

Der kommerzielle Einsatz von Open Source Software und kommerzielle Open Source-Geschäftsmodelle

Zur Bedeutung von Open Source Software in Unternehmen und als
Grundlage für Geschäftsmodelle

Raphael Leiteritz
Dipl. Inf.
Cand. MBA INSEAD

raphael@leiteritz.com
<http://www.leiteritz.com>

Raphael Leiteritz ist Unternehmensgründer und ehemaliger Wissenschaftler
an der TU Berlin. Im Moment absolviert er ein MBA-Studium am INSEAD
in Frankreich und Singapur.

Inhaltsverzeichnis

1	Einleitung	3
2	Open Source Software.....	4
2.1	Definition von OSS	4
2.2	Geschichte von OSS	5
2.3	Open Source vs. proprietäre Software.....	7
2.3.1	Unterschiede in der Wertschöpfung.....	7
2.3.2	Unterschiede in der Entwicklung.....	9
2.3.3	Unterschiede in den Lizenzen	10
2.4	Open Source Entwickler-Nutzer-Beziehungen	12
2.5	Beispiele bekannter Open Source Entwicklungsprojekte	14
2.6	Märkte	17
2.6.1	OSS-Produktmarkt	18
2.6.2	OSS-Dienstleistungsmarkt	24
2.6.3	OSS-Internetmarkt	25
3	Nachfragerseite: Kommerzieller Einsatz von OSS	28
3.1	Vorteile des kommerziellen Einsatzes von OSS	29
3.1.1	Sicherheit.....	29
3.1.2	Performance	31
3.1.3	Offenheit/Flexibilität.....	32
3.1.4	Stabilität	33
3.1.5	Preisvorteil	34
3.2	Grenzen des kommerziellen Einsatzes von OSS.....	36
3.2.1	Mangel an Applikationen.....	36
3.2.2	OSS auf dem Desktop	37
3.2.3	Investitionsschutz	38
3.2.4	Administrierbarkeit	38
4	Anbieterseite: Kommerzielle Open Source Geschäftsmodelle	40
4.1	Wertschöpfungsprozesse bei IT-Unternehmen	40
4.2	Modell zur Analyse von Geschäftsmodellen	42
4.3	Produkt-Geschäftsmodelle	43
4.3.1	Geschäftsmodell OSS-Distributor.....	44
4.3.2	Geschäftsmodell OSS-Applikations-Anbieter	50
4.3.3	Geschäftsmodell OSS-Appliance-Hersteller.....	55
4.4	Dienstleistungs-Geschäftsmodell	58
4.5	Mediator-Geschäftsmodell	62
4.6	Sonstige Geschäftsmodelle	66
4.7	Zusammenfassung.....	68
5	Die Zukunft von OSS im kommerziellen Bereich	73
6	Abkürzungen	75
7	Glossar.....	76
8	Literaturverzeichnis.....	79

*„There has to be a new, intermediate model in which the freeware community does what it is good at, and the commercial community does what it is good at (apart from screwing people), and both communities are better off for having cooperated with each other.“
Open Source-Entwickler Larry Wall, 3/1998*

1 Einleitung

In den letzten Jahren hat Open Source Software (OSS) große Resonanz in der Computerindustrie hervorgerufen. OSS und insbesondere der prominenteste Vertreter Linux hat sich mit teilweise dreistelligen jährlichen Wachstumsraten vom Hobby-System zum anerkannten Serversystem entwickelt.

Gleichzeitig sind zahlreiche wissenschaftliche Arbeiten zum Thema OSS entstanden. Diese beschäftigen sich mit den unterschiedlichsten Aspekten des Open Source Phänomens und reichen von technischen, soziologischen und psychologischen bis hin zu rechtlichen Analysen. Die vorhandenen Analysen der kommerziellen Aspekte von OSS hingegen sind relativ wenig umfassend und beschränken sich jeweils nur auf einzelne Gesichtspunkte.

Ziel dieser Arbeit ist deshalb eine umfassende Untersuchung von OSS im kommerziellen Bereich. Die Analyse wird dabei in die Nachfrager- (kommerzielle Nutzer von OSS) und in die Anbieterseite (Anbieter von Produkten und Dienstleistungen für OSS) unterteilt.

In Kapitel zwei werden die Grundlagen von OSS behandelt. Dabei werden die Entstehungsgeschichte, wichtige Eigenschaften von OSS und die Unterschiede zu herkömmlicher Software beschrieben. Anschließend erfolgt die Einordnung von OSS in den IT-Markt.

Im folgenden Kapitel drei werden Gründe für und gegen den Einsatz von OSS in Unternehmen und Institutionen geschildert und die Entwicklung des kommerziellen Einsatzes von OSS aufgezeigt. Welche Vor- und Nachteile ergeben sich für Nachfrager? Welche Stärken und Schwächen hat OSS im Vergleich zu herkömmlicher Software?

Neben der ursprünglich nichtkommerziellen OSS-Bewegung haben sich weltweit Unternehmen rund um OSS gegründet. Sie bilden die Anbieterseite von OSS. Welche Möglichkeiten gibt es, mit OSS Geld zu verdienen? Wie unterscheiden sich OSS-Geschäftsmodelle, welche OSS-Geschäftsmodelle sind tragfähig? Das vierte Kapitel untersucht diese OSS-Unternehmen und beschreibt die existierenden Geschäftsmodelle für OSS. Diese werden analysiert, verglichen und die Weiterentwicklungen und Veränderungen der Geschäftsmodelle werden aufgezeigt.

Die Arbeit schließt im Kapitel fünf mit einer Zusammenfassung des kommerziellen Einsatzes von OSS auf Nachfragerseite und der OSS-Geschäftsmodelle auf Anbieterseite und gibt einen Ausblick auf die kommerzielle Zukunft von OSS.

