mittelständische Unternehmen, die noch keine integrierte Geschäftsanwendung einsetzen. Das

monatliche Entgelt richtet sich nach der Anzahl der Anwender. Darin enthalten sind Software, Infrastruktur, Service und Support. Zudem gibt es Gruppenpreise für Benutzer, die nur in begrenztem Umfang auf die Software zugreifen, beispielsweise zur Erfassung von Reisekosten.

Neben diesen beiden prominenten Beispielen gibt es eine Vielzahl weiterer SaaS-Lösungen, die am Markt angeboten werden. Gemäß einer Befragung von 52 deutschen Softwareanbietern plant knapp die Hälfte dieser Softwarehäuser, ihre Lösungen zukünftig auch auf SaaS-Basis einzuführen; 26 Prozent haben keine solchen Pläne, 25 Prozent machten keine Angabe (Lünendonk 2007, S. 135).

3 SaaS aus Anwendersicht

3.1 SaaS versus klassische Standardsoftware

Bei SaaS-Lösungen handelt es sich um Standardsoftware. Daher führt die SaaS-Nutzung grundsätzlich zu den gleichen potenziellen Vorteilen, die der Einsatz von Standardsoftware im Vergleich zur Nutzung von Individualsoftware verspricht. Dazu gehören etwa tendenziell geringere Kosten, schnellere Verfügbarkeit sowie ein höherer Reifegrad. Dementsprechend sind mit der Anwendung von SaaS-Lösungen im Vergleich zu Individualsoftware in der Regel jedoch auch die entsprechenden Nachteile von Standardsoftware verbunden. Hierzu gehören insbesondere die höhere Diskrepanz zwischen unternehmensspezifischen Anforderungen und der Softwarelösung sowie fehlende Wettbewerbsvorteile.

Im Folgenden wird ein Vergleich zwischen SaaS-Lösungen und Standardsoftware für betriebswirtschaftliche Aufgabenstellungen auf Basis klassischer Lizenzmodelle (im Weiteren verkürzt als klassische Standardsoftware bezeichnet) durchgeführt. Dabei sollen die mit der Implementierung und Nutzung der verschiedenen Softwarearten verbundenen Kosten untersucht werden.

Bei der Nutzung einer SaaS-Lösung fallen periodisch konstante Kosten für Betrieb, Support und Wartung der Lösung an. Häufig wird daher von einem Mietmodell gesprochen. Auch bei einem Update der Software entstehen bei den meisten Anbietern keine weiteren Kosten. Darüber hinaus ist davon auszugehen, dass

die Anwender zusätzlich zur Softwaremiete Implementierungskosten zu tragen haben. Diese fallen etwa im Rahmen von Projekten zur technischen und organisatorischen Integration der SaaS-Lösung an. Eine besondere Herausforderung besteht in der Integration der SaaS-Lösung in bestehende Inhouse-Systeme, die nur von einem der befragten Experten als weitgehend unkritisch angesehen wird. Demgegenüber haben die Anwender beim klassischen Modell neben den Implementierungskosten einmalige Lizenzgebühren zu zahlen. Zudem sind jährliche Support- und Wartungskosten zu tragen, die in der Regel etwa 20 Prozent der Lizenzkosten betragen. Darüber hinaus sind im klassischen Modell Updatekosten zu berücksichtigen, die etwa alle sieben bis zehn Jahre anfallen.

In der Regel werden die Implementierungskosten (Hardware, Software, Geschäftsprozessanwendungen, Personalkosten), einschließlich Lizenzen, für eine klassische Standardsoftwarelösung höher sein als für eine SaaS-Lösung (Altmann et al. 2007, S. 40). Dies liegt u. a. daran, dass aufgrund der Betriebssystem- und Plattformunabhängigkeit in den meisten Fällen keine oder nur geringe zusätzliche IT-Kosten anfallen, da SaaS auf jeder Plattform funktioniert, für die es einen entsprechenden Webbrowser gibt. Aus diesem Grund ist tendenziell auch von einer schnelleren Verfügbarkeit von SaaS-Lösungen auszugehen.

