

Die Softwareindustrie

Peter Buxmann · Heiner Diefenbach
Thomas Hess

Die Softwareindustrie

Ökonomische Prinzipien,
Strategien, Perspektiven

Professor Dr. Peter Buxmann
Technische Universität Darmstadt
Information Systems/Wirtschaftsinformatik
Hochschulstraße 1
64289 Darmstadt
Buxmann@is.tu-darmstadt.de

Professor Dr. Thomas Hess
Ludwig-Maximilians-Universität München
Institut für Wirtschaftsinformatik
und Neue Medien
Ludwigstraße 28 VG
80539 München
thess@bw1.lmu.de

Dr. Heiner Diefenbach
TDS AG
Konrad-Zuse-Straße 16
74172 Neckarsulm
Heiner.Diefenbach@tds.de

ISBN 978-3-540-71828-4

e-ISBN 978-3-540-71830-7

DOI 10.1007/978-3-540-71830-7

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

© 2008 Springer-Verlag Berlin Heidelberg

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funk-sendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Herstellung: LE-TEX Jelonek, Schmidt & Vöckler GbR, Leipzig

Einbandgestaltung: WMX Design GmbH, Heidelberg

Gedruckt auf säurefreiem Papier

9 8 7 6 5 4 3 2 1

springer.com

Vorwort

Kaum eine andere Branche hat die Gesellschaft und die Unternehmenswelt so nachhaltig verändert wie die Softwareindustrie. So ist die Softwareunterstützung von inner- sowie zwischenbetrieblichen Geschäftsprozessen heute genauso eine Selbstverständlichkeit wie das „Googeln“ nach Informationen oder die Nutzung von Navigationsgeräten. Es ist daher auch wenig überraschend, dass die Softwareindustrie gemäß IDC bei einem weltweiten jährlichen Umsatz von 230 Milliarden US-Dollar (ohne komplementäre Dienstleistungen) und mit einem Zuwachs von jährlich sieben Prozent in Europa sowie bis zu neun Prozent in Asien zu den bedeutendsten Wachstumsmärkten zählt. Dabei gilt, dass die Softwarebranche so international wie kaum eine andere ist. Dementsprechend konkurrieren Softwareunternehmen weltweit um Kunden und zunehmend auch um Mitarbeiter. Aber nicht nur aufgrund der internationalen Softwaremärkte, sondern auch wegen der besonderen Eigenschaften des Gutes Software gelten für Softwareanbieter eigene Spielregeln. Hiervon handelt dieses Buch.

Es wäre aufgrund der Besonderheiten und auch der Attraktivität der Branche zu erwarten, dass es – ähnlich wie etwa für den Industrie- und Bankensektor, für Versicherungen oder für die Medien- und Energiewirtschaft – eine spezielle Betriebswirtschaftslehre oder Lehrstühle für die Softwareindustrie aus der Wirtschaftsinformatik heraus gibt. Dies ist bisher jedoch überraschenderweise nicht der Fall. Diese Lücke war für uns ein Anlass, die Software Economics Group Darmstadt-München (www.software-economics.org) zu gründen sowie das vorliegende Buch zu schreiben.

Im ersten Kapitel werden wir nach einer kurzen Beschreibung der Anfänge der Softwareindustrie auf die verschiedenen Player in dieser Branche eingehen. Einen Schwerpunkt bildet die Untersuchung der unterschiedlichen Erlösquellen, also das Lizenz- und Servicegeschäft. Die ökonomischen Prinzipien der Softwareindustrie werden im zweiten Kapitel thematisiert. Hierzu gehören die spezifischen Eigenschaften des Gutes Software sowie die besonderen Merkmale von Softwaremärkten. Darauf aufbauend werden im dritten Kapitel Strategieempfehlungen für die Softwareindustrie abgeleitet. Dabei betrachten wir Kooperations-, Preis-, Vertriebs- und In-

dustrialisierungsstrategien. In diesem Zusammenhang werden wir auch neue Technologien, wie das SOA-Konzept und das Model Driven Engineering, untersuchen. Dabei liegt der Schwerpunkt allerdings eher auf der Analyse der Auswirkungen auf die Softwareindustrie als auf der Darstellung technischer Implementierungsdetails. Aufgrund des hohen Internationalisierungsgrads der Branche gehen wir im vierten Kapitel auf die Globalisierung der Softwareentwicklung ein. Dabei werden wir uns mit Offshore- sowie Open-Source-Projekten und ihren Auswirkungen auf die Softwareindustrie beschäftigen.

Bei der Erstellung des Buches konnten wir auf die Arbeiten der bereits oben angesprochenen Software Economics Group Darmstadt-München zurückgreifen. Im Rahmen dieses Projekts ist auch das empirische Material entstanden, das eine wichtige Grundlage verschiedener Abschnitte ist. So haben wir eine Vielzahl von Expertengesprächen geführt. Wir möchten uns sehr herzlich bei den folgenden Personen aus der Softwareindustrie dafür bedanken, dass sie ihr Wissen und ihre Erfahrungen mit uns geteilt bzw. uns auf andere Weise unterstützt haben: Dieter Berz, Rüdiger Bolt, Christian Boos, Debjit Chaudhuri, Akhil Eswaran, Alexander Gassmann, Martin Haas, Jürgen Henn, Martin Helling, Lutz Heuser, Franz Hollich, Kim Langer, Horst Kinzinger, Thomas Lünendonk, Knut Manske, Thomas Meyer, Karsten Öhler, Jens Pfennig, Jürgen Powik, Nikolai Puntikov, Ganesh Ramamoorthy, Andreas Reinicke, Stefan Ried, Morten Rolfes, Thomas Roth, Karl-Heinz Streibich, Christopher Sürie, Ferenc Szilágyi, Justin Vaughan-Brown, Joachim Voegelé und Martin Winkler.