2 Open Source Software

In diesem Kapitel werden zunächst einige zentrale Fakten behandelt, die zum Verständnis der weiterführenden Analysen notwendig sind. Was ist Open Source Software? Was sind die wesentlichen Unterschiede zwischen Open Source und proprietärer Software? Im Verlauf wird die Definition von OSS und die Entstehungsgeschichte von OSS geschildert. Weiterhin werden in diesem Kapitel die wichtigsten Open Source-Projekte vorgestellt.

2.1 Definition von OSS

Unter OSS versteht man Software, deren Quellcode („source code“), also die geschriebenen Anweisungen des Programmierers, im Gegensatz zu herkömmlicher Software („proprietärer“ oder „closed source Software“) frei zugänglich ist.

Die wesentlichen Eigenschaften von OSS sind:¹

- Jeder hat das Recht, die Software nach eigenem Ermessen zu nutzen.
- Der Quelltext muss jedem Benutzer offengelegt werden oder es muß auf eine frei zugängliche Stelle verwiesen werden, wo er erhältlich ist.
- Der Benutzer hat das Recht, die Software zu modifizieren und in modifizierter Form weiterzuverteilen.
- Die Lizenz darf niemanden im Verkauf oder der Weitergabe der Software in Form einer Softwarezusammenstellung einschränken.

Im Gegensatz hierzu gilt für proprietäre Software:

- Vervielfältigung, Weiterverbreitung und Modifizierung sind untersagt.
- Das Nutzungsrecht wird in Form einer Lizenz erteilt.
- Eigentümer der Software ist nicht der Anwender, sondern weiterhin der Hersteller, denn er hat das Urheberrecht und die vollständige Kontrolle über das Produkt.

Mit der Aussage über die Quellcodeverfügbarkeit wird keine Aussage darüber gemacht, ob eine Software kostenpflichtig ist. OSS kann sowohl kostenfrei als auch entgeltpflichtig sein. Das gleiche gilt auch für proprietäre Software (die z. B. in Form von Freeware zwar proprietär, aber kostenfrei ist)².

Im folgenden werden die beiden Begriffe „OSS“ und „proprietär“ als Beschreibung für Software mit offenen und verborgenem Quellcode benutzt. Wenn zusätzlich eine Aussage über den Kostenaspekt der Software getroffen werden soll, wird der Begriff „kommerziell“ verwendet.

¹ Für eine vollständige Definition von OSS vgl. www.opensource.org.

² Eine umfassende Abgrenzung von Software-Kategorien ist unter www.gnu.org/philosophy/categories.de.html zu finden.

Neben dem Begriff „Open Source Software“ wird auch der verwandte Begriff „Free Software“ (Freie Software) gebraucht, wobei frei im Sinne von „Freiheit“ und nicht von „kostenlos“ gemeint ist. Der Unterschied zwischen den Begriffen ist, daß OSS eher technisch orientiert ist, wohingegen der Begriff der „Free Software“ breiter angelegt ist. Er stellt die Freiheit des Nutzers im Umgang mit der Software stärker in den Vordergrund und ist auch philosophisch und gesellschaftspolitisch motiviert³.

In dieser Arbeit wird durchgängig der Begriff der OSS als Überbegriff für alle Arten von Software mit offenem Quellcode verwendet, da er ideologiefreier und im kommerziellen Bereich gebräuchlicher ist.

2.2 Geschichte von OSS

Für das Verständnis der Entwicklung von OSS, der Vor- und Nachteile und der späteren Geschäftsmodellanalyse ist es wichtig, die Hintergründe von OSS zu kennen. Im folgenden werden deshalb in Anlehnung an O'Reilly (1999) die für das weitere Verständnis wesentlichen Meilensteine der OSS-Geschichte beleuchtet.⁴

Die Anfänge von OSS sind eng mit den Anfängen der Computerindustrie aus den Jahren 1960 bis 1970 verbunden. Zu dieser Zeit bestand das Computergeschäft fast ausschließlich aus dem Verkauf und der Wartung von Hardware. Software war ein Nebenprodukt, da die meisten Anwender ihre Software selbst entwickelten.

1969 entwickelten Ken Thompson und Dennis Ritchie in den AT&T Bell Telephone Laboratories die erste Version von Unix. AT&T, an der kommerziellen Verwertung gehindert, lizenzierte Unix für einen nominellen Betrag an Universitäten und gegen hohe Summen an kommerzielle Nutzer. Das Ergebnis war eine rege Entwicklungstätigkeit im universitären Umfeld. Ein Austausch der Programme war problemlos möglich, da nahezu alle Universitäten über eine Unix-Lizenz verfügten. Als Koordinationsstelle fungierte die Universität von Berkeley, Kalifornien, die einen eigenen Zweig von Unix, die Berkeley Software Distribution (BSD), entwickelte und an andere Universitäten vertrieb.

Das Jahr 1982 ist das Geburtsjahr des kommerziellen Unix. IBM, HP und DEC veröffentlichten Unix-Versionen unter neuen Namen für ihre eigene Hardware. Sun Microsystems wurde gegründet. AT&T kündigte offiziellen Support für Unix an und veröffentlichte die erste kommerzielle Version von Unix.

Nach einem Kartellverfahren trennte sich AT&T 1984 von 26 Firmen der Bell-Gruppe und durfte fortan als Wettbewerber im Computergeschäft auftreten. Damit nahm die Ära der liberalen Unix-Lizenzierung, des Zugriffs auf den Quellcode sowie des Austauschs von Programmen und Verbesserungen, ein Ende. Die Lizenzgebühren für Unix wurden drastisch angehoben.

Im Schatten dieser Veränderungen gründete Richard Stallman das GNU-Projekt (GNU is not Unix) mit dem Ziel, ein freies Unix zu schaffen. Als Koordinationsstelle und zur

³ Die Geschichte der Free Software Foundation wird in Kapitel 2.2 behandelt. Die Begründung für die bevorzugte Verwendung von „Free Software“ gegenüber „Open Source Software“ liefert www.fsfeurope.org.

⁴ Eine ausführliche Darstellung der Geschichte von OSS ist nicht Ziel dieser Arbeit und wurde auch schon an anderen Stellen vorgenommen (vgl. z. B. Grassmuck 2000).

