

Markt- und Geschäftsmodelle mit Open Source Software

Markus Nüttgens
Universität Trier
Wirtschaftsinformatik II
(Lehrstuhlvertretung)
Postfach 3825, D-54286 Trier
E-Mail: markus@nuettgens.de

Abstract: Das Open Source Konzept eröffnet Anbietern und Nachfragern von Software neue Wege zur Erprobung alternativer Markt- und Geschäftsmodelle. In diesem Beitrag werden zunächst die Grundideen und Denkansätze der konventionellen und der Open-Source-basierten Softwarekonzepte diskutiert und die wesentlichen Merkmale und Merkmalsausprägungen herausgearbeitet. Dies schließt eine Erläuterung der Entstehungsphasen und den spezifischen Charakteristika der Entwicklung und Distribution von Open-Source-Software ein. Hieraus ergeben sich verschiedene strategische Optionen für die Beratung in der Form einer Dienstleistungsgenerierung und (Re-)Finanzierung „offener“ Softwareprojekte. Auf der Grundlage der Positionierungsmöglichkeiten werden vier wesentliche Normstrategien für Anbieter von Open-Source-Produkten und -Dienstleistungen diskutiert und bezüglich Ihrer Umsetzungspotenziale bewertet.

1 Best Practice kostenlos?

Im Internet werden Daten zur flüchtigen Ware. Nahezu alle Musiktitel und selbst ganze Kinofilme sind – nicht immer zur Freude der Inhaber von Lizenzrechten – digitalisiert verfügbar. Ein weiteres Beispiel für „netzfähige“ Produkte ist Software. Hersteller nutzen das Internet seit langem als Hauptvertriebskanal und stellen Software häufig bewusst kostenlos zur Verfügung. Auf diese Weise können Interessenten z.B. Vorabversionen („beta-Versionen“) auf ihre Funktionstüchtigkeit prüfen und Fehler an den Hersteller melden. Die Idee dieser vernetzten Qualitätskontrolle ist nicht neu. Unter dem Begriff „Open Source“ ist man bereits einen Schritt weiter und vernetzt über ein offenes Lizenzmodell alle Akteure und Lebenszyklen eines Software-Produktes (vgl. *Open Sources, Open Source Revolution 1999; O'Reilly, Open Source 2000*).

Ausgangspunkt ist hier die Frage, warum die Nutzer von Software diese über das Testen hinaus nicht auch gleich verbessern oder sogar erweitern sollten. Als Konsequenz wird der jeweilige Quellcode von Open Source-Software – im Gegensatz zu herkömmlichen Software-Geschäftsmodellen – offen und kostenlos im Internet zugänglich gemacht. Auf diese Weise werden die Fähigkeiten von Entwicklern zur Qualitätssteigerung und Erweiterung weltweit genutzt. Erste Erfahrungen der letzten Jahre haben gezeigt, dass Fehlfunktionen von Software durch diese vernetzte Art der Entwicklung auf einem wesentlich geringeren Niveau gehalten werden können, als bei nicht-öffentlichem

Software-Quellcode. Natürlich ist auch die Nutzung von Open Source-Software nicht kostenlos. Wie bisher entsteht ein Großteil der Total Cost of Ownership durch Einführung, Anwenderschulungen, Support etc.

Mit dem Betriebssystem Linux als prominentestem Beispiel ist Open Source jedoch von einer idealistischen Idee zu einer Bedrohung etablierter Software-Anbieter geworden (vgl. *OpenSource.Org, Halloween 2000*). Eines der derzeit erfolgreichsten freien Software-Projekte ist Apache, ein freier Web-Server, der die Spitzenposition unter den meist benutzten Web-Servern hält (vgl. *The Apache Software Foundation 2000*). Dieses Produkt ist derzeit unangefochtener Marktführer bei Web-Servern. Bei inzwischen mehr als zehn Millionen Servern weltweit laufen derzeit ca. 60 Prozent aller Sites unter Apache (vgl. *Netcraft, Web Server Survey 2000*). Das Projekt wird u.a. von IBM durch Bereitstellung von Entwicklern vorangetrieben.