Unseren Mitarbeiterinnen und Mitarbeitern Björn Brandt, Julia Gebele, Markus Hahn, Patrick Johnscher, Sven Schade, Jochen Strube, Thomas Widjaja und Christian Wolf danken wir für ihre fachliche Unterstützung bei der Erstellung dieses Buches.

Bei Anette von Ahsen, Melanie Hüttenberger und Gerrit Pohl bedanken wir uns sehr herzlich für die vielen und umfangreichen „Bug Reports“. Ebenso dankbar sind wir Niels Thomas vom Springer-Verlag für die sehr angenehme Zusammenarbeit und die vielen guten Verbesserungsvorschläge.

Wir freuen uns auf Ihre Kommentare und Anmerkungen und wünschen Ihnen viele Anregungen und auch viel Spaß beim Lesen des Buchs.

Peter Buxmann, Heiner Diefenbach, Thomas Hess im September 2007

Inhaltsverzeichnis

1	Die Spielregeln in der Softwareindustrie.....	1
1.1	Software und Softwaremärkte: Ausgewählte Besonderheiten der Softwareindustrie im Überblick.....	1
1.2	Die Anfänge der Softwareindustrie.....	2
1.3	Softwareanbieter im engeren und weiteren Sinne.....	3
1.4	Erlösquellen für Softwareanbieter.....	11
1.4.1	Lizenzmodelle und -erlöse.....	11
1.4.2	Dienstleistungserlöse in der Softwareindustrie.....	14
2	Ökonomische Prinzipien der Softwareindustrie.....	18
2.1	Eigenschaften digitaler Güter.....	18
2.2	Netzeffekte auf Softwaremärkten: The Winner takes it all.....	20
2.2.1	Netzeffekte: Grundlagen und Definitionen.....	21
2.2.2	Auswirkungen von Netzeffekten auf Softwaremärkte.....	24
2.2.3	Struktur von Softwaremärkten.....	28
2.2.4	Netzeffekte als Wettbewerbsfaktor.....	31
2.2.5	Ein Anwendungsbeispiel: Zweiseitige Netzeffekte und Plattformstrategien in der digitalen Spieleindustrie.....	33
2.2.6	Grenzen der Netzeffekttheorie.....	37
2.3	Das Standardisierungsproblem.....	39
2.3.1	Ansatz und Hintergründe.....	39
2.3.2	Das zentrale Standardisierungsproblem als Optimierungsproblem.....	43
2.3.3	Das dezentrale Standardisierungsproblem – Eine spieltheoretische Darstellung.....	46
2.3.4	Das Standardisierungsproblem – Lessons learned.....	49
2.4	Transaktionskostentheorie: Auf der Suche nach den Grenzen eines Softwareunternehmens.....	51
2.4.1	Ansatzpunkt und Elemente der Transaktionskostentheorie.....	51
2.4.2	Arbeitsteilung zwischen Unternehmen aus Sicht der Transaktionskostentheorie.....	54

2.4.3	Strukturelle Veränderungen der Transaktionskosten: The Move to the Middle	57
2.4.4	Ausblick: Intermediäre und Transaktionskosten.....	58
2.5	Softwareentwicklung als Agency-Problem: Anreiz- kompatible Entlohnung und effiziente Kontrolle.....	59
2.5.1	Principal-Agent-Beziehungen.....	60
2.5.2	Anreizkompatible Vergütungsschemata	62
2.5.3	Kontrollsysteme	67
3	Strategien für Softwareanbieter	69
3.1	Kooperations- und Übernahmestrategien.....	69
3.1.1	Kooperationen in der Softwareindustrie	70
3.1.2	Mergers & Acquisitions in der Softwareindustrie	80
3.2	Vertriebsstrategien	88
3.2.1	Gestaltung des Vertriebssystems: Organisation und Vertriebswege in der Softwareindustrie.....	89
3.2.2	Gestaltung der Beziehungen zu Vertriebspartnern und Key Accounts.....	95
3.2.3	Kennzahlensysteme als Instrument für das Vertriebscontrolling in der Softwareindustrie	97
3.2.4	Gestaltung der Verkaufsaktivitäten.....	103
3.3	Preisstrategien	110
3.3.1	Grundüberlegungen	110
3.3.2	Preisdifferenzierungsstrategien.....	112
3.3.3	Aggregationsstrategien	116
3.3.4	Dynamische Preisstrategien.....	122
3.3.5	Lizenzierungsmodelle und Preisstrategien	125
3.3.6	Ansätze zur Preissetzung für Individualsoftware- anbieter	127
3.4	Industrialisierungsstrategien	130
3.4.1	Industrialisierung als Managementkonzept	131
3.4.2	Erfolg versprechende Ansatzpunkte	134
3.4.3	Serviceorientierte Architekturen: Grundlagen und Aus- wirkungen auf die Industrialisierung in der Software- branche.....	142
4	Globalisierung der Softwareentwicklung: Offshoring und Open-Source-Projekte.....	156
4.1	Outsourcing und Offshoring der Softwareentwicklung	157
4.1.1	Formen des Outsourcings und Offshorings	158
4.1.2	Der Markt für Outsourcing und Offshoring.....	162
4.1.3	Motive für Outsourcing und Offshoring	165