Erwirtschaftung von Einnahmen durch den Versand der im Rahmen des GNU-Projektes erstellten Software und Dokumentation wurde 1984 die Free Software Foundation gegründet. Stallman betrachtete es als ein natürliches Recht, seine Programme mit seinen Kollegen zu teilen, insbesondere vor dem Hintergrund, daß das Verteilen von Software im Gegensatz zu physikalischen Gütern nahezu ohne jeglichen Aufwand und mit nur marginalen Kosten zu erreichen ist. Durch die zu dieser Zeit immer restriktiver werdenden Lizenzen der kommerziellen Softwareanbieter sah sich Stallman in diesem bis dahin üblichen Umgang mit Software gehindert.

Mit der GPL (GNU General Public License) entwickelte Stallman den Begriff des Copyleft als Wortspiel auf das ihm so unbequeme Copyright. Hauptaufgabe des Copyleft ist sicherzustellen, daß Software, die ihm unterliegt, frei bleibt, d. h. jeder, der sie benutzt und weiterentwickelt, hat die gleichen Rechte wie der ursprüngliche Autor. Dieses Ziel stellt die GPL durch die Forderung sicher, daß jedes Programm, das auch nur eine Zeile Code enthält, der der GPL unterliegt, wieder der GPL unterliegen muß.

Diese Eigenschaft macht die GPL nur schwer geeignet für die Entwicklung kommerzieller Software, da sie mit jeder anderen Lizenz, die mehr Einschränkungen enthält, inkompatibel ist. Die Akzeptanz von OSS beschränkte sich daher über 10 Jahre lang fast ausschließlich auf den privaten und universitären Bereich.

1991 startete der Finne Linus Torvalds das Linux-Projekt mit dem Ziel, eine freie Unix-Alternative für den IBM PC zu entwickeln. Damit stand das letzte fehlende Element, der Kernel (das „Herzstück“ jedes Betriebssystems), für das GNU-System zur Verfügung. Erstmals in der Geschichte der Datenverarbeitung war ein komplett freies Betriebssystem verfügbar.

Der Open Source-Protagonist Eric Raymond begründet die Triebfeder für die OSS-Entwicklung in seinem berühmten Aufsatz „The Cathedral and the Bazaar“ später so: „Every good work of software starts by scratching a developer's personal itch“ – OSS-Autoren entwickeln also zuallererst Lösungen für eigene Probleme.

1998 erregte Netscape Aufsehen mit der Ankündigung, den weit verbreiteten, aber durch den Internet Explorer von Microsoft bedrängten Browser „Netscape Navigator“, als OSS freizugeben.

Die Jahre 1999 bis 2000 waren geprägt durch ein stürmisches Wachstum von OSS und insbesondere Linux. Dies äußerte sich in einer wachsenden öffentlichen Aufmerksamkeit und einem Marktanteil von Linux im Serverbereich von 25% im Jahre 2000.

Die weltweite Software-Industrie wird von amerikanischen Unternehmen dominiert.⁵ OSS ist dagegen ein primär europäisches Phänomen. Der Ursprung von Linux liegt mit Linus Torvalds in Finnland. Die Mehrzahl der an OSS mitarbeitenden Entwickler stammen aus Europa (53,7%), wobei Deutschland eine führende Position in Europa einnimmt (21,2% aller Entwickler weltweit kommen aus Deutschland) (Robles 2001). Auch der Verbreitungsgrad von Linux ist nirgendwo höher als in Europa (Quellen IDA/Unisys 2001, Lancashire 2001, Wheeler 2001a).

⁵ Eine ausführliche Analyse der Gründe liefern Hoch et al. 1999 (S. 28 ff.)

2.3 Open Source vs. proprietäre Software

OSS unterscheidet sich in einigen Punkten wesentlich von proprietärer Software. Für die weitergehende Untersuchung der Bedeutung von OSS auf Nachfrager- und Anbieterseite ist es wichtig, diese wesentlichen Unterschiede zwischen diesen konkurrierenden Softwarekonzepten zu verstehen. Dabei interessieren insbesondere folgende Punkte:

- Welche Unterschiede in der Wertschöpfung gibt es?
- Wie unterscheiden sich die Entwicklungsprozesse?
- Welche unterschiedlichen Lizenzmechanismen liegen zugrunde?

2.3.1 Unterschiede in der Wertschöpfung

In den Anfängen der IT-Industrie dominierte das Geschäft mit Hardware. Software wurde als Ergänzung betrachtet, die teilweise sogar verschenkt wurde, um die Hardware attraktiver zu machen. Der Markt war in der Hand von wenigen Anbietern.

In den neunziger Jahren begann der Erfolg des Client-Server-Prinzips. Die großen Rechnersysteme wurden leistungsfähiger, kleiner und preiswerter, die Personal Computer erreichten den Arbeitsplatz. Auch die Betriebssysteme wurden leistungsfähiger und für normale Anwender bedienbar. Gleichzeitig wurden insbesondere mit dem Betriebssystem Windows (und anfänglich noch mit OS/2) erstmals ein einheitlicher Industriestandard geschaffen.

Durch diese Umstände wurde der Software-Markt immer wichtiger. Auf der gesamten Welt wurden tausende von Entwicklungsunternehmen mit dem Ziel gegründet, Software für unterschiedlichste Anwendungsfelder, basierend auf dem einheitlichen Industriestandard, zu entwickeln.

Diese Softwareentwicklungsunternehmen investieren in Forschung und Entwicklung der Software. Zusätzlich ergeben sich noch weitere Aufgaben wie Beratung, Implementation/Integration und Unterstützung im laufenden Betrieb. Diese späteren Schritte in der Wertschöpfungskette können dabei auch über Partner angeboten werden. Hierzu schließen sich die Entwicklungsunternehmen mit Dienstleistern (Systemintegratoren bzw. Value Added Reseller (VAR)) zusammen, die keine eigene Software entwickeln. Die Wertschöpfung dieser Partnerunternehmen liegt in der individuellen Beratung der Kunden und der Anpassung an spezielle Kundenbedürfnisse.