Selbst Regierungen denken inzwischen darüber nach, ihre Verwaltungen mit Open Source-Software auszustatten, da sie nicht ausschließen können, dass bei proprietärer Software Dritten der Zugang zu sensiblen Daten ermöglicht wird. So plant Frankreich derzeit einen Gesetzesentwurf, dass zukünftig bei Neuentwicklungen im öffentlichen Sektor nur noch Open Source Software zugelassen werden soll. Spätestens mit der Offenlegung des Source Codes zu der Verschlüsselungssoftware PGP („Pretty Good Privacy“) wurde eindrucksvoll demonstriert, dass Open Source kein Sicherheitsrisiko darstellt sondern vielmehr erst eine Kontrolle und Einsichtnahme durch Dritte ermöglicht.

Für das Beratungsgeschäft werden die geschilderten Entwicklungen nicht folgenlos bleiben. Um Kunden aus strategischer Sicht bei der Gestaltung von Systemlandschaften zu beraten und in operativer Hinsicht bei der Einführung zu unterstützen, muss das Kompetenzportfolio entsprechend erweitert werden. Für die Beratungsunternehmen selbst zeichnet sich eine verminderte Abhängigkeit von einzelnen Herstellern ab. Darüber hinaus eröffnet sich die Chance, in Eigeninitiative spezifische Lösungen bzw. Erweiterungen zu entwickeln. Geschieht dies gemeinsam mit dem Kunden, entsteht eine neue Dimension der Kundenbindung.

Im folgenden wird das Open Source Konzept kurz erläutert (vgl. *vertiefend zu Konzept und Entwicklung von Open Source Nüttgens/Tesei, Open Source 2000*). Im Anschluß daran werden auf der Grundlage des Open Source Konzeptes Handlungsoptionen für Unternehmen und Berater aufgezeigt und bewertet.

2 Was ist Open Source?

Open Source bedeutet wörtlich aus dem Englischen übersetzt „Freie Quelle“. Gemeint ist damit die freie Verfügbarkeit des Software-Quellcodes, der beliebig genutzt und verändert werden kann. Dieser Ansatz steht zunächst dem klassischen Ansatz des Software-Vertriebs entgegen, der seinen Quellcode in der Regel vor Dritten schützt, um zu verhindern, dass dieser kopiert wird und eine unkontrollierte Verteilung der Software einsetzt.

Derzeit sind die Software-Geschäftsmodelle für „Open Source“ und „Closed Source“ noch scheinbar widersprüchlich. Tabelle 1 stellt die wesentlichen Merkmale der beiden Geschäftsmodelle gegenüber. Während traditionelle Geschäftsmodelle primär der linken Spalte entsprechen, tendieren Open Source Geschäftsmodelle zu den Merkmalsausprägungen der rechten Spalte.

Merkmal	Merkmalsausprägung	
	Traditionell	Open Source
Nutzungsrecht	Lizenzgebühr	Freie Weiterverbreitung
Plattform	Proprietär	Neutral
Programmcode	Kompiliert	Quellcode
Rechtsform	Unternehmung	Community
Kapitalisierung	Software	Zusatznutzen
Organisationsform	Zentral	Virtuell
Motivation	Monetär	Selbstverwirklichung
Autorenschaft	Anonym	Entwickler
Vertriebsform	Vertragshandel	Brokerage

Tabelle 1: Merkmale von Softwaremärkten

Die Begriffe „freie Software“ und „offene Software“ werden im folgenden synonym zu Open Source-Software gebraucht. Sie bringen zum Ausdruck, dass Software von jedem genutzt, kopiert und verteilt werden darf. Dies kann unentgeltlich oder gegen eine Gebühr geschehen, auf jeden Fall muss aber der Quellcode für jeden Interessenten verfügbar sein. Die Gebühr ist keine Lizenzgebühr für Software, sondern soll die Kosten des Anbieters für die Bereitstellung decken. Proprietäre Software steht dem Open Source-Konzept entgegen. Die Nutzung, (Neu-) Verteilung oder Modifikation von proprietärer Software erfordert in jedem Fall eine Genehmigung des Eigentümers.