4.1.4	Standortwahl von Softwareanbietern	170
4.1.5	Nearshoring versus Farshoring: Die Entfernung zum Kunden als Erfolgsfaktor?	173
4.2	Open Source Software und die Softwareindustrie	182
4.2.1	Charakteristika von Open Source Software	182
4.2.2	Open-Source-Projekte: Prinzipien und Motivation der Softwareentwickler	189
4.2.3	Open Source Software aus Sicht des Anwenders	193
4.2.4	Strategien kommerzieller Softwareanbieter.....	195
4.2.5	Open Source Software für ERP – Erste Überlegungen....	198
Literaturverzeichnis		201
Sachregister		213

Abkürzungsverzeichnis

BOT	Build Operate Transfer
BSA	Business Software Alliance
BSD	Berkeley Software Distribution
CASE	Computer Aided Software Engineering
CIM	Computational Independent Model
CRM	Customer Relationship Management
CVS	Concurrent Versions System
EDI	Electronic Document Interchange
EPK	Ereignisgesteuerte Prozesskette
ERP	Enterprise Resource Planning
GE	Geldeinheiten
GNU	GNU is not Unix
GPL	General Public License
IE	Internet Explorer
ITT	Indian Institute of Technology
KPI	Key Performance Indicator
LGPL	Library/Lesser General Public License
LoC	Lines of Code
M&A	Mergers & Acquisitions
MDE	Model Driven Engineering
MPL	Mozilla Public License
OSI	Open Source Initiative
OSS	Open Source Software
PDF	Portable Document Format
PIM	Platform Independent Model
PSM	Platform Specific Model
ROI	Return on Investments
SaaS	Software as a Service
SIIA	Software & Information Industry Association
SCM	Supply Chain Management
SOA	Serviceorientierte Architektur
SQL	Structured Query Language
THES	The Times Higher Education Supplement
UDDI	Universal Description, Discovery, and Integration

URI	Uniform Resource Identifier
WSDL	Web Services Description Language
W3C	World Wide Web Consortium
ZB	Zahlungsbereitschaft

1 Die Spielregeln in der Softwareindustrie

1.1 Software und Softwaremärkte: Ausgewählte Besonderheiten der Softwareindustrie im Überblick

Die Softwareindustrie unterscheidet sich grundsätzlich von anderen Branchen. Dies ist zum einen auf die spezifischen Eigenschaften des Produkts Software, zum anderen auf die Struktur von Softwaremärkten zurückzuführen.

Eine Besonderheit von *Softwareprodukten* besteht darin, dass sie sich, wie jedes andere digitale Gut auch, zu geringen Kosten reproduzieren lassen. Die variablen Kosten gehen also gegen Null. Diese Kostenstruktur führt etwa dazu, dass das Lizenzgeschäft eines Softwareanbieters – zumindest auf den ersten Blick – in der Regel profitabler ist als das Servicegeschäft, wie wir noch ausführlich diskutieren werden. Darüber hinaus kann Software beliebig häufig und ohne Qualitätsverluste kopiert werden. Ist eine Kopie im Internet erst einmal im Umlauf, lassen sich Urheber- bzw. Verfügungsrechte faktisch nicht mehr durchsetzen. Dies gilt insbesondere für wenig erklärungsbedürftige Produkte auf Business-to-Consumer-Märkten. Es ist zudem relativ einfach, von einem einmal erstellten Softwareprodukt verschiedene Versionen oder Bündel zu erstellen und diese zu unterschiedlichen Preisen an verschiedene Kundengruppen zu verkaufen.

Auch für *Softwaremärkte* gelten einige Besonderheiten. Wie kaum eine andere Branche ist die Softwareindustrie durch eine starke Internationalisierung gekennzeichnet. Software lässt sich global verteilt entwickeln und in Sekundenschnelle zu vernachlässigbaren Kosten über das Internet vertreiben. Daraus resultiert auch ein weltweiter Wettbewerb zwischen den Softwareanbietern. Im Vergleich zu anderen Branchen spielt hierbei der Heimvorteil auf den nationalen Märkten der Anbieter in vielen Segmenten nur noch eine untergeordnete Rolle. So erzielten deutsche Softwareanbieter im Durchschnitt mehr als die Hälfte ihrer Umsätze im Ausland. Dabei beträgt der Exportanteil am Umsatz der beiden größten deutschen Softwareanbieter SAP und Software AG beispielsweise 80 bzw. 85 Prozent (Lünnendonk 2007, S. 52 f.). Zudem führt der Netzeffektcharakter von Soft-

ware dazu, dass es sich bei Softwaremärkten häufig um so genannte „Winner-takes-it-all“-Märkte handelt. Vor diesem Hintergrund lässt sich beispielsweise auch die Vielzahl von Unternehmensübernahmen erklären.

Diese und andere spezielle ökonomische Prinzipien und Spielregeln werden wir aufgreifen und ausführlich untersuchen. Sie bilden letztlich die Grundlage für die Formulierung von Strategien und Geschäftsmodellen für die Softwareindustrie.

Doch zunächst wollen wir uns in aller Kürze mit der historischen Entwicklung der Softwareindustrie beschäftigen.

1.2 Die Anfänge der Softwareindustrie

Die Softwareindustrie ist eine relativ junge Branche. Die Anfänge gehen auf die frühen fünfziger Jahre zurück, als es noch üblich war, Software und Hardware zu bündeln und gemeinsam zu verkaufen. Die Software war damals also integrierter Bestandteil der Hardware und es wurde noch ausschließlich von Programmcode gesprochen – der Begriff Software wurde erstmals im Jahr 1959 verwendet (Campbell-Kelly 1995, S. 80). In dieser Zeit entstanden in den USA die ersten kleineren Firmen, die Programmcode bzw. Software im Rahmen von individuellen Auftragsprojekten entwickelten (Hoch et al. 2000, S. 27 f.).

