

PETER BUXMANN
HEINER DIEFENBACH
THOMAS HESS

Die Softwareindustrie

Ökonomische Prinzipien,
Strategien, Perspektiven

Zweite Auflage



Springer

Die Softwareindustrie

2. Auflage

Peter Buxmann · Heiner Diefenbach · Thomas Hess

Die Softwareindustrie

Ökonomische Prinzipien, Strategien,
Perspektiven

2. Auflage

 Springer

Prof. Dr. Peter Buxmann
TU Darmstadt
Fachgebiet Wirtschaftsinformatik
Hochschulstr. 1
64289 Darmstadt
Deutschland
buxmann@is.tu-darmstadt.de

Dr. Heiner Diefenbach
TDS AG
Konrad-Zuse-Str. 16
74172 Neckarsulm
Deutschland
heiner.diefenbach@tds.fujitsu.com

Prof. Dr. Thomas Hess
Ludwig-Maximilians-Universität München
Institut für Wirtschaftsinformatik
und Neue Medien
Ludwigstr. 28
80539 München
Deutschland
thess@bwl.uni-muenchen.de

ISBN 978-3-642-13360-2

e-ISBN 978-3-642-13361-9

DOI 10.1007/978-3-642-13361-9

Springer Heidelberg Dordrecht London New York

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

© Springer-Verlag Berlin Heidelberg 2008, 2011

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, 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.

Einbandentwurf: WMXDesign GmbH, Heidelberg

Gedruckt auf säurefreiem Papier

Springer ist Teil der Fachverlagsgruppe Springer Science+Business Media (www.springer.com)

Vorwort zur zweiten Auflage

Seit dem Erscheinen der ersten Auflage unseres Buches sind mittlerweile knapp drei Jahre vergangen. In dieser Zeit hat sich in der Softwareindustrie einiges verändert. Und auch die verstärkte wissenschaftliche Auseinandersetzung mit der Softwareindustrie hat neue Erkenntnisse hervorgebracht. Mit dieser zweiten Auflage versuchen wir, diesen Entwicklungen Rechnung zu tragen. So haben wir dieses Buch beispielsweise um die Themengebiete Software as a Service sowie Plattformkonzepte grundlegend erweitert. Darüber hinaus enthält dieses Buch neue Teilabschnitte, u. a. zum Outsourcing -Verhalten von Softwareanbietern und -anwendern, zur Bereitstellung betriebswirtschaftlicher Software als Open Source Software, zur Softwareauswahl sowie zu den Wertschöpfungsketten in der Softwareindustrie. Neu ist ebenfalls die noch klarere Aufteilung des Buchs in einen allgemeinen Grundlagenteil und einen Teil mit speziellen Themenstellungen.

Wir bedanken uns sehr herzlich bei unseren Mitarbeiterinnen und Mitarbeitern der Software Economics Group Darmstadt-München (www.software-economics.org) für ihre Unterstützung. Insbesondere haben Dr. Alexander Benlian, Dr. Björn Brandt, Christoph Burkard, Tobias Draisbach, Jin Gerlach, Thomas Göрге, Daniel Hilbert, Christian Hörndlein, Sonja Lehmann, Janina Matz und Dr. Christian Wolf neue Inhalte beigesteuert, die zum Teil auch Gegenstand der jeweiligen Dissertationen sind bzw. waren.

Bei Anette von Ahsen bedanken wir uns erneut sehr herzlich für ihre Unterstützung bei der Erstellung dieser zweiten Auflage. Sie hat uns nicht nur viele „Bug Reports“, sondern darüber hinaus noch die passenden „Patches“ geliefert. Jasmin Kaiser hat uns hervorragend bei der Manuskripterstellung unterstützt. Ebenso dankbar sind wir Dr. Niels Thomas vom Springerverlag für die erneut sehr angenehme Zusammenarbeit.

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

Darmstadt, Deutschland
Neckarsulm, Deutschland
München, Deutschland
Oktober 2010

Peter Buxmann
Heiner Diefenbach
Thomas Hess

Inhaltsverzeichnis

Teil I Allgemeine Grundlagen

1 Die Spielregeln in der Softwareindustrie	3
1.1 Software und Softwaremärkte: Ausgewählte Besonderheiten der Softwareindustrie im Überblick	3
1.2 Die Anfänge der Softwareindustrie	4
1.3 Typen von Softwareanbietern und Auswahlentscheidungen der Anwender	5
1.3.1 Softwareanbieter im engeren und weiteren Sinne	5
1.3.2 Die Auswahl von Software	12
1.4 Geschäftsmodelle in der Softwareindustrie	17
1.5 Dienstleistungserlöse in der Softwareindustrie	20
2 Ökonomische Prinzipien der Softwareindustrie	23
2.1 Eigenschaften digitaler Güter	23
2.2 Netzeffekte auf Softwaremärkten: The Winner takes it all	24
2.2.1 Netzeffekte: Grundlagen und Definitionen	25
2.2.2 Auswirkungen von Netzeffekten auf Softwaremärkte	27
2.2.3 Struktur von Softwaremärkten	31
2.2.4 Netzeffekte als Wettbewerbsfaktor	33
2.2.5 Ein Anwendungsbeispiel: Zweiseitige Netzeffekte und Plattformstrategien in der digitalen Spieleindustrie	34
2.2.6 Grenzen der Netzeffekttheorie	38
2.3 Das Standardisierungsproblem	40
2.3.1 Ansatz und Hintergründe	40
2.3.2 Das zentrale Standardisierungsproblem als Optimierungsproblem	43
2.3.3 Das dezentrale Standardisierungsproblem – Eine spieltheoretische Darstellung	45
2.3.4 Das Standardisierungsproblem – Lessons learned	48