Dass die Implementierungskosten für SaaS-Lösungen in der Regel niedriger sein werden, liegt aber auch daran, dass die Customizing-Möglichkeiten für diese Lösungen meistens geringer sind als bei klassischen Standardsoftwarelösungen. Im Rahmen der Experteninterviews schätzten acht von zehn Befragten die Anpassbarkeit von Software an die Unternehmensbedürfnisse als sehr wichtig ein. Insofern wird der Kostenvorteil von SaaS-Lösungen durch eine geringere Anpassung an die organisatorischen Anforderungen relativiert.

Darüber hinaus wird häufig als Vorteil für SaaS angeführt, dass diese Lösungen zu einer geringeren Abhängigkeit vom Softwareanbieter führen (Dubey und Wagle 2007, S. 1). Grundsätzlich ließe sich dieser Vorteil mit den tendenziell geringeren Implementierungskosten begründen. Im Folgenden soll dieser Zusammenhang etwas genauer betrachtet werden. Bislang ist zu beobachten, dass für einen Anwender

der Wechsel seiner Standardsoftwarelösung in vielen Fällen mit hohen Switching-Costs verbunden ist. Dies gilt insbesondere für ERP-Systeme, weshalb ein Wechsel des Anbieters in der Praxis relativ selten zu beobachten ist (Buxmann et al. 2008, S. 31–32). Doch was ist der Grund für diese hohen Switching-Costs? In erster Linie sind es nicht die Lizenzkosten für eine alternative Softwarelösung. Vielmehr gilt gerade auf dem ERP-Markt, dass diese Software auch die Geschäftsprozesse der Anwender abbildet und möglicherweise auch gestaltet hat. Ein Wechsel des Anbieters würde deshalb auch erhebliche Kosten für Organisationsänderungen nach sich ziehen. Dieser Zusammenhang ist grundsätzlich auch bei SaaS-Lösungen zu beobachten. Sobald diese Lösungen in die IT-Landschaft der Anwender integriert werden, ist ein solcher Lock-in-Effekt nicht zu vermeiden. Dabei gilt, dass der entsprechende Lock-in und die damit verbundene Abhängigkeit vom Anbieter umso höher sind, je mehr in diese Integration investiert wurde (Buxmann et al. 2008, S. 31). Aufgrund der geringeren Customizing-Möglichkeiten sowie dem häufigen Einsatz offener Standards im Rahmen von serviceorientierten Architekturen ist im Vergleich zu klassischer Standardsoftware tendenziell ein geringerer Lock-in in SaaS-Lösungen zu erwarten. Ein einfacher Anbieterwechsel wird nach Meinung der befragten Experten dadurch erschwert, dass die Migration der Daten als sehr aufwändig eingeschätzt wird.

Zudem sind bei der Entscheidung zwischen SaaS-Angeboten und klassischer Standardsoftware auch die Auswirkungen der unterschiedlichen Strukturen der Zahlungsströme auf Jahresabschluss und Besteuerung zu berücksichtigen (Bernecker 2007, S. 505–506).

3.2 SaaS als Outsourcingentscheidung

Bei der Inanspruchnahme eines SaaS-Angebots handelt es sich um Outsourcing, d. h. die Auslagerung von Funktionen oder Prozessen an Dritte (Amberg und Wiener 2006; Hirschheim et al. 2006; Knolmayer 2007). Dies bedeutet auch, dass sich einige potenzielle Vor- und Nachteile der SaaS-Nutzung grundsätzlich aus denen des Outsourcings ableiten lassen. Hierzu zählen insbesondere die Vorteile der Konzentration auf Kernkompetenzen, höhere Flexibilität, Zugang zu externem Know-how sowie Kosteneinsparungspotenziale einerseits und die Nachteile des Verlusts