2.1 Entstehungsphasen

Schon zu Beginn der Softwareentwicklung wurden in den 60er und 70er Jahren Programme zwischen Entwicklergruppen ausgetauscht, um sie zu lesen, zu verändern oder um Teile daraus in anderer Software zu verwenden (Software Sharing Communities). Im Jahr 1982 veröffentlichten die Unternehmungen IBM, HP und DEC kommerzielle Versionen des Betriebssystems Unix für ihre eigene Hardware. Mitglieder bestehender Gruppen wurden aus den wenigen Forschungsabteilungen der Unternehmungen und Universitäten abgeworben und arbeiteten zunehmend an kommerzieller Software. Die einfache Kommunikationsstruktur, die bis zu diesem Zeitpunkt ausreichte, genügte nun nicht mehr für den Zusammenhalt der vielen

einzelnen Gruppen und so kam es dazu, dass diese sich sukzessive auflösten. Es entstand ein Vakuum in der Software-Produktion, welches zu diesem Zeitpunkt nur Unternehmungen mit ihrer kommerziellen Software zu füllen wussten. Dies führte zu einer Abnahme sogenannter freier Software. Unternehmungen und Forschungseinrichtungen entschieden sich zunehmend für den Einsatz kommerzieller Software, die nicht offen, d. h. proprietär war. Die technologische Entwicklung unterstützte diesen Trend, in dem die Betriebssysteme an die Hardware gekoppelt wurden und somit alle neuen Prozessoren ihr eigenes, nicht offenes Betriebssystem mitbrachten.

Um den kooperativen Geist der Software-Sharing Communities zurückzubringen, initiierte der ehemalige MIT-Mitarbeiter Richard Stallman das GNU-Projekt, für das er die Free Software Foundation (FSF) gründete. Ziel war damals die Schaffung eines freien und offenen Unix-Betriebssystems (vgl. Müller, *Die Philosophie des GNU 1999*, S. 17). Obwohl nie wirklich ein Unix-Kernel realisiert wurde, ermöglichte die Vielzahl entstandener freier Systemsoftware und verschiedene Programmierwerkzeuge die Entwicklung von Linux Anfang der 90er Jahre. Ein Student namens Linus Torvalds hatte 1991 begonnen, einen freien Unix-Kernel für PCs mit Intel 80386 Mikroprozessor zu entwickeln. Sein schneller Erfolg zog Entwickler über das Internet an, welche ihm halfen, das Betriebssystem Linux als erstes vollfunktionsfähiges freies Unix zu entwickeln. Linux wurde zu dieser Zeit von vielen Entwicklern noch nicht wahrgenommen. Es dauerte noch etwa 5 Jahre bis sich dieser Trend einer breiten Öffentlichkeit zeigte. Erst Ende der 90er Jahre wurden die zentralen Aktivitäten der Entwickler auf das Internet und Linux gerichtet. Das World Wide Web (WWW) machte das Internet schließlich zum Massenmedium, wodurch sich die Anzahl potentieller Entwickler enorm vergrößerte. Unternehmungen, die wie HP, DEC oder IBM proprietäres Unix verkauften, praktizierten ein schlechtes Marketing, so dass die Firma Microsoft durch aggressives Marketing in der Lage war, mit dem Windows-Betriebssystem große Marktanteile zu erringen. Linux erlangte 1994 zunehmend Stabilität und etablierte sich als Plattform für Entwickler. Die verfügbaren Programmiersprachen und Werkzeuge animierten Entwickler, neue Projekte unter Linux zu entwickeln bzw. bereits bestehende dorthin zu portieren. Durch den Einsatz von Linux an Universitäten, Forschungseinrichtungen und Unternehmungen entstanden weitere Projekte, u. a. in den Bereichen Programmiersprachen, Datenbanken, Grafik oder Desktop. Die steigende Themenvielfalt, die freie Software abdeckt, führte zu einer Vergrößerung der Community der freien Software und zu einer steigenden Anzahl themenspezifischer Communities.