Eine Aufwertung der Software erfolgte 1969, als das amerikanische Justizministerium von IBM den getrennten Ausweis von Hardware und Software auf Rechnungen verlangte. In den siebziger Jahren entstand daraufhin eine Reihe von Unternehmen, die sich ausschließlich mit der Entwicklung von Software beschäftigten. Hierbei ist natürlich in erster Linie Microsoft zu nennen: Das von Bill Gates zusammen mit Paul Allen gegründete Unternehmen begann anfangs Programmiersprachen – zunächst BASIC, dann weitere, wie FORTRAN und COBOL – für verschiedene Prozessoren und Rechner zu entwickeln. Erst später entstand im Rahmen einer Zusammenarbeit mit IBM MS-DOS, das zum Standard für Betriebssysteme wurde und wesentlich zur Verbreitung des Personal Computers beitrug (Ichbiah 1993, S. 91-116). Schließlich entschied Microsoft, auch Anwendungen an-

zubieten und damit in Konkurrenz z. B. zu Lotus zu treten. Bereits 1983 kündigte Bill Gates in der Zeitschrift Business Week an, dass es das Ziel von Microsoft sei, alle PC-Software zukünftig aus einer Hand anzubieten (Ichbiah 1993, S. 141).

Eine parallele Erfolgsgeschichte begann etwa zeitgleich mit den Anfängen von Microsoft im badischen Walldorf, als Dietmar Hopp, Hans-Werner Hector, Hasso Plattner, Klaus Tschira und Claus Wellenreuther ein Unternehmen gründeten, das sich auf die Entwicklung von Software für betriebswirtschaftliche Funktionen und Prozesse spezialisierte – die SAP AG war geboren. Zunächst stellte die Firma Software für Großrechner her, später wurden die Anwendungen für Client-Server-Umgebungen angeboten. Heute ist die SAP das größte europäische Softwarehaus und weltweiter Marktführer im Bereich von Enterprise Resource Planning Software (ERP-Software).

Bereits an diesen beiden Beispielen lässt sich eine Besonderheit der Softwareindustrie erkennen: Am Markt setzt sich häufig nur eine Technologie bzw. ein Anbieter durch. So verdrängte MS-DOS das Betriebssystem CPM ebenso, wie sich das Tabellenkalkulationsprogramm Excel gegen Konkurrenzprodukte wie Lotus 1-2-3 durchsetzte. Mittlerweile ist Microsoft weltweit führender Anbieter von Office-Anwendungen, Browsern und Betriebssystemen. Ebenso findet auch auf dem Markt für betriebswirtschaftliche Software aktuell eine Konsolidierung statt (siehe hierzu Abschnitt 3.1.2). Wir werden auf die Besonderheiten dieser Märkte zurückkommen, wollen uns aber zunächst mit den Akteuren der Softwareindustrie beschäftigen.

1.3 Softwareanbieter im engeren und weiteren Sinne

Betrachten wir im Folgenden die verschiedenen Typen von Softwareanbietern. Hierbei soll zwischen Softwareanbietern im engeren und im weiteren Sinne unterschieden werden.

Die Aufgabe eines *Softwareanbieters im engeren Sinne* ist die Entwicklung von Software. Dies gilt unabhängig davon, um welche Art von Software es sich hierbei handelt. Software lässt sich nach verschiedenen Merk-

malen differenzieren. Ein häufig verwendetes Kriterium ist die Nähe zur Hardware (Mertens et al. 2005). Demnach kann Software in Systemsoftware (z. B. Betriebssysteme), systemnahe Software (z. B. Datenbanksysteme oder Middleware) und Anwendungssoftware (z. B. für Textverarbeitung oder für das Rechnungswesen) unterschieden werden. Ebenfalls lässt sich zwischen Software für kommerzielle und private Anwender unterscheiden. Ein drittes Kriterium zur Klassifikation von Software, das für unsere Betrachtung wesentlich wichtiger ist, ist der Standardisierungsgrad. Als Extremformen sind hier Individualsoftware und Standardsoftware zu unterscheiden.

Individualsoftware wird auf der Basis von spezifischen Anforderungen maßgeschneidert entwickelt, und zwar entweder von den anwendenden Unternehmen selbst – in der Regel in der IT-, in manchen Fällen aber auch in den jeweiligen Fachabteilungen – oder von einem externen Softwarehaus. Die Ausgestaltung der Verträge im Falle der Inanspruchnahme eines solchen Drittanbieters ist jedoch sehr unterschiedlich. Nach nahezu einhelliger Auffassung handelt es sich bei Projekten zur Individualsoftwareherstellung um Werkverträge (Marly 2001). Auf eine ökonomische Untersuchung von Dienst- und Werkverträgen gehen wir in Abschnitt 2.5 ein. Insbesondere in Indien boomt die Branche im Bereich der Individualsoftwareentwicklung. So haben sich einige erfolgreiche Softwareanbieter mit enormen Wachstumsraten etablieren können, wie etwa Tata Consultancy Services, Wipro Technologies, Infosys Technologies sowie Cognizant Technology Solutions. Daneben existieren aber auch viele kleinere oder mittelständische Firmen, die in Hochlohnländern häufig noch national bzw. regional tätig sind. Demgegenüber sind die Individualsoftwareanbieter mit Sitz in Niedriglohnländern in den meisten Fällen global aufgestellt und stehen miteinander im Wettbewerb. Wir werden auf diese Nearshore- und Farshore-Anbieter und die sich daraus ergebenden Auswirkungen auf die Softwareindustrie in Abschnitt 4.1 näher eingehen.