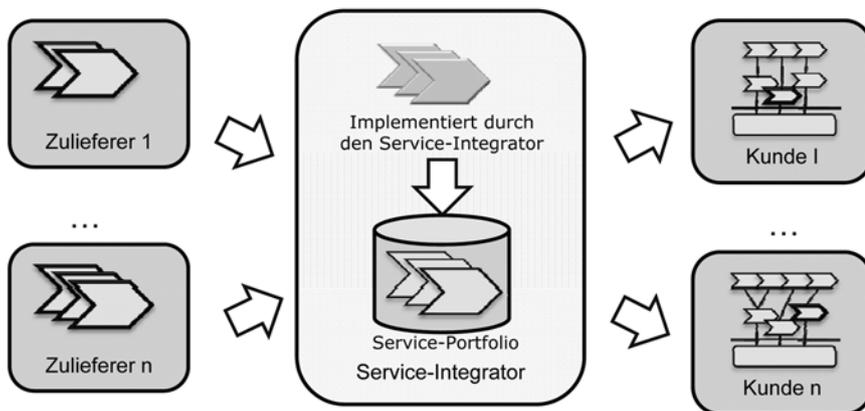


Bild 1 Beispiel für ein mögliches Zulieferernetzwerk in der Softwareindustrie (Lang et al. 2008, S. 3)

von Know-how, Sicherheitsbedenken sowie das Eingehen von Abhängigkeiten andererseits (Amberg und Wiener 2006, S. 37–56).

Als ein zentraler Vorteil von Outsourcing wird in der Regel angeführt, dass sich die Anwender stärker auf ihre Kernkompetenzen konzentrieren können. Dies gilt grundsätzlich auch für die Nutzung von SaaS-Angeboten: Da die Anwender Aufgabenstellungen wie Support, Wartung und Betrieb auslagern, können die auf diese Weise freigesetzten Ressourcen für andere Bereiche eingesetzt werden, etwa um die eigenen Kernkompetenzen zu stärken.

Zudem kann SaaS, wie das Outsourcing anderer Aufgabenstellungen, einen Beitrag dazu leisten, das IT-Management beim Anwender zu vereinfachen und flexibler zu gestalten. Beispielsweise wird der Anbieter die jeweils aktuellste Version der Applikation bereitstellen und es können möglicherweise auch rechtliche Rahmenbedingungen leichter erfüllt werden. Sechs der zehn befragten Experten erwarteten eine höhere Flexibilität durch den Einsatz von SaaS-Lösungen, keiner rechnete mit einer abnehmenden Flexibilität.

Darüber hinaus kann die Auslagerung an einen oder mehrere SaaS-Anbieter Zugang zu intern nicht vorhandenem Know-how eröffnen. Andererseits zeigen viele Erfahrungen in Outsourcingprojekten jedoch auch, dass sich die Aufgabe von Know-how zukünftig möglicherweise als nachteilig erweisen kann, da dadurch die Abhängigkeit von einem Outsourcinganbieter zunimmt. Diese Argumentation gilt grundsätzlich auch bei der Nutzung von SaaS-Angeboten.

Ein weiteres häufig angesprochenes Thema ist die Sicherheit. Hierbei wird insbesondere die Auslagerung interner

sensibler Daten an externe Dritte von den Kunden als riskant eingeschätzt (Graham-Cumming 2007, S. 9).

Die häufig mit Outsourcing verbundenen Kosteneinsparungspotenziale können grundsätzlich auch bei der Nutzung von SaaS-Angeboten realisiert werden, jedoch fehlen dazu bislang noch umfangreiche Erfahrungen.

4 SaaS aus Anbietersicht

4.1 Potenziale und Risiken

Im letzten Abschnitt wurden mögliche Vor- und Nachteile der SaaS-Nutzung für Anwender aus denen des Outsourcings abgeleitet. Ähnlich lässt sich auch anbieterseitig argumentieren. So kann der Betreiber einer SaaS-Lösung, sofern er eine große Anzahl von Kunden besitzt, Skaleneffekte und somit Kostenvorteile realisieren. Dies betrifft die Bereiche IT-Infrastruktur, Hard- und Software sowie Personal. Falls es sich beim Anbieter von SaaS auch um den Hersteller der angebotenen Software handelt, kann dieser zusätzlich Einsparpotenziale in der Entwicklung realisieren. Da die Software lediglich auf seiner Plattform betrieben wird, entfallen die in der Regel aufwändigen Anpassungen hinsichtlich der Kompatibilität mit verschiedenen Betriebssystemen (von Gunten 2007, S. 37; Jacobs 2005, S. 38).