Die Erweiterung durch Mitglieder aus unterschiedlichen Interessensbereichen führte auch zu unterschiedlichen Auffassungen bezüglich der sinnvollen Definition freier Software. Es gab schon immer mehrere unterschiedlich ausgestaltete Lizenzen für freie Software und mit dem Engagement von Unternehmungen wie Netscape, Troll Tech oder IBM kamen noch einige neue hinzu.

Der Begriff Open Source wurde erst durch die Open Source Initiative, eine gemeinnützige Organisation mit dem Ziel, den Open Source Gedanken zu verbreiten, einer größeren Öffentlichkeit bekannt. Der Erfolg der Bezeichnung Open Source in der Öffentlichkeit war an vielen Artikeln über Linux und der bevorstehenden Freigabe des Browser-Quellcodes von Netscape zu erkennen. Am 23. Februar 1998 gab die Firma

Netscape bekannt, dass sie den Begriff Open Source nutzen würde. Der O'Reilly Verlag veröffentlichte ebenfalls eine Nachricht, in der er bekanntgab, den Begriff Open Source zu verwenden. Damit war das erste Ziel der Open Source Initiative erreicht und es folgten entsprechende Ankündigungen von Corel, Sun Microsystems, IBM, SCO, Hewlett Packard, Oracle, Informix, SGI, SAP. Veröffentlichungen in Computerzeitschriften und Magazinen wie The Economist, Computerworld und The Forbes Magazine belegen, dass freie Software von einer größeren Öffentlichkeit wahrgenommen wird.

Die angesprochenen Merkmale von Open Source Software werden in der Open Source-Definition zusammengefasst (Open Source, The Open Source Definition 2000). Da die Bezeichnung Open Source beschreibend ist, kann sie nicht als Warenzeichen oder Schutzmarke eingetragen werden. Aufgrund des Bedarfs der freien Software-Gemeinschaft nach einer verlässlichen Bezeichnung für Open Source Software hat die Open Source Initiative (OSI) ein sogenanntes „certification mark“ als Prüfsiegel registrieren lassen. Wenn das Prüfsiegel „OSI certified“ auf einer Software erscheint, steht sie unter einer Lizenz, die mit der Open Source Definition konform ist. Lizenzen sind keine Ware, folglich können sie auch nicht beim U.S. Patent and Trademark Office registriert werden können. Die Zertifizierung erfolgt in zwei Schritten:

1. OSI veröffentlicht eine Liste von Lizenzen, die der Open Source Definition genügen. Gibt es eine Lizenz, die noch nicht in der Liste steht und der Open Source Definition genügen soll, so kann diese zur Prüfung an <license-approval@opensource.org> gesandt werden.
2. Um die „OSI certified“ mark auf einer Software Distribution zu verwenden, muss die Software unter einer von der OSI veröffentlichten Open Source-konformen Lizenz stehen.

Das Prüfsiegel kann ausschließlich auf Software angewandt werden, nicht auf Lizenzen. Dies soll sicherstellen, dass ein Softwarepaket mit den enthaltenen Lizenzen eine Open Source Distribution ist.

2.2 Entwicklung und Distribution

Dem Open Source-Ansatz liegen spezifische Modelle hinsichtlich Entwicklung und Distribution zugrunde, die im folgenden kurz erläutert werden.