- 2.4 Transaktionskostentheorie: Auf der Suche nach den Grenzen eines Softwareunternehmens 49
 - 2.4.1 Ansatzpunkt und Elemente der Transaktionskostentheorie 50
 - 2.4.2 Arbeitsteilung zwischen Unternehmen aus Sicht der Transaktionskostentheorie 52
 - 2.4.3 Strukturelle Veränderungen der Transaktionskosten: The Move to the Middle 54
 - 2.4.4 Ausblick: Intermediäre und Transaktionskosten 56
- 2.5 Softwareentwicklung als Agency-Problem: Anreizkompatible Entlohnung und effiziente Kontrolle 57
 - 2.5.1 Principal-Agent-Beziehungen: Definitionen und Grundlagen 57
 - 2.5.2 Anreizkompatible Vergütungsschemata 59
 - 2.5.3 Kontrollsysteme 62
- 3 Strategien für Softwareanbieter 65**
 - 3.1 Kooperations- und Übernahmestrategien 65
 - 3.1.1 Kooperationen in der Softwareindustrie 65
 - 3.1.2 Mergers & Acquisitions in der Softwareindustrie 78
 - 3.2 Vertriebsstrategien 87
 - 3.2.1 Gestaltung des Vertriebssystems: Organisation und Vertriebswege in der Softwareindustrie 87
 - 3.2.2 Gestaltung der Beziehungen zu Vertriebspartnern und Key Accounts 92
 - 3.2.3 Kennzahlensysteme als Instrument für das Vertriebscontrolling in der Softwareindustrie 94
 - 3.2.4 Gestaltung der Verkaufsaktivitäten 98
 - 3.3 Preisstrategien 105
 - 3.3.1 Grundüberlegungen 105
 - 3.3.2 Parameter der Preisgestaltung für Softwareprodukte 106
 - 3.3.3 Preisstrategien von Softwareanbietern – empirische Ergebnisse 122
 - 3.3.4 Ansätze zur Preissetzung für Individualsoftwareanbieter 126
 - 3.4 Entwicklungsstrategien 128
 - 3.4.1 Strukturierung der Softwareentwicklung 128
 - 3.4.2 Softwareunterstützung für die Softwareentwicklung 133
 - 3.4.3 Personalführung in der Softwareentwicklung 136

Teil II Spezielle Themen

4 Outsourcing und Offshoring der Softwareentwicklung	141
4.1 Überblick	141
4.2 Formen des Outsourcings und Offshorings	142
4.3 Motive für Outsourcing und Offshoring	145
4.3.1 Kosteneinsparungen	146
4.3.2 Erhöhung der Flexibilität	147
4.3.3 Konzentration auf Kernkompetenzen	147
4.3.4 Know-how-Zukauf	148
4.3.5 Ausnutzung des „Follow-the-Sun“-Prinzips	148
4.4 Standortwahl von Softwareanbietern	149
4.5 Outsourcing durch Softwareanwender	152
4.5.1 Outsourcing der Neuentwicklung von Individualsoftware	152
4.5.2 Outsourcing der Anpassung von Standardsoftware	155
4.5.3 Outsourcing der Weiterentwicklung und Wartung von Anwendungssoftware	158
4.5.4 Zufriedenheit der Anwender mit Onshore-, Nearshore- und Farshoreanbietern	162
4.6 Nearshoring versus Farshoring: Die Entfernung zum Kunden als Erfolgsfaktor?	164
4.6.1 Sprachliche und kulturelle Barrieren in Offshore-Projekten	164
4.6.2 Die Bedeutung persönlicher Treffen für den Projekterfolg	166
4.6.3 Herausforderungen und Chancen der Zeitverschiebung	168
4.7 Outsourcing durch Softwareanbieter	170
4.7.1 Status quo der Spezialisierung und Arbeitsteilung: Einsichten aus drei Fällen	170
4.7.2 Zur zukünftigen Arbeitsteilung in der Softwareindustrie	179
5 Plattformkonzepte	189
5.1 Überblick	189
5.2 Produktplattformen in der Softwareindustrie	190
5.2.1 Kostenstruktur plattformbasierter Softwareentwicklung	190
5.2.2 Organisatorische Unterstützung der Etablierung von Produktplattformen	193
5.2.3 Add-on: Industrialisierung als Managementkonzept für die Softwareindustrie	195

5.3	Branchenplattformen in der Softwareindustrie	197
5.3.1	Offenheit einer Branchenplattform	197
5.3.2	Das Management der Komplementäre	201
6	Software as a Service: die Anwendungsebene des Cloud Computing	205
6.1	Überblick	205
6.2	Einsatzgebiete und Beispiele	206
6.3	SaaS aus Anwendersicht – Chancen und Risiken	209
6.3.1	Grundüberlegungen	209
6.3.2	Empirische Untersuchung zu Chancen und Risiken der SaaS-Anwender	213
6.4	SaaS aus Anbietersicht – Preisstrategien und Geschäftsmodelle	217
6.4.1	Grundüberlegungen	217
6.4.2	Empirische Untersuchung der Preisstrategien und Geschäftsmodelle für SaaS-Anbieter	218
6.4.3	Fallstudie zum Vergleich nutzungsabhängiger und nutzungsunabhängiger Preismodelle	222
7	Open Source Software	227
7.1	Überblick	227
7.2	Charakteristika von Open Source Software	227
7.3	Open-Source-Projekte: Prinzipien und Motivation der Softwareentwickler	232
7.3.1	Ablauf und Organisation von Open-Source- Projekten	233
7.3.2	Zur Motivation der Beitragenden	234
7.4	Open Source Software aus Sicht des Anwenders	236
7.5	Engagement kommerzieller Softwareanbieter	237
7.6	Quelloffene ERP-Systeme	239
	Literaturverzeichnis	249
	Sachverzeichnis	261