Darüber hinaus kann es tendenziell einfacher werden, den Kunden Produktvarianten zur Verfügung zu stellen. So können beispielsweise auf der gleichen Plattform verschiedene Lösungen, etwa für Großkunden sowie für mittelständische Kunden, angeboten werden.

Für SaaS-Anbieter existieren grundsätzlich natürlich die gleichen Risiken wie für jeden Softwareanbieter (Buxmann et al. 2008, S. 18–19): So besteht ein wesentliches Merkmal digitaler Güter darin, dass die Erstellung der ersten Einheit in der Regel zu hohen Kosten führt, die Reproduktion jedoch zu sehr geringen variablen Kosten möglich ist. Dabei ist keineswegs gewährleistet, dass ein Entwicklungsvorhaben auch tatsächlich zum Erfolg führt. Ein Individualsoftwareanbieter ist daher geneigt, die Finanzierung von Entwicklungskosten sowie das Risiko ganz oder zumindest teilweise auf seinen Auftraggeber abzuwälzen. Einem SaaS- oder Standardsoftwareanbieter bietet sich diese Option hingegen nicht; er trägt das komplette Risiko, kann im Falle des Erfolgs aber auch überproportional profitieren. Bei den Entwicklungskosten handelt es sich darüber hinaus um so genannte „Sunk-Costs“, d. h. um Kosten, die unwiderruflich entstanden sind, nicht mehr beeinflusst werden können und damit auch nicht mehr entscheidungsrelevant sind.

Zudem kann die im letzten Abschnitt angesprochene reduzierte Abhängigkeit der Anwender natürlich ein Risiko für SaaS-Anbieter darstellen.

4.2 Neue Geschäftsmodelle für SaaS-Anbieter

Bislang wurde implizit davon ausgegangen, dass Softwarehersteller ihre eigenen Produkte als SaaS-Lösungen anbieten. Eine Vision könnte aber auch so aussehen, dass Software- und Serviceanbieter zukünftig Services anderer (spezialisierter) Anbieter an Stelle selbst erstellter Services in SaaS-Lösungen integrieren (Buxmann et al. 2008, S. 152–155). Offene Standards, wie etwa Webservice-Protokolle, bieten hierfür grundsätzlich eine gute Voraussetzung. Die Integration von Services anderer Anbieter könnte beispielsweise für Standardsoftwareanbieter interessant sein, die selten nachgefragte Nischenlösungen im Rahmen ihrer Softwarelösung anbieten möchten. Eine Eigenentwicklung dieses „Nischenservices“ könnte sehr aufwändig oder durch einen Mangel an fachlichem Know-how erschwert sein. Eine Alternative besteht jedoch darin, die Entwicklung einem anderen Anbieter zu übertragen, bei dem es sich etwa um ein kleines Softwarehaus handeln könnte.

Theoretisch lässt sich eine solche Entwicklung leicht erklären: Durch die Zerle-

gung der bisher monolithischen Software in über offene Schnittstellen miteinander verbundene Services wird die Voraussetzung dafür geschaffen, Teile der Software (eben die Services) auch von Zulieferern zu beziehen. Würde ein Zulieferer mehrere Kunden bedienen, wäre zu erwarten, dass der von ihm erstellte Service kostengünstiger wird, da die Reproduktionskosten des digitalen Guts Software nahezu Null sind und dies zu einer Fixkostendegression führt. Damit steigt die Wahrscheinlichkeit, dass ein Fremdbezug des Services für das betrachtete Unternehmen vorteilhafter als die Eigenentwicklung wäre.

Gleichwohl ist Software ein Erfahrungsgut, das heißt, der Wert ist vor der Nutzung kaum oder nur sehr schwer zu beurteilen. Dies wird tendenziell dazu führen, dass sich Unternehmen nicht beliebig an einem „Servicemarkt“ bedienen, sondern eher auf längerfristig angelegte Kooperationen setzen. Dabei könnte ein Zulieferernetzwerk entstehen, wie es etwa aus dem Automobilsektor seit Jahren bekannt ist. **Bild 1** illustriert diese Idee an einem einfachen Beispiel.