Die Vorgehensweise bei der Entwicklung freier Software war bis 1997 nicht dokumentiert und ausschließlich an bekannten freien Software-Projekten, wie z. B. Linux ersichtlich. Raymond hat im Mai 1997 erstmalig versucht, Faktoren einer erfolgreichen Vorgehensweise zu dokumentieren. Nach seinem Essay „The Cathedral and the Bazaar“ wird das Software-Entwicklungsmodell auch Basarmethode genannt. Raymond hat bei seiner Analyse des Software-Entwicklungsmodells von Linux folgende Grundregel aufgestellt (Raymond, *The Cathedral and the Bazaar* 2000):

„Veröffentliche früh und häufig, delegiere alles was sich delegieren lässt und sei offen bis zum Punkt des heillosen Durcheinanders, genannt Chaos.“

Weiterhin wurden folgende Regeln von ihm festgelegt:

- Jedes gute Programm hat seinen Ursprung in einer für den Entwickler störenden Unzulänglichkeit.
- Gute Programmierer wissen, was geschrieben werden muss. Große Programmierer wissen, was neu geschrieben werden muss und was wiederverwendet werden kann.
- Plane etwas zu verwerfen, denn du wirst es so oder so tun.
- Wenn man die richtige Einstellung hat, sind es die interessantesten Probleme, die einen finden.
- Wenn man das Interesse an einem Programm verloren hat, dann besteht die letzte Pflicht in der Weitergabe an einen kompetenten und fachkundigen Nachfolger.
- Seine Benutzer als Mitentwickler einzubeziehen, ist die einfachste Art, den Programmcode schnell zu verbessern und die Effizienz der Fehlersuche zu steigern.
- Veröffentliche früh, oft und höre auf die Kunden.
- Ist die Basis an Mitentwicklern und Beta-Testern ausreichend groß, dann wird nahezu jedes Problem schnell charakterisiert werden und die Lösung jemand offensichtlich sein.
- Debugging (Austesten und Fehlerbeseitigen) ist parallelisierbar.
- Wenn man seine Tester als die wertvollste Hilfsquelle behandelt, werden sie als eine solche reagieren.
- Nach guten eigenen Ideen ist das Erkennen guter Ideen anderer am besten. Manchmal ist letzteres aber noch besser.
- Oft kommen die besten und innovativsten Lösungen durch die Erkenntnis, dass der Lösungsansatz und das daraus resultierende Konzept falsch waren.
- Perfektion (im Design) ist nicht erreicht, wenn nichts mehr hinzuzufügen ist sondern wenn nichts mehr entfernt werden kann.
- Vorausgesetzt, der Projektleiter hat als Hilfsmittel ein Kommunikationsmedium mindestens von der Qualität des Internet und er versteht es, ohne Zwang zu führen, sind viele Köpfe unvermeidlich besser als einer.

Auf der Grundlage dieses Entwicklungsmodells hat sich ein komplementäres Distributionsmodell für Open-Source Produkte etabliert.

Entwickler stellen die Software im Internet für Benutzer zur Verfügung. Die Benutzer können selbst nach Software suchen, diese testen und einsetzen. Ein Benutzer kann per E-Mail in direkten Dialog mit Entwicklern treten und an der Entwicklung der Software teilnehmen, indem er die Software beurteilt und diese Beurteilung an die Entwickler sendet. Eine weitere Möglichkeit besteht darin, Bündel zusammengestellter Software bei einem Software-Händler zu kaufen. Im Unterschied zu proprietärer Software bezahlt man für freie Software ausschließlich für den Service der Bereitstellung, während bei proprietärer Software Lizenzgebühren zu entrichten sind. Konkret bedeutet dies, dass eine einmal erworbene freie Software nach Belieben verteilt werden kann. Die Distributoren stellen eigenständig Software-Pakete zusammen und verteilen diese gebündelt an Benutzer. Weiterhin hat der Benutzer die Möglichkeit, freie Software

direkt vom Entwickler zu beziehen. Distributionsformen sind hauptsächlich Download-Möglichkeiten via Internet und der CD-Versand, der entweder direkt nach schriftlicher, elektronischer oder telefonischer Bestellung erfolgt oder indirekt, indem CDs als Beilage in Fachzeitschriften und Sonderheften oder als Messe- und Werbegeschenke verteilt werden.

3 Strategische Optionen für die Beratung

Open Source stellt die Entwicklung proprietärer Software und daran anknüpfende Beratungsleistungen in Frage, bietet Software-Herstellern und Beratungen aber gleichzeitig große Chancen zur Neuausrichtung.