Abkürzungsverzeichnis

AGB	Allgemeine Geschäftsbedingungen
ASD	Adaptive Software Development
ASP	Application Service Providing
BOT	Build Operate Transfer
BPEL	Business Process Execution Language
BSA	Business Software Alliance
BSD	Berkeley Software Distribution
CAR	Cumulative Abnormal Return
CAAR	Cumulative Average Abnormal Return
CASE	Computer Aided Software Engineering
CCC	Content Communication and Collaboration
CEO	Chief Executive Officer
CIM	Computational Independent Model
CIO	Chief Information Officer
CMMI	Capability Maturity Model Integration
CORBA	Common Object Request Broker Architecture
CRM	Customer Relationship Management
CTO	Chief Technology Officer
CVS	Concurrent Versions System
DACH	Deutschland, Österreich, Schweiz
DCOM	Distributed Component Object Model
EDI	Electronic Document Interchange
EPK	Ereignisgesteuerte Prozesskette
EDV	Elektronische Datenverarbeitung
EPK	Elektronische Prozesskette
ERP	Enterprise Resource Planning
GE	Geldeinheiten
GNU	GNU is not Unix
GPL	General Public License
HTTP	Hypertext Transfer Protocol
IE	Internet Explorer
IP	Intellectual Property

ISO	International Organization for Standardization
ITT	Indian Institute of Technology
KMU	Kleine und mittelständische Unternehmen
KPI	Key Performance Indicator
LGPL	Library/Lesser General Public License
LoC	Lines of Code
M&A	Mergers & Acquisitions
MDE	Model Driven Engineering
MIT	Massachusetts Institute of Technology
MPL	Mozilla Public License
ODT	Open Document Format
OEM	Original Equipment Manufacturer
OSI	Open Source Initiative
OSS	Open Source Software
PDF	Portable Document Format
PIM	Platform Independent Model
PM	Platform Model
PMC	Point of Marginal Cheapness
PME	Point of Marginal Expensiveness
PPS	Produktionsplanungs- und -steuerungssystem
PSM	Platform Specific Model
PSM	Price Sensitivity Meter
ROI	Return on Investments
RPC	Remote Procedure Call
SaaS	Software as a Service
SIIA	Software & Information Industry Association
SCM	Supply Chain Management
SOA	Serviceorientierte Architektur
SQL	Structured Query Language
SW	Software
TIME	Telekommunikation/Information/Medien/Entertainment
UDDI	Universal Description, Discovery, and Integration
URI	Uniform Resource Identifier
US-GAAP	United States Generally Accepted Accounting Principles
VB	Vertriebsbeauftragte(r)
VHS	Video Home System
WSDL	Web Services Description Language
W3C	World Wide Web Consortium
XP	eXtreme Programming
XML	Extensible Markup Language
ZB	Zahlungsbereitschaft

Teil I
Allgemeine Grundlagen

Kapitel 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 erzielen 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 81 bzw. 89% (Lünen-donk 2009, S. 388). Zudem führt der Netzeffektcharakter von Software 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 anzubieten 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 [Abschn. 3.1.2](#)). Wir werden auf die Besonderheiten dieser Märkte zurückkommen, wollen uns aber zunächst mit den Akteuren der Softwareindustrie beschäftigen.

1.3 Typen von Softwareanbietern und Auswahlentscheidungen der Anwender

1.3.1 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 Merkmalen 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 2009, S. 236 f.). Auf eine ökonomische Untersuchung von Dienst- und Werkverträgen gehen wir in [Abschn. 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 [Abschn. 4.6](#) 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 führender Anbieter von Unternehmenssoftware und drittgrößter unabhängiger Softwarelieferant der Welt. Weltweit beschäftigt die SAP 47.578 Mitarbeiter in 50 Niederlassungen. Die SAP-Lösungen sind bei 89.000 Kunden im Einsatz.

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 inkonsistenter 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 100% 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 individuellen Anwendungslösung zusammenzuführen (siehe hierzu [Abschn. 4.7.2](#)).

Zukünftig wird der Anteil von Standardsoftwarelösungen im Portfolio der Unternehmen tendenziell zunehmen, wie eine von uns durchgeführte Umfrage unter

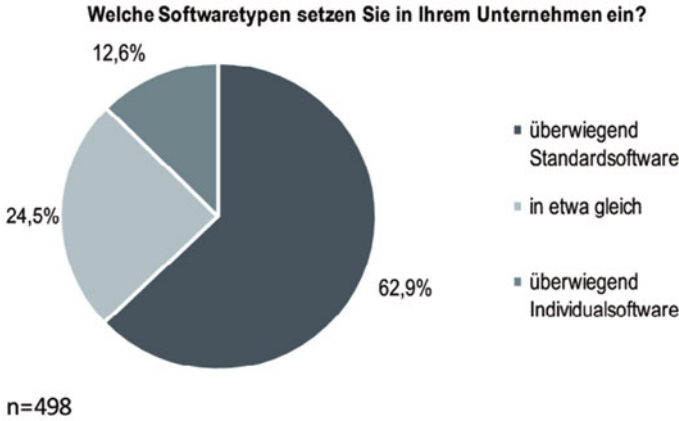


Abb. 1.1 Anteile Softwaretypen

498 deutschen CIOs zeigt (Buxmann et al. 2010). So nutzen 62,9% der befragten Unternehmen überwiegend Standardsoftwarelösungen, in 24,5% der Fälle setzen Unternehmen Standard- und Individualsoftware in etwa zu gleichen Teilen ein und bei 12,6% der antwortenden Unternehmen wird überwiegend Individualsoftware eingesetzt (siehe Abb. 1.1).