Allerdings ist derzeit noch unklar, ob ein Softwareanbieter, ein anderer Intermediär oder der Anwender selbst diese Integrationsleistung übernehmen wird. Offen ist auch, welche dieser Konstellationen für den Anwender am vorteilhaftesten ist. In jedem Fall bietet diese Neuordnung der Wertschöpfungskette für Anbieter Chancen, die Rolle eines Intermediärs sowie Integrators einzunehmen und den Kunden integrierte SaaS-Lösungen, die sich aus Services unterschiedlicher Anbieter zusammensetzen, anzubieten.

Literatur

- Altmann, Jörn; Ion, Mihaela; Bany Mohammed; Ashraf Adel* (2007): Taxonomy of Grid Business Models. In: *Veit, Daniel J.; Altmann, Jörn* (Hrsg.): Grid Economics and Business Models, 4th International Workshop, GECON 2007. Berlin, Heidelberg, S. 29–43.
- Amberg, Michael; Wiener, Martin* (2006): IT-Offshoring. Management internationaler IT-Outsourcing-Projekte. Physica, Heidelberg.
- Bernecker, Tobias* (2007): Controlling-Lexikon. Betreibermodelle. In: *Controlling* 19 (8/9), S. 505–506.
- Buxmann, Peter; Diefenbach, Heiner; Hess, Thomas* (2008): Die Software-Industrie. Ökonomische Prinzipien, Strategien, Perspektiven. Springer, Berlin u. a.
- Cloer, Thomas* (2007): Starkes Wachstum für SaaS. In: *Computerwoche*, Nr. 33, S. 1.

- Dubey, Abhijit; Wagle, Dilip* (2007): Delivering software as a service. *McKinseyQuarterly*, May 2007. http://www.leapfrogventuresinc.com/PDF_Files/art_3_Software_as_a_Service-McKinsey.pdf, Abruf am 2008-01-24.
- Graham-Cumming, John* (2007): SaaS is here to stay. http://www.softscan.de/content/de/support/saas_whitepaper, Abruf am 2008-01-08.
- Gunten, Andreas von* (2007): Mit Software als Service zum Erfolg. In: *InfoWeek.ch*, Nr. 02, 2007-02-05, S. 35–38.
- Günther, Oliver; Tamm, Gerrit; Hansen, Lars; Meseg, Thomas* (2001): Application Service Providers: Angebot, Nachfrage und langfristige Perspektiven. In: *WIRTSCHAFTSINFORMATIK* 43 (6), S. 555–567.
- Hirschheim, Rudy; Heinzl, Armin; Dibbern, Jens* (2006): Information Systems Outsourcing. Springer, Heidelberg.
- Illgner, Herbert* (2008): SaaS für den Mittelstand: SAP Business ByDesign, die neue On-Demand-Lösung der SAP. In: *Hess, Thomas* (Hrsg.): Software as a Service: Strategische Perspektiven und praktische Bedeutung. Springer, Berlin u. a., S. 15–24.
- Jacobs, Dean* (2005): Enterprise Software as Service. Online services are changing the nature of software. In: *ACM Queue* 3 (6), S. 36–42.
- Knolmayer, Gerhard F.* (2007): Sourcing-Entscheidungen aus den Perspektiven des Produktions- und Informationsmanagements. In: *Specht, Dieter* (Hrsg.): Insourcing, Outsourcing, Offshoring. Beiträge zur Produktionswirtschaft. Gabler, Wiesbaden, S. 1–30.
- Lang, J. Christian; Widjaja, Thomas; Buxmann, Peter; Domschke, Wolfgang; Hess, Thomas* (2008): Optimizing the Supplier Selection and Service Portfolio of a SOA Service Integrator. In: *Proceedings of the 41th Hawaii International Conference on System Sciences (HICSS 2008)*, Hawaii, S. 1–10.
- Lünendonk, Thomas* (2007): Führende Standard-Software-Unternehmen in Deutschland. Studie der Lünendonk GmbH, Bad Wörishofen.
- Zencke, Peter* (2007): Einsatz von Standardsoftware im Mittelstand. In: *WIRTSCHAFTSINFORMATIK* 49 (Sonderheft), S. S122–S124.

Hier steht eine Anzeige.

 Springer