Diese Vorgehensweise hat für das Softwareentwicklungsunternehmen den Vorteil, daß es sich auf die Softwareentwicklung konzentrieren kann. Vertrieb und Marketing werden abgekoppelt. Mit der entsprechenden Anzahl an Partnern kann dabei sogar ein Multiplikatoreffekt erzielt werden.

Das Verfahren kann anhand des folgenden Bildes in einer Kette von Wertschöpfungsschritten dargestellt werden (angelehnt an Cimetiere o.J. und Zerdick et al. 1999):

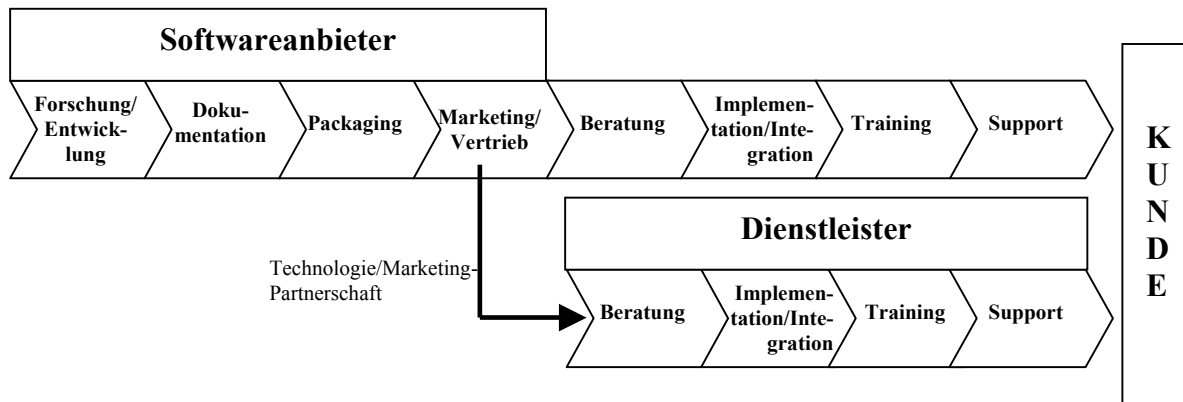


Abbildung 1: Die klassische Wertschöpfung der Softwareindustrie⁶

Die Softwareentwicklungsunternehmen sind damit (neben der Innovation im universitären Bereich) die eigentliche Quelle der Innovation in der Softwareindustrie.

Mit dem Aufkommen von OSS hat sich dieses Innovationsmonopol verändert. OSS ist mit Quellcode und (meist) kostenfrei erhältlich. Das Entwicklungsmonopol wird aufgebrochen, jede Partei kann die Software verändern, etwas hinzufügen und anpassen. Der Prozeß der Softwareentwicklung wird dadurch auf mehrere Akteure verteilt, offener, transparenter und vielfältiger.

Die folgende Abbildung veranschaulicht die veränderte Konstellation durch OSS:

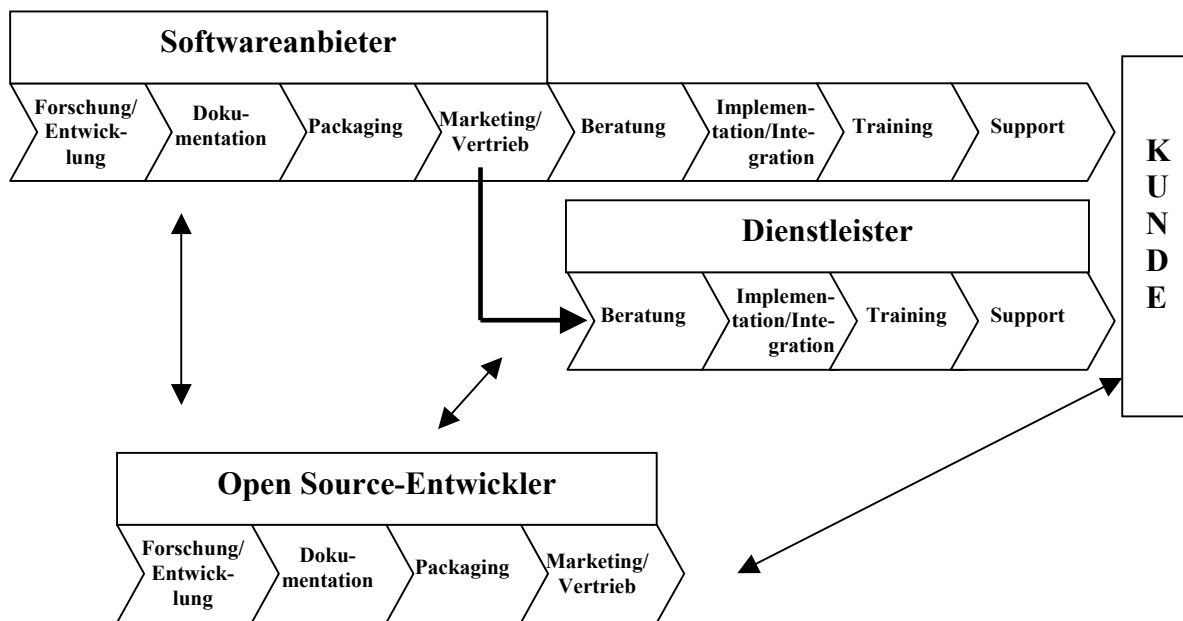


Abbildung 2: Veränderung der Wertschöpfung durch OSS

⁶ Das Konzept der Wertschöpfungskette für die Beschreibung von Unternehmensprozessen geht zurück auf Porter (1995).