Neben der Erhebung des Status quo zur Nutzung von Standard- und Individualsoftware wurden die Unternehmen auch gefragt, wie sie die zukünftige Entwicklung bezüglich des Einsatzes dieser Softwaretypen einschätzen. 66,7% der Befragten stimmten der Aussage zu, dass in Unternehmen zukünftig tendenziell mehr Standardsoftware zu Lasten von Individualsoftware eingesetzt werden wird. 20,2% der Unternehmen sprachen sich gegen diese Aussage aus. 13,1% äußerten sich indifferent (siehe Abb. 1.2). 34,0% der Teilnehmer stimmten in diesem Zusammenhang

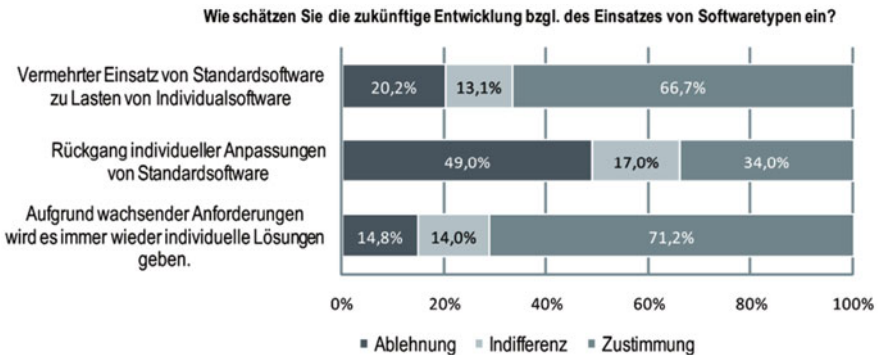


Abb. 1.2 Einschätzung der zukünftigen Entwicklung für die Anteile der Softwaretypen

der Aussage zu, dass zukünftig individuelle Anpassungen von Standardsoftware zurückgehen werden, knapp die Hälfte (49,0%) lehnte diese These dagegen ab. Der weitaus größte Teil (71,2%) der antwortenden Unternehmen ist der Auffassung, dass es aufgrund wachsender Anforderungen der Anwender immer wieder individuelle Lösungen für spezielle Problemstellungen geben wird, die in die gesamte IT-Landschaft zu integrieren sind.

Zukünftig werden Unternehmen somit mehr Standardsoftwarelösungen einsetzen; allerdings werden weiterhin individuelle Anpassungen notwendig sein. Für Spezialprobleme, für die es (noch) keine standardisierte Anwendungssoftware auf dem Markt gibt, wird man nach Maßgabe von Wirtschaftlichkeitsüberlegungen tendenziell nach wie vor Individualsoftware entwickeln oder entwickeln lassen, die in die IT-Landschaft der Unternehmen einzubinden ist.

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 1.3 fasst unsere Klassifikation von Softwareanbietern zusammen.

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 2009)

- IT-Beratungs- und Systemintegrations-Unternehmen (mit Angeboten im Bereich IT-Beratung, Entwicklung von Individualsoftware etc.),
- IT-Serviceunternehmen (mit Leistungen wie Outsourcing, ASP, Schulung etc.),
- Business Innovation/Transformation Partner (Management- und IT-Beratung sowie System-Realisierung).

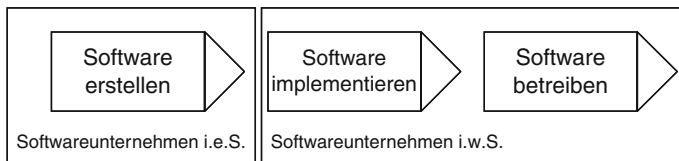


Abb. 1.3 Klassifikation von Softwareunternehmen

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. 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. lediglich 32% der betrachteten IT-Projekte innerhalb der Zeit- und Budgetvorgaben erfolgreich abgeschlossen (The Standish Group International 2009, S. 2). 44% überschritten diese Planwerte erheblich und 24% wurden überhaupt nicht zu Ende geführt. Bei den als komplex eingestuften Projekten wurden Durchlaufzeit und Budget um mehr als das Doppelte überschritten.

Der Betrieb von IT-Lösungen gehört seit Jahren zum Outsourcing-Geschäft (siehe hierzu auch Kap. 4). Hier gibt es eine Vielzahl von Playern in einem weltweit umkämpften Markt, die wir auch zu den Softwareanbietern i. w. S. zählen.

In diesem Themenbereich ist in der letzten Zeit mit Cloud-Computing eine neue Form des IT-Bezugs entstanden. Wie bei solchen neuen Entwicklungen häufig zu beobachten, überbieten sich Experten, Anbieter und Marktforschungsgesellschaften mit einer Vielzahl unterschiedlicher Definitionen und Marktprognosen mit exponentiell ansteigenden Wachstumskurven. Wir wollen hier auf die Definition des National Institute of Standards and Technology zurückgreifen. Demnach handelt es sich bei Cloud Computing um ein Modell, „das einen komfortablen, bedarfsabhängigen und netzbasierten Zugriff auf eine gemeinsam benutzte Menge konfigurierbarer Rechenressourcen ermöglicht, die schnell, mit geringem Verwaltungsaufwand und ohne (menschliche) Interaktion mit einem Anbieter bereitgestellt und wieder freigegeben werden können“ (National Institute of Standards and Technology 2009).