Standardsoftware wird in der Regel für den Massenmarkt entwickelt. Dabei gehen die Anbieter von weitestgehend standardisierten Bedürfnissen der potenziellen Nutzer aus. Im Folgenden wollen wir kurz die Entstehung und Erfolgsfaktoren von Standardsoftware am Beispiel der SAP AG betrachten.

Das System R von SAP

Die SAP AG ist das größte europäische Softwareunternehmen und weltweiter Marktführer im Bereich von ERP-Software. Dabei hat SAP inzwischen über 100.000 Installationen bei mehr als 40.000 Kunden getätigt. Über die Hälfte der weltweit 1.000 größten Unternehmen sind Kunden der SAP AG.

Kernaspekte

Der Erfolg von SAP lässt sich auf drei Kernaspekte zurückführen: die Idee der Entwicklung von Standardsoftware, das Anbieten integrierter Lösungen und Echtzeitverarbeitung. 1981 implementierte das Unternehmen diese Visionen zum ersten Mal mit SAP R/2 und legte seinen Schwerpunkt damit auf die Entwicklung betriebswirtschaftlicher Systeme. 1991 präsentierte SAP das System R/3, das ursprünglich nicht als Ablösung für R/2 gedacht war, sondern dieses als Mittelstandslösung ergänzen sollte. Der generelle Wechsel von Großrechnerlösungen zum Client-Server-Prinzip, der zu dieser Zeit im Gange war, machte R/3 aber gerade für größere Unternehmen interessant und verhalf dem System damit zum Erfolg. Mittlerweile hat SAP seine Produktpalette weiterentwickelt, worauf wir aber an anderer Stelle eingehen.

Standardsoftware

Wie bereits erwähnt, wurde Software anfangs meistens im Rahmen von Individualprojekten entwickelt. Dabei wurden die Software und gegebenenfalls auch die Hardware exakt auf die Anforderungen des Auftraggebers zugeschnitten. Demgegenüber plante SAP von Anfang an die Entwicklung eines standardisierten Systems, das sich mehrfach einsetzen lässt. SAP war eines der ersten Unternehmen, die dieses Konzept konsequent verfolgten.

Integrierte Lösung

Grundlage der Anwendung eines SAP-Systems ist eine integrierte Datenbank, auf die alle Anwendungen zugreifen können. Eine solche integrierte Datenbasis war in Unternehmen in den siebziger Jahren weitgehend unbekannt. Die Folge waren enorme Kosten aufgrund redundanter und inkon-

sistenter Daten. Darauf aufbauend entwickelte SAP nach und nach Module für verschiedene Funktionsbereiche, wie etwa Rechnungswesen und Controlling, Logistik oder Personalwirtschaft. Diese Pakete waren zunächst primär für industrielle Unternehmen konzipiert und konnten einzeln oder auch gemeinsam zusammen mit einer Datenbank gekauft werden.

Im Gegensatz zu Office-Anwendungen, wie sie etwa von Microsoft angeboten werden, zeigte sich, dass viele der betriebswirtschaftlichen Bereiche branchenspezifische Besonderheiten aufweisen. Daher wird von der SAP mittlerweile eine Vielzahl von Branchenlösungen angeboten, auch für den zunächst weniger beachteten Dienstleistungssektor.

Echtzeitverarbeitung von Daten

Bis Ende der siebziger Jahre war es in Unternehmen üblich, mit Lochkarten zu arbeiten, d. h., die Daten wurden über Lochkarten in den Computer eingelesen und erst später verarbeitet. Die Idee der Echtzeitverarbeitung war bei den Gründern von SAP von Beginn an ein Kernaspekt und wurde in allen Systemen umgesetzt. Dies erklärt auch das Kürzel „R“ in den Produktnamen von SAP, das für „Realtime“ steht.

Quellen: Leimbach, T. (2007) Vom Programmierbüro zum globalen Softwareproduzenten. Die Erfolgsfaktoren der SAP von der Gründung bis zum R/3-Boom. In: Zeitschrift für Unternehmensgeschichte 52: 5–34; www.sap.com.

Die Grenze zwischen Individual- und Standardsoftware ist fließend. So kann betriebswirtschaftliche Standardsoftware zumindest bis zu einem gewissen Grad auch an die individuellen Bedürfnisse der Anwender angepasst werden. Allerdings verschlingen Einführungsprojekte, die eine Parametrisierung bzw. ein Customizing sowie gegebenenfalls eine Erweiterung der Software umfassen, häufig Millionenbeträge. Daher ist es für den Anwender in der Regel sinnvoll, lediglich kleinere Anpassungen vorzunehmen, da eine komplette Veränderung der Standardsoftware, bis sie hundertprozentig auf seine Bedürfnisse zugeschnitten ist, sehr teuer ist und zudem zu Problemen beim Umstieg auf Folgeversionen führen kann. Darüber hinaus bieten neuere Ansätze, wie serviceorientierte Architekturen, dem Anwender zumindest grundsätzlich die Möglichkeit, sich die besten Softwarepakete zu Teilbereichen herauszusuchen, diese auf seine Bedürfnisse anzupassen und mit Hilfe einer Integrationssoftware zu einer indivi-

duellen Anwendungslösung zusammenzuführen (siehe hierzu Abschnitt 3.4.3).

Neben Softwareanbietern im engeren Sinne – also den Unternehmen, die Standard- oder Individualsoftware entwickeln – gibt es Anbieter für Dienstleistungen bzw. Services in den späteren Phasen im Lebenszyklus einer Softwarelösung. Wir bezeichnen diese Unternehmen im Folgenden als *Softwareanbieter im weiteren Sinne*. Diese Dienstleistungen umfassen zum einen die Unterstützung von Anwendern bei der Implementierung der Softwarelösungen sowie zum anderen den Betrieb der Produkte. Abbildung 1-1 fasst unsere Klassifikation von Softwareanbietern zusammen.