Durch die neuen OSS-Entwickler verändert sich dabei nicht nur das Monopol der herkömmlichen Softwareentwicklung. Auch die Wertschöpfungskette selber kann dabei verändert werden, denn der Quellcode steht jedem Marktteilnehmer offen:

- Dienstleister können Veränderungen am Quellcode vornehmen und auf individuelle Kundenbedürfnisse anpassen.
- Softwareanbieter müssen Software nicht von Grund auf selber entwickeln, sondern können auf eine OSS-Basis zurückgreifen (vgl. Geschäftsmodell OSS-Appliance-Hersteller in Kapitel 4.3.3).
- Nicht zuletzt kann auch der Kunde selber das Endprodukt beeinflussen und erhält dadurch eine größere Unabhängigkeit.

Howe (2000) prognostiziert die Veränderung der Softwarewertschöpfung durch die OSS-Entwicklung wie folgt:

„Proprietary software vendors will suddenly see software development as an unfair fight: their mercenary band of captive developers against a battalion of Internet-armed revolutionaries“
Howe (2000), zitiert nach Delio (2000) (S. 1).

Das Konzept der Wertschöpfungskette wird im Kapitel 4 bei der Untersuchung der OSS-Geschäftsmodelle weiter vertieft.

2.3.2 Unterschiede in der Entwicklung

In der klassischen IT-Industrie wird Software in Unternehmen entwickelt. Dabei wird ein interner, zentralisierter und klar organisierter Softwareentwicklungsprozeß verfolgt (vgl. Hoch 1999) und entsprechende Infrastruktur genutzt.

Das OSS-Phänomen überrascht auch deshalb so sehr, weil es vergleichbare, in manchen Aspekten sogar überlegene, Software produziert, ohne über diese Entwicklungsstrukturen zu verfügen.⁷ OSS-Projekte werden durch Projektmitglieder, die über das Internet miteinander verbunden sind, dezentral entwickelt.⁸ Dabei existieren in der Regel keine hierarchischen Beziehungen zwischen den Mitgliedern, sondern es gilt die informelle Autorität der besten Entwickler. Die Entwicklung erfolgt nicht „top-down“ wie bei klassischer Softwareentwicklung, sondern „bottom-up“.

Bei proprietärer Softwareentwicklung gibt es eine klare Trennung zwischen Entwicklungsphase und Testphase. Beta-Tester suchen nach Fehlern in der Entwicklung und melden diese an die Programmierer. Bei OSS kann jeder Anwender gleichzeitig Tester und Entwickler sein. Aufgrund des offenen Quellcodes haben alle Personen die Möglichkeit, den Quellcode zu lesen, zu verändern und damit zu Mitentwicklern zu werden.

⁷ Eine detaillierte Darstellung aller Aspekte des OSS-Entwicklungsprozesses ist nicht Schwerpunkt dieser Arbeit. An dieser Stelle sei nur auf die Grundzüge der OSS-Entwicklungsstruktur hingewiesen, für eine Vertiefung empfehlen sich Grassmuck (2000), Rosenberg (2000), Raymond (1998a/b), Ardal (2001).

⁸ Nach Ardal (2001) beträgt die Zahl der OSS-Entwickler weltweit über 40.000.

Raymond (1998b) vergleicht die beiden Entwicklungsmodelle als Kathedralen- (klassische Softwareentwicklung) und Basar-Modell (OSS-Entwicklung). Die Kathedrale wird monolithisch und starr erbaut, während der Basar als Modell für ein auf den ersten Blick unorganisiertes Durcheinander steht, auf den zweiten Blick aber eine besondere Leistungsfähigkeit aufgrund des starken Austausches und der Partizipation der Mitglieder erlangt. Die Wechselwirkung, die gegenseitige Anspornung, die gemeinsame Fehlersuche und das kollektive „Brainstorming“ führen dabei zu der besonderen Qualität der OSS.

Bezroukov (2001) und andere haben später das Modell von Raymond kritisch untersucht und auf Schwächen in der Argumentation hingewiesen. Zum Beispiel stellt Bezroukov in Frage, ob der Prozeß der Linux-Kernelentwicklung wirklich dem Basar-Modell entspricht. Er sieht den Prozeß, der stark um die „Gallionsfigur“ Linus Torvalds fokussiert ist, als Beispiel für einen in Wirklichkeit sehr zentralisierten Prozeß, der eher dem Kathedralenmodell entspräche. Nichtsdestotrotz ist das Modell von Raymond ein Meilenstein in der Geschichte von OSS und stellte einen ersten umfassenden Erklärungsansatz für das OSS-Entwicklungsmodell dar.

Ein über das Basar/Kathedralen-Modell hinausgehender Erklärungsansatz ist das Modell des „cooking pot“ (Gosh 2000). Im Basar-Modell kann der Besitzer eines Gutes zwar wechseln, aber der Wert bleibt gleich. Beim cooking-pot-Modell schaffen alle Zutaten (ingredients) im Kochtopf einen Wert, der höher ist als die Summe aller Einzelwerte der Zutaten.

2.3.3 Unterschiede in den Lizenzen

Die Lizenzierung von Software spielt in der klassischen Software-Industrie eine große Rolle. Proprietäre Software wird nach Bessen (2001) mit Hilfe von vier Typen von Schutzinstrumenten abgesichert:

- Verborgener Austausch (trade secrecy): Die Software wird als Binärprodukt⁹ ausgetauscht. Ein reverse engineering (Schließen auf den ursprünglichen Quellcode) ist dadurch praktisch nicht möglich. Weitere Aspekte sind Verschwiegenheitsvereinbarungen mit Mitarbeitern und Partnern, geltende Rechtssprechung und technische Kopiermechanismen.
- Lizenzen: Durch die Lizenzgestaltung kann der Hersteller die Verteilung der Software beschränken. Der Kunde erhält nur ein Nutzungsrecht.
- Copyright: Proprietäre Software ist urheberrechtlich geschützt und dadurch nicht 1:1 kopierbar. Vergleichbare Produkte müssen sich technisch unterscheiden.
- Patente: Seit Mitte der 80er Jahre ist Software in den USA patentfähig. Mit Hilfe von Patenten können hinter der Software liegende Erfindungen („software-ideas“)¹⁰ geschützt werden.

⁹ In Maschinencode übersetzter Quellcode, nicht mehr vom Menschen lesbar.