Dabei basiert Cloud Computing auf dem Prinzip der Virtualisierung und Verteilung der IT-basierten Serviceleistungen auf weltweit verteilte Rechner. Aus ökonomischer Perspektive haben die Anbieter den Vorteil, dass sie verfügbare Ressourcen besser ausnutzen und damit angebotsseitige Skaleneffekte realisieren können. Daher ist es auch nicht verwunderlich, dass insbesondere große Anbieter sich diesen Markt erschließen wollen, wie beispielsweise Amazon, Google und Microsoft. Dabei wird das Spektrum von Cloud-Angeboten häufig unterschieden, wie in Abb. 1.4 dargestellt.

Software as a Service (SaaS) bezeichnet Softwarelösungen auf der Anwendungsebene, die grundsätzlich geeignet sind, in einer Cloud angeboten zu werden (siehe hierzu auch Kap. 6). Cloud-Anbieter offerieren darüber hinaus auf der Plattformebene (Platform as a Service) beispielsweise den Betrieb von Datenbanken oder auch Middleware-Lösungen. Wie klassische Outsourcing-Provider auch,

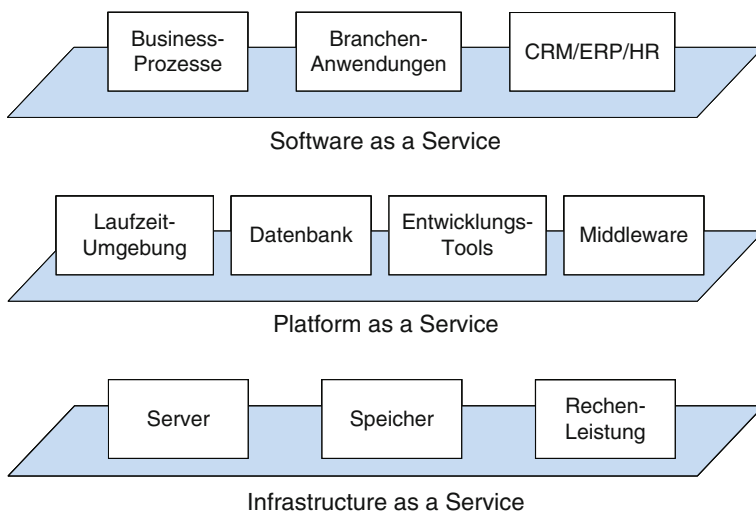


Abb. 1.4 Angebotspektrum Cloud Computing (Vaquero et al. 2009)

bieten Cloud-Anbieter Hardware- und Infrastrukturdienste, wie etwa Server- und Speicherkapazitäten sowie Rechenleistungen, an (Infrastructure as a Service).

Dabei muss man sich im Klaren sein, dass es sich bei Cloud Computing um nichts revolutionär Neues handelt. Die Anbieter offerieren ihren Kunden Outsourcing-Angebote, wie das auch viele andere Hersteller seit Jahren tun. Auch das Prinzip der Virtualisierung ist nicht wirklich neu. Interessant kann das Konzept jedoch werden, wenn es den Cloud-Computing-Anbietern gelingt, ihren Kunden Lösungen anzubieten, deren Kosten signifikant unter denen von klassischen Outsourcing-Anbietern liegen. Vor diesem Hintergrund ist unter diesen Softwareanbietern i. w. S. ein Wettbewerb zu erwarten, der zukünftig zu deutlich sinkenden Preisen beim Betrieb von IT-Lösungen führen wird.

1.3.2 Die Auswahl von Software

Grundlage der Entwicklung und Gestaltung von Geschäftsmodellen für Softwareanbieter im engeren sowie im weiteren Sinne sind die Auswahlentscheidungen der Kunden. Die folgenden Abschnitte sollen aufzeigen, welche Prozesse und Entscheidungsfaktoren bei der Auswahl und Bewertung alternativer Softwarelösungen in Unternehmen (Abschn. 1.3.2.1) und Haushalten (Abschn. 1.3.2.2) existieren.

1.3.2.1 Softwareauswahl in Unternehmen

In der Regel durchlaufen Unternehmen mehrere sequenzielle Phasen bis zur endgültigen Auswahlentscheidung (siehe Abb. 1.5). Hierbei wird der Lösungsraum der zur Verfügung stehenden Alternativen schrittweise eingegrenzt. Typische Kontextvariablen sind Unternehmensaspekte, wie z. B. Branchenzugehörigkeit, Größe oder zu unterstützende Funktionen, bzw. Umfeldvariablen, wie etwa Technologiestandards. Aus den Rahmenbedingungen heraus werden Ziele definiert, die mit der Anschaffung der Software verfolgt werden sollen. Der nächste Schritt stellt in der Regel die Definition von Kriterien dar. Kriterien sind relevante Merkmale für die Beschreibung eines Softwaresystems und die Grundlage für eine Bewertung.

Mit Hilfe von Marktübersichten werden potenzielle Alternativen ermittelt. Typisch ist das sich anschließende zweistufige Bewertungs- und Auswahlverfahren. Bei der Grobauswahl werden die Alternativen anhand so genannter K.O.-Kriterien bewertet. Ihre Erfüllung ist notwendige Voraussetzung für eine detailliertere Untersuchung. Für die verbliebenen Alternativen werden auf Grundlage detaillierter Anforderungsdefinitionen (Pflichtenhefte), die aus den Kriterien abgeleitet werden,

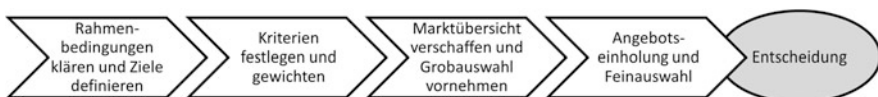


Abb. 1.5 Softwareauswahl in Unternehmen

entsprechende Angebote eingeholt bzw. Informationen angefordert. Basierend auf den auf diese Weise gewonnenen Informationen wird eine Feinauswahl vorgenommen, die in einen Entscheidungsvorschlag mündet. Einige Ansätze berücksichtigen ferner die an die Entscheidung anschließenden Phasen, wie Vertragsverhandlungen und funktionale Testläufe zur Bestimmung des Anpassungsaufwandes.