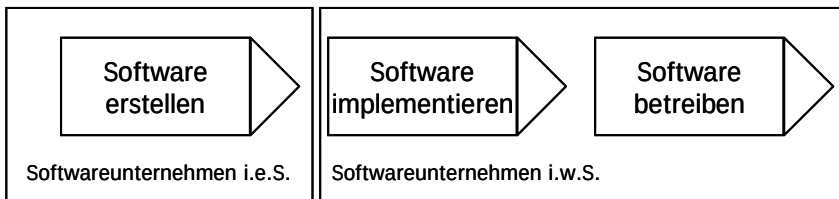


Abbildung 1-1: Klassifikation von Softwareunternehmen

Insbesondere für erklärungsbedürftige Softwarelösungen, also solche, die sich nicht leicht implementieren und in ein Anwendungsumfeld integrieren lassen, existiert eine enorme Nachfrage nach Dienstleistungen. Demzufolge gibt es eine Vielzahl von Anbietern auf diesem Markt. Dabei handelt es sich um (Lünendonk 2006)

- IT-Service-Dienstleister,
- Systemintegratoren,
- Systemhäuser sowie
- Unternehmensberatungen.

Eine besonders große Rolle spielt traditionell die Unterstützung der Anwender bei SAP-Einführungsprojekten. Dies liegt daran, dass – wie bereits angesprochen – die Software im Rahmen dieser Projekte an die speziellen Bedürfnisse der Anwender angepasst wird. Entsprechende Kenntnisse liegen in den Anwendungsunternehmen in vielen Fällen nicht vor oder es stehen dort nicht genügend Mitarbeiter mit einem solchen Know-how zur Verfügung. Diese Einführungsprojekte sind grundlegend für den Nutzen, den die Software für die Kunden stiftet, da sie direkt in die inner- sowie zwischenbetrieblichen Prozesse eingreift. In diesem Zusammenhang sind

häufig auch kleinere Programmierarbeiten, etwa die Entwicklung von Schnittstellen zwischen heterogenen Systemen, zu erledigen.

Dabei setzt dieses Geschäft wie kaum ein anderes eine Vertrauensbasis zwischen den Kunden und den Dienstleistungsanbietern voraus. Detlev Hoch et al. benutzen in diesem Zusammenhang sogar den Begriff des „Glaubens“. Demnach muss der Kunde daran glauben, dass das Dienstleistungsunternehmen bzw. die für das Projekt eingesetzten Mitarbeiter ein hervorragendes Know-how haben und in der Lage sind, ihr Versprechen zu halten, die Probleme der Kunden zu lösen (Hoch et al. 2000, S. 160). Denn während ein Anwender die Möglichkeit hat, die Software eines Standardsoftwareherstellers zu testen, ist dies im Beratungsgeschäft nicht möglich. Dem Kunden bleibt häufig nichts anderes übrig, als seine Entscheidung auf die Referenzen der Beratungshäuser und Systemintegratoren zu stützen. Daher ist es auch das vorrangige Ziel der Marketingaktivitäten von diesen Dienstleistungsunternehmen, Vertrauen in ihre Kompetenz aufzubauen. Hierzu gehören insbesondere die folgenden Maßnahmen (Hoch et al. 2000, S. 162-178):

- Sponsoring von IT-Konferenzen,
- Diskussionszirkel mit hochrangigen Vertretern der IT- und Softwareindustrie,
- Veröffentlichungen in Fachmagazinen und wissenschaftlichen Zeitschriften sowie
- Anzeigen und TV-Spots.

Aufgrund der Komplexität von Einführungs- und Integrationsprojekten ist die Auswahl des Dienstleistungsunternehmens für den Kunden von zentraler Bedeutung. Dies gilt insbesondere auch, weil in den meisten Fällen das Budget überschritten wird bzw. viele dieser Projekte komplett scheitern. So wurden nach einer Untersuchung der Standish Group International Inc. von etwa 30.000 IT-Projekten lediglich 28 Prozent innerhalb der Zeit- und Budgetvorgaben erfolgreich abgeschlossen. 49 Prozent überschritten diese Planwerte erheblich und 23 Prozent wurden überhaupt nicht zu Ende geführt. Bei den als komplex eingestuften Projekten wurden Durchlaufzeit und Budget um mehr als das Doppelte überschritten.

Inzwischen lagern viele Unternehmen auch den Betrieb der Anwendungssoftware an Serviceanbieter aus. Wird die Anwendung direkt von Dritten gehostet, spricht man auch von Software as a Service (SaaS). Dabei zahlt der Nutzer beispielsweise pro User und Monat einen bestimmten Mietbetrag an den Anbieter. Auch wenn dieses Geschäftsmodell keines-