¹⁰ Aus Sicht des Autors ist Bessen bei der Bezeichnung „software ideas“ etwas ungenau. Nicht die „Idee“ für eine Software ist patentierbar, sondern der Lösungsweg oder das Verfahren, um die Idee umzusetzen. Vgl. Gehring 2001.

Hinter dem Erfolg von OSS dagegen stehen offene Lizenzen, die dem Anwender einen hohen Freiheitsgrad lassen. Die wichtigste Open Source-Lizenz ist die GPL-Lizenz (GNU General Public License). Zentrale Inhalte der GPL sind:¹¹

- Freies Kopieren, freies Verteilen der Software (nicht notwendigerweise kostenfrei!).
- Der Source Code muß dem Benutzer zugänglich gemacht werden.
- Änderungen am Quellcode dürfen gemacht werden, wenn dies vermerkt wird.
- Haftungsausschluß.
- Weiterentwicklungen müssen auch der GPL unterliegen.
- Die GPL darf nicht eingeschränkt werden.

Die GPL-Lizenz besitzt einen „viralen Charakter“, verbreitet sich also immer weiter. Die Veränderung des Quellcodes ist erlaubt, aber mit der Bedingung verbunden, daß diese Veränderung wiederum unter der GPL-Lizenz steht. Das macht die GPL-Lizenz für kommerzielle Softwareanbieter relativ unattraktiv, da im Fall der Benutzung von GPL-Komponenten immer der komplette Quellcode freigegeben werden muß (siehe auch die Beschreibung des Geschäftsmodells OSS-Applikationsanbieter in Kapitel 4.3.2).

Die GPL verbietet dem Anbieter nicht, mit der Software Geld zu verdienen. GPL-Software darf nach den Lizenzbestimmungen auch verkauft werden – allerdings darf der Käufer die Software umsonst weitergeben, insofern ist das Verfahren kommerziell nicht praktikabel. Zusatzleistungen wie z. B. die Vervielfältigung von Datenträgern oder das Anbieten von Support und Gewährleistung werden von der GPL gestattet (s. Geschäftsmodell OSS-Distributor und OSS-Dienstleister im Kapitel 4).

Die GPL ist nicht die einzige OSS-Lizenz. Das folgende Schaubild zeigt eine Reihe von weiteren Lizenzen, eine detaillierte Beschreibung geht allerdings über den Rahmen dieser Arbeit hinaus.

	Kostenlos	Freie Verteilung	Uneingeschränkter Gebrauch	Quellcode vorhanden	Quellcode modifizierbar	Derivate müssen freisein	Keine Vermischung mit proprietärer Software ¹²
Proprietär	-	-	-	-	-	-	-
Shareware	?	Ja	-	-	-	-	-
Freeware	Ja	Ja	Ja	-	-	-	-
BSD	Ja	Ja	Ja	Ja	Ja	-	-
LGPL¹³	Ja	Ja	Ja	Ja	Ja	Ja	-
GPL	Ja	Ja	Ja	Ja	Ja	Ja	Ja

Tabelle 1: Vergleich der Eigenschaften wichtiger OSS-Software-Lizenzen (Quelle Roehrl et al. 2002)

¹¹ Vgl. auch Kapitel 2.2

¹² Die in den Spalten genannten Kriterien entsprechen weitestgehend der offiziellen Definition von Open Source Software unter www.opensource.org.

¹³ Lesser General Public License.

2.4 Open Source Entwickler-Nutzer-Beziehungen

Bei proprietärer Software sind die Rollen klar verteilt. Das Innovationsmonopol liegt beim Entwickler, es gibt evtl. Dienstleister für die Integration und am Ende den Nutzer, der das Endprodukt in der Art und Weise einsetzt, in der es vom Entwickler geplant wurde. Die Entwicklung und Nutzung von OSS dagegen ist ein dynamischer Prozeß, der sich zwischen Inhabern unterschiedlicher Rollen abspielt.

Mit dem folgenden Schaubild will der Autor die beteiligten Parteien und den iterativen Prozeß der Softwareentwicklung beschreiben. Dabei kann eine Person auch mehrere Rollen einnehmen und diese können über die Zeit auch wechseln.