Abbildung 1 zeigt einen strategischen Handlungsrahmen IT-bezogener Beratungen im Open Source-Kontext. Deren Positionierung ergibt sich zum einen aus dem Beratungsfokus: Richtet sich die Beratung eng an einer spezifischen Anwendungssoftware und deren Funktionsumfang aus oder stehen branchen- bzw. funktionsbereichsspezifische Probleme im Vordergrund der Betrachtung, für die entsprechende Software-Pakete gesucht werden? Zum anderen erfolgt die Einordnung nach dem Entwicklungsansatz der primär empfohlenen oder einzuführenden Software-Lösungen, wobei zwischen Open Source und „Closed Source“ für proprietäre Systeme unterschieden wird.

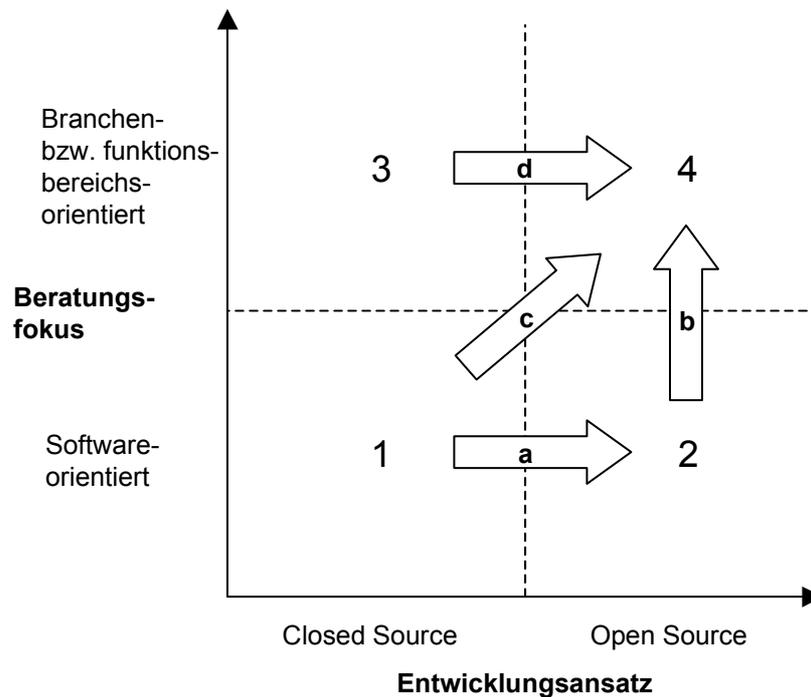


Abbildung 1: Strategischer Handlungsrahmen IT-bezogener Beratungen

Als Positionierungsmöglichkeiten stehen die einzelnen Felder für folgende Beratungsleistungen:

1. Einführung und Anpassung proprietärer Software-Systeme
Anbieter sind vor allem die großen IT-Beratungen im Standardsoftware- und Office-Umfeld.
2. Einführung und Anpassung von Open Source-Systemen
Anbieter sind primär Unternehmungen, die aus der Open Source-Community hervorgegangen sind und Open Source-Software testen, dokumentieren und distribuieren. Als Beratungsleistungen bieten sie Einführung, Schulung und Support an.
3. Entwicklung von technisch-organisatorischen Lösungen unter Einsatz proprietärer Software-Systeme
Anbieter sind bspw. Prozess-Beratungen mit hoher Branchen- oder Funktionsbereichskompetenz, die auf Basis von Organisationsanalysen proprietäre Systeme einführen.
4. Entwicklung von technisch-organisatorischen Lösungen unter Einsatz und Entwicklung von Open Source-Systemen
Anbieter sind Beratungen aus dem Organisations- oder IT-Umfeld, die technisch-organisatorische Lösungen konzipieren und durch den Einsatz spezifischer und teilweise selbst entwickelter oder erweiterter Open Source-Systeme umsetzen.