wegs neu ist – so bietet beispielsweise die Datev ihren Kunden solche Lösungen schon seit langer Zeit an – wird es als sehr zukunftsfähig eingeschätzt (Lünendonk 2007). Potenzielle Vorteile für den Anwender bestehen insbesondere in geringeren Setup-Kosten sowie in einer schnelleren Implementierung. Demgegenüber werden SaaS-Lösungen in der Regel nicht so umfangreich an individuelle Kundenwünsche anpassbar sein, wie dies bei ERP-Software heute der Fall ist. Dabei stellt sich die Frage, für welchen Typ von Softwareanwendungen bzw. für welche Branchen SaaS Potenzial besitzt. Gemäß einer Studie von Lünendonk sind insbesondere die Bereiche Customer Relationship Management (CRM)- sowie ERP-Software besonders gut für SaaS geeignet (Lünendonk 2007, S. 132). Ein weiteres Ergebnis dieser Umfrage besteht darin, dass dieses Geschäftsmodell insbesondere für mittelständische Kunden interessant ist. Auf Hintergründe werden wir in Abschnitt 3.2.1 näher eingehen. In Bezug auf die Branchen ergaben sich weniger auffällige Unterschiede, d. h., das Modell scheint grundsätzlich für unterschiedlichste mittelständisch geprägte Branchen interessant zu sein. Darüber hinaus war ein Ergebnis einer Befragung von 52 deutschen Softwareanbietern, dass 49 Prozent der Softwarehäuser planen, für ihre Lösungen SaaS einzuführen, 26 Prozent haben keine solchen Pläne, 25 Prozent machten keine Angabe (Lünendonk 2007, S. 135).

Ein Beispiel für ein solches SaaS-Modell ist das Angebot von salesforce.com, das mittlerweile zu einem starken Wettbewerber für die etablierten CRM-Anbieter SAP und Siebel geworden ist.

Software as a Service bei salesforce.com

Das Geschäftsmodell

Salesforce.com ist ein amerikanischer Application Service Provider, der im Bereich von CRM-Systemen mittlerweile eine starke Marktposition hat. Umsätze werden bei salesforce.com nicht durch den Verkauf von Softwarelizenzen, sondern durch so genannte „Subscription Fees“ für die von den Kunden gemieteten Komponenten des CRM-Systems generiert. Dabei werden die vom Kunden gewünschten Softwarekomponenten zusammen mit der gesamten Infrastruktur, dem Support und Service über das Internet bereitgestellt.

Die Produktpalette von salesforce.com umfasst die Bereiche Vertriebsautomatisierung, Service & Support, Marketing-Automatisierung und

Partner Relationship Management. Salesforce.com bietet seine CRM-Anwendungen in vier verschiedenen Versionen an (Team Edition, Professional Edition, Enterprise Edition und Unlimited Edition).

Preismodell

Der Kunde zahlt für die gemieteten Softwarekomponenten und Serviceleistungen monatliche Nutzungsgebühren. Damit entfallen für die Anwender die oftmals hohen einmaligen Kosten für die Beschaffung der Softwarelizenzen und die Implementierung der Software. Zudem kann auch die Aktualisierung von Software-Versionen für den Kunden durch Anwendung des SaaS-Modells vereinfacht werden. Die Verträge von salesforce.com haben in der Regel eine Mindestlaufzeit von einem Jahr und richten sich preislich nach der Anzahl der Anwender, die auf das System Zugriff haben sollen.

Unternehmensentwicklung

Die Firma salesforce.com wurde 1999 von Marc Benioff, einem ehemaligen Vorstandsmitglied von Oracle, gegründet. Bei der Produktentwicklung standen ein niedriger Preis, eine einfache Anwendung und Installation und vor allem die Möglichkeit unternehmensindividueller Anpassungen sowie die Kompatibilität mit anderen Systemen im Vordergrund. Bereits zwei Jahre nach seiner Gründung wurde das Unternehmen als der am schnellsten wachsende CRM-Anbieter auf dem Markt bezeichnet. Für seine innovativen Produkte hat salesforce.com außerdem eine Reihe von Auszeichnungen erhalten. Inzwischen betreut das Unternehmen ca. 30.000 Kunden, beschäftigt rund 2.000 Mitarbeiter und generierte 2006 einen Umsatz von fast 500 Millionen US-Dollar.

Quellen: www.salesforce.com, www.marketwatch.com

1.4 Erlösquellen für Softwareanbieter

Aus der zuvor dargestellten Unterscheidung zwischen Softwareanbietern im engeren sowie im weiteren Sinne lassen sich auch die Erlösquellen dieser Unternehmen ableiten. Demnach kann ein Softwareanbieter Lizenz- und/oder Dienstleistungserlöse erzielen.

1.4.1 Lizenzmodelle und -erlöse

Im Folgenden betrachten wir klassisch-kommerzielle Lizenzmodelle; auf Lizenzen für Open Source Software gehen wir dagegen in Abschnitt 4.2 ein. Bei der Lizenzpolitik ist grundsätzlich eine Vielzahl von unterschiedlichen Alternativen denkbar, die zudem miteinander kombiniert werden können. Dies führt dazu, dass Preislisten von Softwareanbietern häufig sehr komplex und wenig transparent sind. Wir werden auf die Preisstrategien für Softwareanbieter in Abschnitt 3.3 zurückkommen.

Zunächst werden einige grundlegende Lizenzmodelle vorgestellt. Insbesondere wollen wir darauf eingehen, welche dieser Modelle von den Softwareanbietern bzw. Anwendern präferiert werden. Zu diesem Themenfeld hat die Software & Information Industry Association eine Umfrage unter 698 Experten durchgeführt. Befragt wurden 487 Softwareanbieter und 211 Anwender (SIIA et al. 2006). Abbildung 1-2 zeigt einen Überblick über verschiedene Lizenzmodelle und inwieweit Softwareanbieter bzw. Anwender diese Modelle jeweils bevorzugen.

Die Ergebnisse sind zunächst nur wenig überraschend. Die Anwender favorisieren beispielsweise eher die Abrechnung nach Anzahl der gleichzeitig ein Softwarepaket nutzenden Anwender (concurrent user), während die Softwareanbieter die personifizierte Lizenzvergabe (named user) oder nach Arbeitsplätzen (seat) vorziehen. Die Motive sind klar: Die Anwender erhoffen sich von dem erstgenannten Modell geringere Kosten bzw. eine höhere Flexibilität; die Anbieter erwarten von der zweiten Alternative demgegenüber höhere Erlöse.