Im Folgenden soll detaillierter auf die Entwicklung eines Zielsystems, die Kriterienableitung und -gewichtung sowie auf die während der Grob- und Feinauswahl relevanten Aktivitäten eingegangen werden.

Die Formulierung der Ziele, die mit der Einführung bzw. dem Einsatz einer Software in einem bestimmten Bereich erreicht werden sollen, stellt in der Regel den Ausgangspunkt der Analyse und den Maßstab zur Beurteilung der Alternativen dar. Bei der Ableitung der Ziele können unterschiedliche Verfahren zum Einsatz kommen: Zum einen werden Ziele top-down aus den obersten Unternehmenszielen bzw. aus der IT-Strategie abgeleitet. Das kann in der Weise geschehen, dass oberste Unternehmensziele, wie beispielsweise die Gewinnsteigerung, auf das konkrete Entscheidungsproblem „Auswahl einer Standardsoftware für . . .“ übertragen werden. Eine Variante dieser Zielableitung stellt eine Konkretisierung der obersten Ziele durch Unterziele im Sinne einer Zweck-Mittel-Beziehung bezogen auf das Entscheidungsproblem dar. Beispiele hierfür sind die Senkung der Durchlaufzeit, die Senkung von Lagerbeständen oder die Verkürzung von Reaktionszeiten.

Ein anderer Ansatz geht entgegengesetzt zur oben vorgestellten Strategie bottom-up vor. Diese Ziele haben einen stark inhaltlich-funktionalen Fokus mit Bezug zum konkreten Entscheidungsproblem bzw. zu existierenden Strukturen (wie z. B. die bestehende IT-Architektur). Beispiele sind Ziele in Form allgemeiner Softwaremerkmale, wie Datenbank- oder Hardwareherstellerunabhängigkeit, Kompatibilität mit bestehenden Versionen oder spezifische anwendungsbezogene Funktionalitäten, wie beispielsweise eine automatische Teilenummernvergabe mit Prüfziffer oder die Verbreiterung der Informationsbasis. Diese Aspekte stellen oftmals eher bereits Anforderungen dar und nicht Ziele im engeren Sinn, die um ihrer selbst willen verfolgt werden. In der Praxis wird deshalb meist ein kombiniertes Verfahren aus top-down und bottom-up-Vorgehensweise zur Zielbestimmung als Voraussetzung der Kriterienableitung angewendet. Allerdings wird auf das erforderliche Gegenstromprinzip im Sinne eines Abgleichens und Anpassens oftmals zugunsten einer vereinfachten Aufzählung verzichtet.

Werden Kriterien aus dem Zielsystem abgeleitet, so stellen sie in der Regel die unterste Ebene eines solchen Systems dar. Die Auswirkung einer bestimmten Systemeigenschaft auf die angestrebten Ziele ist damit transparent. Im Sinne der Zielhierarchie werden die Auswahlkriterien häufig in software-, implementierungs- und anbieterbezogene Merkmale untergliedert. Neben den eigentlichen Leistungsmerkmalen der Software spielen bei komplexeren Systemen insbesondere Implementierungskriterien eine wesentliche Rolle, da sie neben den Anschaffungskosten erhebliche Auswirkungen auf die Gesamtaufwendungen für die Software haben können. Die Implementierung und Einführung komplexerer Software-Systeme werden oftmals von Softwareanbietern bzw. von Implementierungspartnern begleitet, deren Expertise und Professionalität im Vorfeld der Softwareauswahl in der Regel

Tabelle 1.1 Typische Auswahlkriterien bei der Auswahl von ERP-Systemen

Auswahlkriterien (Software)	Auswahlkriterien (Implementierung/Anbieter)
Funktionalität	Integrationsfähigkeit in bestehende IT-Architektur
Kosten	Implementierungszeit/Anpassungsaufwand
Benutzerfreundlichkeit	Support durch Anbieter
Zuverlässigkeit	Reputation des Anbieters

auch zu evaluieren ist. Neben rein rationalen Kriterien spielen in der Praxis auch häufig politische Kriterien (z. B. Absprachen, Klüngerlei, Machtspiele) eine nicht zu unterschätzende Rolle, die den Auswahlprozess erheblich beeinflussen können (Howcroft und Light 2006).

Tabelle 1.1 zeigt für das Beispiel der Auswahl von ERP-Systemen, welche konkreten Auswahlkriterien für die Dimensionen ‘Software’, ‘Implementierung’ und ‘Anbieter’ in der Praxis herangezogen werden (Keil und Tiwana 2006; Jadhav und Sonar 2009).

Hinsichtlich der Gewichtung der Einzelkriterien im Rahmen der Gesamtscheidung wird in der Regel davon ausgegangen, dass die Kriterien vorhanden bzw. erfüllt sein sollen. Andererseits ist auch denkbar, dass die Erfüllung eines Kriteriums in Abstufungen möglich ist. In einem solchen Fall ist die Angabe eines angestrebten Zielniveaus bzw. die Beurteilung von unterschiedlichen Zielniveaus erforderlich. So ist allein aus einem Kriterium „Datenbankunabhängigkeit“ nicht ersichtlich, ob hierzu die Unterstützung von zwei, drei oder zehn unterschiedlichen Datenbankmanagementsystemen erforderlich ist. Dies ist insbesondere zur Vermeidung von Überbewertungen von Bedeutung, wenn etwa bezüglich eines Ziels lediglich ein satisfizierendes Zielniveau angestrebt wird. Auf der anderen Seite liegen häufig auch Ziele vor, bei denen ein bestimmtes Anspruchsniveau mindestens erreicht werden muss, damit eine Alternative überhaupt in Betracht gezogen bzw. detaillierter untersucht wird. Solche K.O.- oder Killer-Kriterien tragen zwar einerseits dazu bei, den Auswahlprozess effizient zu gestalten. Andererseits bringen sie aber auch die Gefahr mit sich, dass Alternativen aufgrund einer auch nur geringen Unterschreitung des Mindestanspruchsniveaus eines Ziels ausgesondert werden.