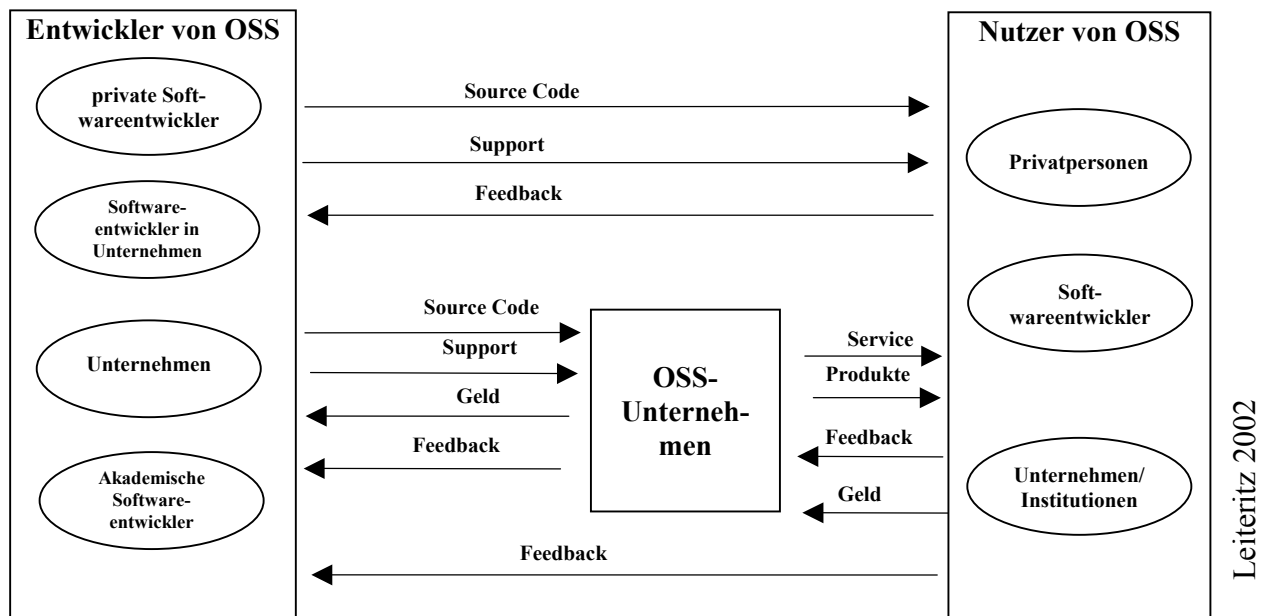


Abbildung 3: Rollen bei der OSS-Entwicklung

Die Parteien auf der Entwicklerseite sind für die Entstehung von OSS verantwortlich, die Parteien auf der Nutzerseite setzen OSS ein (wobei z. B. Softwareentwickler auch beide Rollen gleichzeitig einnehmen können). Zwischen ihnen findet der Austausch von Software auf der einen, aber auch Feedback auf der anderen Seite statt. Anhand dieses Feedbacks kann die Software iterativ verbessert werden. Die OSS-Unternehmen sind dabei entweder als Entwickler- (linke Spalte) und/oder als Mittler (Block in der Mitte) vertreten.

Entwicklerseite

Den Ursprung nimmt OSS durch die Entwickler. Diese können z. B. Privatpersonen sein, die die OSS-Entwicklung als „Hobby“ in ihrer Freizeit betrachten („private Softwareentwickler“). Diese können aber auch Personen in Unternehmen sein, die im Rahmen einer Anstellung

eigenverantwortlich an OSS arbeiten, also durch Unternehmen „gesponsert“ werden („Softwareentwickler in Unternehmen“). Das ist eine häufige Form der Unterstützung der OSS-Community durch OSS-Unternehmen, die dadurch zwar zum Entwicklungsprozeß beitragen, aber Art und Form des Beitrags den Mitarbeitern selber überlassen.

Es gibt Fälle, bei denen sich ein Unternehmen direkt an der OSS-Entwicklung beteiligt („Unternehmen“). Die Verantwortung liegt dann nicht bei einer Einzelperson, sondern das Unternehmen selber steuert das Projekt. Diese Unternehmen werden vom Autor nicht als „OSS-Unternehmen“ (mittlere Spalte) bezeichnet, wenn OSS keine primäre Rolle im Geschäftsmodell spielt.¹⁴

Zu Gruppe der „Akademischen Softwareentwickler“: Man könnte argumentieren, daß die Softwareentwicklern an den Universitäten eine Untermenge der privaten Softwareentwickler und dadurch keine eigene Gruppe darstellen. Erstens aber ist die Anzahl der akademischen Softwareentwickler relativ zahlreich (nach Robles 2001 sind ca. 25% der weltweiten OSS-Entwickler Studenten oder akademisches Personal), so daß sie eine wesentlich Gruppe darstellen. Zweitens wird OSS folglich nicht nur in der privaten Freizeit geschrieben, sondern entsteht aus gesellschaftlich geförderten Forschungsprojekten, Softwareprojekten oder dem akademischen Lehrbetrieb.¹⁵

Nutzerseite

Nutzer von OSS-Software können wiederum in verschiedene Rollen aufgeteilt werden. Diese sind erstens „Privatpersonen“, die OSS für private Zwecke einsetzen. Meistens sind dies ambitioniertere Nutzer von Computersystemen, die sich tiefergehend für OSS interessieren.

Eine weitere Rolle ist die der „Softwareentwickler“. Auch hier könnte man argumentieren, sie sind eine Spezialform der Privatpersonen. Allerdings drückt die eigene Rolle den Aspekt aus, daß viele OSS-Entwickler primär für sich selber oder für „ihresgleichen“ entwickeln (vgl. Raymond 1998a/b).

Diese beiden Rollen dominierten die Nachfragerseite von OSS in den Jahren bis ca. 1998. Ab 1998 begann die zunehmende Kommerzialisierung von OSS und damit wuchs auch das Interesse von „Unternehmen“ bzw. „Institutionen“. Hintergründe hierzu im Kapitel 3, in dem die Nachfragerseite von OSS untersucht wird..

Unternehmen als Mittler

Eine Zwischenposition nehmen die OSS-Unternehmen ein. Sie sind entweder selbst Entwickler, oder sie sind Mittler zwischen der Entwickler- und der Nutzerseite (wenn der

¹⁴ Ein Beispiel hierfür ist das Unternehmen Netscape, das zwar den Browser als OSS freigibt, dennoch aber nicht als OSS-Unternehmen bezeichnet werden kann, da der Browser in der Geschäftsstrategie (mittlerweile) nur eine untergeordnete Rolle spielt.

¹⁵ Einen ausführlichen Vergleich zwischen akademischer Forschung und OSS führt Bezroukov (1999).

Nutzer ein Unternehmen ist).¹⁶ Die wichtigste Ausprägung des Mittlers ist der OSS-Dienstleister. Dieses Modell wird im Kapitel 4.4 untersucht.

2.5 Beispiele bekannter Open Source Entwicklungsprojekte

Nachdem bisher die wichtigsten Eigenschaften von OSS und die wesentlichen Unterschiede zu proprietärer Software beleuchtet wurden, ist es sinnvoll, sich eingehender mit einigen konkreten OSS-Projekten zu beschäftigen. Die im folgenden vorgestellten OSS-Projekte (in Anlehnung an O'Reilly 1999) sind die Grundlage für die spätere Analyse der Nachfragerseite.