Erachtet man Open Source-Systeme aufgrund der aufgezeigten Potenziale als den proprietären Lösungen überlegen, werden die Positionierungen 1 und 3 mittelfristig zu Wettbewerbsnachteilen führen. Der Einsatz geschlossener Softwaretechnologien erschwert die kundenindividuelle Konfiguration und Entwicklung von Zusatzlösungen. Dies ist eine wesentliche Ursache für den vielzitierten „Anwendungsstau“. In Feld 2 positioniert, wird man nur teilweise an stark wertschöpfenden – und damit auch für die Beratung lukrativen – Beratungsprojekten beteiligt sein. Als IT-Dienstleister kann man aber auch hier strategische Erfolgspotenziale aufbauen. Die besten Aussichten werden in diesem Szenario Feld 4 eingeräumt, da mit auf Open Source basierenden technisch-organisatorischen Lösungen der größte Wert für den Beratungskunden geschaffen werden kann. Dem Open Source-Gedanken konsequent folgend, können in diesem Geschäftsmodell Beratungsdienstleistungen offen dokumentiert und damit standardisiert verfügbar gemacht werden, bspw. in Form von Referenzmodellen. Darüber hinaus können auf der Grundlage dieses „offenen“ Wissensmanagements strategische Kooperationen zur gemeinsamen Weiterentwicklung problemspezifischer Lösungen mit Kunden eingegangen werden, die eine langfristige Bindung begünstigen.

Diese Einschätzungen vorausgesetzt, können entsprechend der Pfeile in Abbildung 1 folgende Normstrategien abgeleitet werden:

- a) Aufbau technischer Open Source-Kompetenz
Ist die Kernkompetenz stark technisch orientiert, liegt eine Erweiterung dieser

Implementierungskompetenz auf den Open Source-Bereich nahe, um sich als spezialisierter IT-Dienstleister zu etablieren.

- b) **Aufbau von Branchen- oder Funktionsbereichskompetenz**
Ausgehend von einer starken Technikkompetenz im Open Source-Bereich können dem Kunden durch den Aufbau fachlicher Kompetenz größere Wertschöpfungspotenziale in Form spezifischer Weiterentwicklungen aus einer Hand erschlossen werden, die sich auch in höheren Beraterhonoraren niederschlagen. Allerdings kann auch die Positionierung als spezialisierter IT-Dienstleister sinnvoll sein. Ist der Marktanteil jedoch zu gering, bietet der Aufbau fachlicher Kompetenz die Möglichkeit zur Bildung von Alleinstellungsmerkmalen. Eine Alternative ist die Kooperation mit fachlich orientierten Beratungen.
- c) **Aufbau von technischer Open Source- sowie Branchen- oder Funktionsbereichskompetenz**
Entspricht größtenteils Strategie b); aufgrund der bisherigen technischen Ausrichtung hat der Aufbau von Open Source-Kompetenz Priorität.
- d) **Aufbau allgemeiner und technischer Open Source-Kompetenz**
Die anzustrebenden Kompetenzen liegen zum einen darin, einen jeweils aktuellen Überblick über bestehende Open Source-Lösungen zu haben, um diese zielgerichtet und eventuell verknüpft einsetzen zu können. Zum anderen muss auch technische Kompetenz aufgebaut werden, um Weiterentwicklungen bestehender Systeme vornehmen zu können. Eine Alternative ist die Kooperation mit implementierungsnahen IT-Dienstleistern.

Kurz- bis mittelfristig existiert darüber hinaus die Option, sich als reiner Open Source-Dienstleister von Feld 2 auch nach Feld 1 auszurichten, da (noch) erfolgreiche proprietäre Systeme mit Open Source-Systemen verknüpft bzw. Quellcode proprietärer Systeme öffentlich zugänglich gemacht werden. Daran wird deutlich, dass auch die Anpassungsstrategien der bisherigen Software-Hersteller beachtet werden müssen.