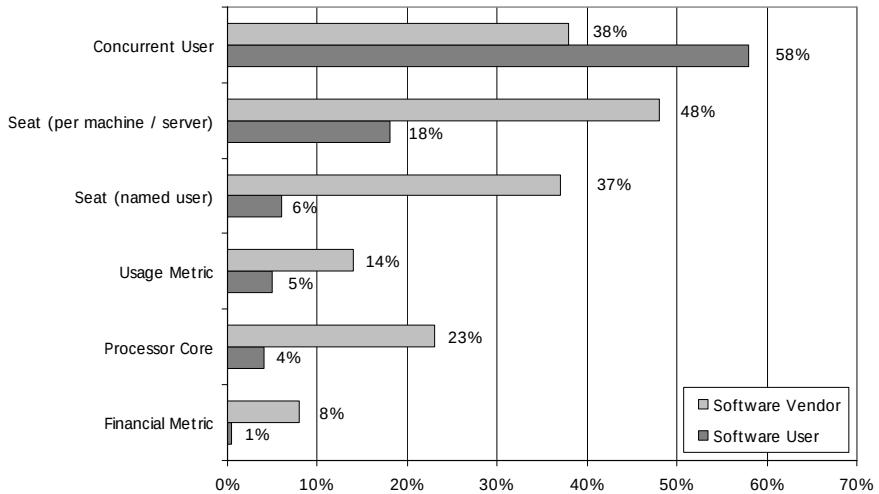


Abbildung 1-2: Von Softwareanbietern bzw. Anwendern präferierte Lizenzmodelle (SIIA et al. 2006)

Wir wollen im Weiteren die nutzungsabhängigen Lizenzmodelle (usage metric) näher betrachten, da ein zentrales Ergebnis der Umfrage ist, dass die Softwareanbieter zukünftig eine stärkere Verbreitung dieses Modells erwarten, wie die folgende Abbildung zeigt.

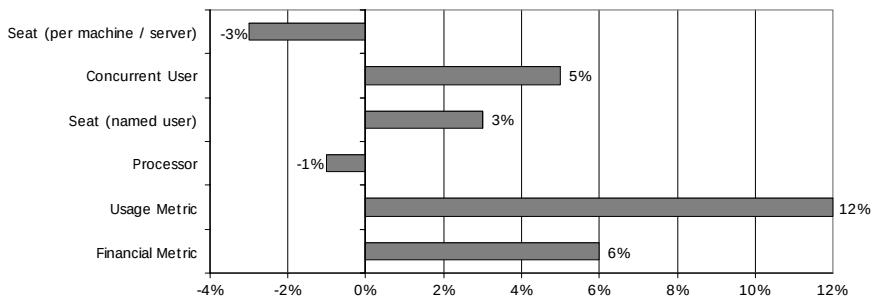


Abbildung 1-3: Erwartete Veränderung der Nutzung von Lizenzmodellen bis 2008 (SIIA et al. 2006)

Gehen wir zunächst kurz darauf ein, was unter einem nutzungsabhängigen Lizenzmodell verstanden wird. Wie der Name es bereits andeutet, werden die Kosten für den Nutzer bei Anwendung des Modells in Abhängigkeit von der Inanspruchnahme der Software ermittelt. Es handelt sich somit um eine Form der wertorientierten Bepreisung, wie sie häufig für digitale Güter vorgeschlagen wird (Shapiro u. Varian 1998). Das bedeutet, dass sich der Preis nicht an den Herstellungskosten, sondern am Wert für den Kunden orientiert. Der Hintergrund dieser Empfehlung ist die Kostenstruktur für digitale Güter, auf die wir in Abschnitt 2.1 dieses Buches eingehen.

Als Grundlage für die Anwendung dieses Lizenzmodells kann eine Vielzahl von Parametern herangezogen werden. Hierzu gehören etwa die Anzahl der mit einer Branchenlösung für Banken verwalteten Konten, die mit einer CRM-Software durchgeführten Kampagnen oder die auf Basis einer Supply-Chain-Management (SCM)-Lösung geplanten Routen.

Jedoch lehnen viele Anwender solche nutzungsabhängigen Lizenzmodelle zunehmend ab. Abbildung 1-4 zeigt, dass die Anwender dieses Modell deutlich negativer bewerten als noch im Jahr zuvor.

Damit stellt sich die Frage, warum die Vorbehalte der Anwender in Bezug auf nutzungsabhängige Lizenzmodelle gewachsen sind. Zum einen ist dahinter eine gewisse – vielleicht auch gesunde – Skepsis gegenüber Veränderungen zu vermuten. Warum sollte ein Softwareanbieter auch seine Lizenzmodelle ändern, wenn nicht, um die eigenen Umsätze zu erhöhen? Zum anderen sind die meisten dieser Modelle so aufgebaut, dass der Anwender, ähnlich wie bei bestimmten Handy-Verträgen, vorab ein Nutzungskontingent einkauft. So müssen die Nutzer beispielsweise vorab planen und festlegen, wie viele Kampagnen mit einer CRM-Software durchzuführen sind oder wie viele Rechnungen mit einem Softwaremodul erstellt werden sollen. Viele Anwender sehen dies als Problem an. Hinzu kommt, dass eine Nachzahlung fällig ist, wenn die tatsächliche Nutzung das zuvor geplante Kontingent übersteigt. Wird das Kontingent hingegen nicht vollständig ausgeschöpft, so erhalten die Anwender in der Regel keine Rückzahlungen oder Gutschriften.