Zur Gewichtung der Kriterien wird typischerweise ein holistischer Ansatz angewendet, wie er beispielsweise aus der Nutzwertanalyse bekannt ist. Dabei erfolgt die Vergabe der Gewichte durch einen direkten Vergleich der Kriterien. So wird z. B. in empirischen Studien zur Auswahl von ERP-Systemen berichtet, dass die Kriterien Funktionalität, Zuverlässigkeit und Kosten am meisten ins Gewicht fallen, während die Benutzerfreundlichkeit und der Implementierungsaufwand als weniger wichtig bewertet werden (Keil und Tiwana 2006). Bei diesem Ansatz erscheint die Vergabe sehr differenzierter Gewichtungen jedoch problematisch. Insbesondere bei komplexen Problemen, zu denen die Softwareauswahl u. a. aufgrund der

großen Anzahl von Bewertungskriterien gehört, besteht allenfalls die Möglichkeit der Rangfolgenbildung in ordinaler Skalierung. Exaktere Angaben unterstellen (Schein-) Genauigkeiten, die in den seltensten Fällen tatsächlich die Präferenz des Entscheiders wiedergeben.

Marktüberblick, Alternativenvorauswahl und -bewertung vollziehen sich in der Praxis häufig interdependent und miteinander vernetzt. Schon die Berücksichtigung bzw. die Nichtberücksichtigung eines Systems beim Marktüberblick stellt eine Auswahl dar. Dieser erste Auswahlsschritt ist aufgrund der großen Marktbreite geprägt durch heuristische Ansätze wie z. B.:

- Berücksichtigung von Systemen, die die Mitbewerber einsetzen,
- Berücksichtigung von Systemen, die gegenwärtig in der Fachpresse große Beachtung finden (z. B. Google Apps Premier, SAP ERP u. a.),
- Berücksichtigung von Systemen, mit denen Mitarbeiter bereits Erfahrungen haben,
- Berücksichtigung von Systemen, bei deren Auswahl sich ein Berater einen Anschlussauftrag im Rahmen der Implementierung erhofft,
- Berücksichtigung von Systemen aufgrund einer zufälligen Auswahl, z. B. durch Messebesuche.

Wie bereits dargestellt, bilden Bewertung und Auswahl einen gekoppelten Prozess, wobei die Anzahl der Alternativen mittels K.O.-Kriterien verringert werden kann. Dabei werden typischerweise die folgenden Verfahren angewendet:

- Auswertung von Systembeschreibungen,
- Auswertung bearbeiteter Ausschreibungsunterlagen bzw. Anforderungskataloge durch die potenziellen Anbieter sowie
- Präsentationen des Systems bzw. konkrete Funktionspräsentation auf Basis definierter Anwendungsszenarien.

Bei der Bewertung bestimmter Kriterien wird dabei häufig eine dreistufige Skala mit den Ausprägungen „erfüllt“, „teilweise erfüllt“, „nicht erfüllt“ zugrundegelegt. Schließlich besteht das Bedürfnis, die Ergebnisse der Einzelbewertung zu einer übergreifenden Bewertung zusammenzufassen.

Da die Eignung eines Systems nicht nur aus seiner funktionalen Übereinstimmung mit den gestellten Anforderungen resultiert, sondern auch von den Kosten des Systems abhängt, werden diese häufig in der Weise in die Bewertung einbezogen, dass ein Kosten-Nutzen-Koeffizient gebildet wird. Auf Basis des Vergleichs entsprechender quantitativer und qualitativer Bewertungen wird schließlich die Auswahlentscheidung getroffen.

1.3.2.2 Softwareauswahl durch Konsumenten

Während es sich bei Auswahlentscheidungen in Unternehmen in der Regel um Gruppenentscheidungen handelt, werden Konsumentenentscheidungen meistens

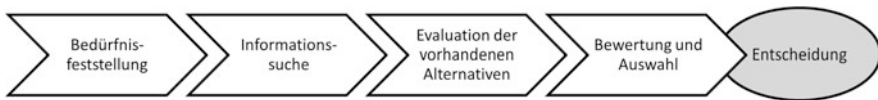


Abb. 1.6 Prozess zur Auswahl von Software durch Konsumenten

alleine getroffen. Ähnlich wie bei Auswahlentscheidungen im Unternehmenskontext durchläuft ein Konsument bei der Auswahl von Produkten im Allgemeinen und Software im Speziellen typischerweise vier Phasen (Blackwell et al. 2003), die in Abb. 1.6 illustriert werden.

Nachdem ein Konsument sich des Bedürfnisses bewusst wird, eine neue Software kaufen zu wollen (Phase 1: Bedürfnisfeststellung), sucht er anschließend nach geeigneten Informationen über alternative Produkte, die das Bedürfnis befriedigen sollen. Dabei identifiziert der Konsument zunächst – mehr oder weniger bewusst – wesentliche subjektive und objektive Kriterien für seine Entscheidungsfindung (Phase 2: Informationssuche). Nach der Informationssammlung verarbeitet der Konsument die vorhandenen Daten und evaluiert jede Alternative vor dem Hintergrund seiner persönlichen Zielkriterien (Phase 3: Evaluation der Alternativen). Schließlich vergleicht und bewertet der Konsument die vorhandenen Alternativen und führt daraufhin eine Auswahl- bzw. Kaufentscheidung durch (Phase 4: Bewertung und Auswahl).