4 Ausblick

Das Open Source Konzept stellt eine alternative Form der Software-Entwicklung und des Software-Vertriebs dar. Es kombiniert bereits vorhandenes Wissen über Software-Entwicklung, -Vertrieb und -Organisationen mit den Effekten der Internet-Ökonomie. Durch die zunehmende Standardisierung von lizenzkostenfreien Software-Schnittstellen und -Formaten wird die Bedeutung proprietärer Softwarelösungen tendenziell abnehmen. Der Bezug benötigter Software aus Open Source-Quellen wie z. B. Software-Brokern oder anderen Unternehmungen erscheint bei mindestens gleicher Leistungsqualität attraktiver als die Abhängigkeit von Herstellern proprietärer Lösungen. Die bisherigen Lizenzgebühren werden dann zunehmend durch Dienstleistungsgebühren ersetzt.

Durch die offene und vernetzte Zusammenarbeit scheinen bessere Lösungen für Unternehmungen realisierbar zu sein. Die sich ergebenden Wertschöpfungspotenziale bieten Beratungsunternehmungen neue Möglichkeiten der Positionierung und Leistungsgestaltung.

Unternehmer fragen heute nicht mehr „Warum soll ich Produkte nutzen, für deren Qualität und Weiterentwicklung keine Unternehmung eine Gewährleistung übernimmt?“ sondern immer öfter „Warum soll ich Software kaufen, deren Qualität nicht öffentlich zur Diskussion gestellt wird?“ oder „Warum soll ich die Kontrolle über zentrale Unternehmungssysteme in die Hand anderer Unternehmungen legen?“

Im betriebssystemnahen Softwaremarkt hat sich – nicht zuletzt dank erfolgreicher Distributoren wie Redhat und SuSE - das Open Source Konzept bereits fest etabliert. Es bleibt abzuwarten, ob sich auch im anwendungsbezogenen Softwaremarkt eine vergleichbare Entwicklung abzeichnen wird. Hierbei kann die Beratungsbranche eine Schlüsselrolle einnehmen.

Literatur

Free Software Foundation, GNU 2000

Free Software Foundation (Hrsg.): GNU's Not Unix!, <URL: <http://www.fsf.org/>>, online: 01.09.2002.

Müller, Die Philosophie des GNU 1999

Müller, M.: Die Philosophie des GNU und die Pragmatik des Open Source, in: O'Reilly & Associates, Inc. (Hrsg.): Open Source – kurz & gut, Köln 1999, S. 17-19.

Netcraft, Web Server Survey 2000

Netcraft (Hrsg.): Netcraft Web Server Survey, <URL: <http://www.netcraft.com/survey/>>, online: 01.09.2002

Nüttgens/Tesei, Open Source 2000

Nüttgens, M.; Tesei, E.: Open Source, in: Scheer, A.-W. (Hrsg.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Nr. 156-158, Saarbrücken 2000.

OpenSource.Org, Halloween 2000

OpenSource.Org (Hrsg.): Halloween I: Open Source Software – A (New?) Development Methodology, <URL: <http://www.opensource.org/halloween/-halloween1.html>>, online: 01.09.2002.

Open Sources, Open Source Revolution 1999

Open Sources: Voices from the Open Source Revolution, Sebastopol 1999.

Open Source, The Open Source Definition 2000

OpenSource.Org (Hrsg.): The Open Source Definition, <URL: <http://www.opensource.org/osd.html>>, online: 01.09.2002.

O'Reilly, Open Source 2000

O'Reilly & Associates, Inc. (Hrsg.): Open Source – kurz & gut, Köln 1999.

Raymond, The Cathedral and the Bazaar 2000

Raymond, E. (Hrsg.): The Cathedral and the Bazaar, <URL: <http://www.tuxedo.org/~esr/writings/cathedral-bazaar/>>, online: 01.09.2002.

Stallman, GNU 2000

Stallman, R.: Richard Stallman on freedom and the GNU GPL, <URL: http://linuxworld.com/linuxworld/lw-1999-11/lw-11-rms_p.html>, online 01.09.2002.

The Apache Software Foundation 2000

The Apache Software Foundation (Hrsg.): The Apache Software Foundation, <URL: <http://www.apache.org>>, online: 01.09.2002.