Im Gegensatz zu Auswahlentscheidungen im Unternehmen spielen bei Auswahlentscheidungen von Konsumenten neben rein objektiven Kriterien (wie Kosten oder Funktionalität einer Software) insbesondere individuelle bzw. interne (wie z. B. die Persönlichkeit, Emotionen, Wahrnehmungen oder der Lebensstil des Konsumenten) und situative bzw. externe Faktoren (z. B. Kultur oder Mund-zu-Mund-Propaganda) eine Rolle. Um aus Anbietersicht besser verstehen zu können, welche wesentlichen Kriterien Konsumenten bei der Software-Auswahl heranziehen, sollen diese nachfolgend vertiefend beleuchtet werden.

Wie oben bereits angeführt, spielen bei Konsumentenentscheidungen neben rein rationalen Auswahlkriterien insbesondere weichere Faktoren eine Rolle. Ein umfassendes Modell, das diesen Besonderheiten speziell in digitalen Produktwelten Rechnung trägt, ist das Modell der Erfahrungswerte digitaler Güter nach Mathwick et al. 2001 (siehe Abb. 1.7). Es veranschaulicht, dass digitale Produkte (wie z. B. Spielesoftware) für Konsumenten einen Erfahrungswert besitzen, der vor dem eigentlichen Kauf (bewusst oder unbewusst) evaluiert wird. Der Erfahrungswert setzt sich dabei aus utilitaristischen und hedonischen Merkmalen zusammen.

Utilitaristische Merkmale sind überwiegend objektiver Natur und entsprechen typischerweise Kosten-Nutzen-Kalkülen. In die Kosten-Nutzen-Betrachtung fließen wiederum weitgehend rationale Bewertungskriterien, wie z. B. die Funktionalität, Kompatibilität oder die Anschaffungskosten der Software, ein. Der mit der Software einhergehende Service beschreibt auf der anderen Seite, welche Unterstützungs- und Garantieleistungen (z. B. Hotline oder Gewährleistungsansprüche) der Nutzer beim Erwerb der Software erhält.

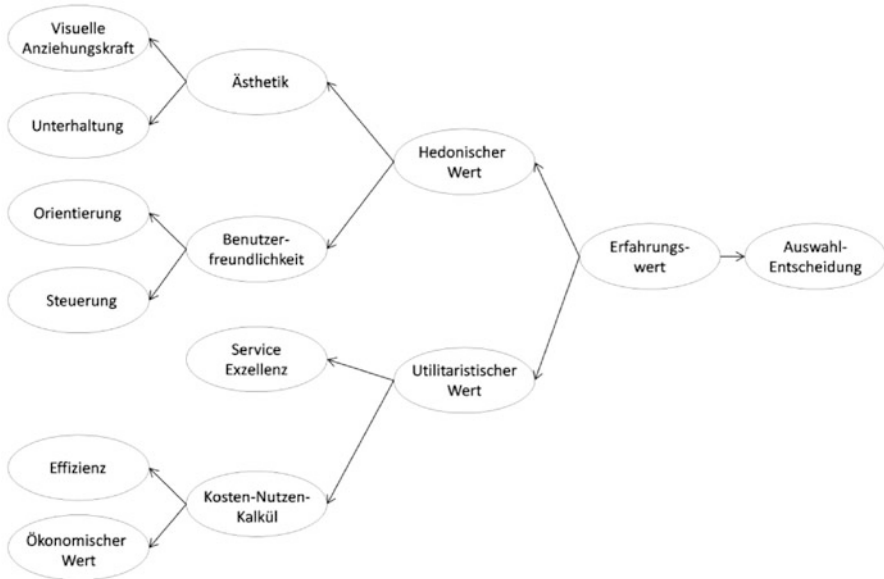


Abb. 1.7 Einflussfaktoren des Erfahrungswertes digitaler Güter (in Anlehnung an Mathwick et al. 2001)

Hedonische Merkmale sprechen dagegen die oben angeführten persönlichen und situativen Bedürfnisse von Konsumenten an. So spielen ästhetische Vorlieben von Konsumenten bei der Auswahl eine große Rolle wie auch die Benutzerfreundlichkeit der Software. Beide Faktoren umfassen konkretere Subkriterien, wie z. B. die visuelle Anziehungskraft (etwa Farbgestaltung, 3D-Effekte) oder die Art und Weise der Steuerung der Software (z. B. mit Maus, Tastatur, Controller), die wiederum spezifische, individuelle Kaufmotive des Konsumenten ansprechen. Insgesamt lassen sich über die Beeinflussung der Subkriterien die darüberliegenden hedonischen und utilitaristischen Werte und somit der gesamte Erfahrungswert einer Software beeinflussen, der wiederum einen Einfluss auf die Software-Auswahl und Kaufentscheidung des Konsumenten ausübt. Die Subkriterien lassen sich in diesem Sinne mehr oder weniger stark vonseiten eines Softwareanbieters beeinflussen. Die Gestaltung von Geschäftsmodellen für Softwarehersteller ist Gegenstand des folgenden Abschnitts.

1.4 Geschäftsmodelle in der Softwareindustrie

In diesem Abschnitt wollen wir eine Klassifikation von Geschäftsmodellen für die Softwareindustrie vorstellen (Popp und Meyer 2010).

Ein Geschäftsmodell definiert allgemein, welche Güter oder Dienstleistungen von einem Unternehmen angeboten werden, und wie es dafür entlohnt wird.