FreeBSD

FreeBSD stammt von der Berkeley Software Distribution Unix ab. Es begann im Jahr 1993 als eine Portierung des BSD 4.3-Lite Release auf eine Intel-Plattform. Das FreeBSD-Projekt besteht aus einem Kern von 16 Entwicklern, die den Vorstand bilden. Es gibt mehr als 120 Entwickler, die an dem Projekt arbeiten, die meisten davon unentgeltlich. Einige werden von ihren Arbeitgebern, die ein eigenes kommerzielles Interesse an FreeBSD haben, bezuschußt. Das Ziel von FreeBSD ist es, Quellcodes mit den geringstmöglichen Restriktionen zu verteilen. Der Großteil des Systems wird unter der Original-BSD-Lizenz vertrieben, die sowohl den kommerziellen und als auch den freien Einsatz der Software erlaubt. Die verbleibenden Teile sind unter der GNU Public License, der Artistic License und anderen Lizenzen erhältlich.

Linux

Linux als bekanntestes Open Source-Projekt ist so bemerkenswert, weil seine Entwicklergemeinschaft international zusammengesetzt ist und trotzdem ein sehr hohes Niveau der Kooperation aufweist. Im Mittelpunkt steht Linus Torvalds aus Helsinki, Finnland (mittlerweile in den USA). Während sich Unix schnell in mehrere Dutzend Versionen aufteilte, hat Linux es geschafft, als einheitliches System auf vielen unterschiedlichen Architekturen zu bestehen.

Immer mehr Hersteller wie z. B. Oracle, Informix, SAP R/3, Corel, Netscape usw. portieren ihre Software auf Linux. Einige Hardware-Hersteller, wie z. B. Sun und IBM, liefern ihre Maschinen auf Wunsch mit Linux aus. Für Sun gibt es bei den preiswertesten Maschinen gar keine Portierung ihres eigenen Solaris-Betriebssystems mehr.

DNS und BIND

Der BIND Daemon (Berkeley Internet Name Daemon) ist außerhalb der Technikszenen fast völlig unbekannt, aber jeder kennt die Dienstleistung, die er zur Verfügung stellt: das Domain Name System (DNS). Der Verbund von Name-Servern ermöglicht die Umsetzung von

¹⁶ Privatpersonen werden nur selten die Dienste von kommerziellen Serviceanbietern nutzen, für solche Probleme existiert eine Kultur der „Selbsthilfe“ durch andere OSS-Nutzer auf Internetportalen und in Internet-Newsgruppen.

Adressen wie 207.25.98.191 in eine menschenlesbare Form wie z. B. ibm.com, harvard.edu oder whitehouse.gov. Ursprünglich von Paul Mockapetris im Jahr 1984 entwickelt, wird die Entwicklung von DNS zur Zeit von Paul Vixie vom ISC (Internet Software Consortium) gesteuert.

Das DNS-System ist die Grundlage für die gesamte Internet-Industrie, einschließlich aller Internet Service-Provider, Produzenten von Web-Sites, der Haupt-DNS-Server und vieler Softwarefirmen. BIND ist fester Bestandteil aller Unix-Systeme und auch Microsoft liefert eine Version von BIND aus.

Sendmail

Sendmail wurde ursprünglich 1981 von Eric Allman entwickelt und ist mit ca. 75 bis 80% Marktanteil der dominierende Mail Transport Agents (MTA) im Internet. Sendmail nimmt Emails der Benutzer entgegen, leitet sie weiter und stellt sie dem Empfänger zu.

Apache

Apache ist eines der erfolgreichsten OSS-Projekte. Apache wurde zum am weitesten verbreiteten Server für Internetseiten (Netcraft 2001). Im Juni 1998 hat IBM die offizielle Unterstützung für die Apache-Gruppe angekündigt. IBM liefert Apache im Rahmen ihres WebSphere-Produktes aus. Die Ankündigung von IBM war ein wichtiger Schritt, mit dem OSS viel Akzeptanz im kommerziellen Bereich gewonnen hat.

Samba

Samba ermöglicht das Betreiben eines File- und Druck-Servers für Windows- Clientsysteme auf einem Linux/Unix-System. Es ersetzt somit einen herkömmlichen Windows-Server. Samba wurde von Andrew Tridgell an der Australian National University entwickelt.

Perl

Perl (Practical Extraction and Report Language) wurde 1986 von Larry Wall entwickelt und ist ein wichtiges Werkzeug für System- und Netzwerk-Administratoren sowie zur Internet-Programmierung. Große Sites wie Yahoo, Netscape, CNET, Amazon und Excite benutzen Perl, um ihre Web Sites zu managen und interaktive Dienste zur Verfügung zu stellen. Perl ist die am häufigsten benutzte Sprache zur Entwicklung von Internet-Diensten und interaktiven Datenbankabfragen. Es gibt etwa 500.000 Perl-Programmierer und viele Millionen Perl-Anwender.

Python

Python ist eine interpretierte, objektorientierte Sprache, die ein weiteres Beispiel der typischen Open Source-Entwicklung ist: Python fing als ein kleines Tool an und wurde schnell zu einem universellen Programmierwerkzeug. Guido von Rossum, der Chefentwickler von Python, sieht es als eine verbesserte Perl-Version.

KDE

KDE (K Desktop Environment) ist eine grafische Oberfläche für Unix/Linux. Es verbindet einfache Benutzung und graphisches Design mit der technologischen Stärke von Unix und wird inzwischen mit fast allen Linux-Distributionen ausgeliefert. www.kde.org ist die zentrale Web Site.

GNOME

GNOME (GNU Network Object Model Environment) ist ein GNU-Projekt mit dem Ziel, einen ausschließlich auf freier Software basierenden Desktop zu entwickeln und konkurriert mit KDE. GNOME wird von Red Hat unterstützt.

GIMP

Gimp (GNU Image Manipulation Program) wird häufig als das freie Photoshop bezeichnet. Das leistungsfähige Werkzeug zur Bildbearbeitung kann für die Fotoretouche, Bildkomposition und Bildmanipulation eingesetzt werden.

PGP

PGP (Pretty Good Privacy) gilt als der Standard in der Verschlüsselung von Emails. Der Autor Phil Zimmermann wählte das Modell der freien Software, um eine weite Verbreitung seiner Software sicherzustellen.

2.6 Märkte

Die bisherigen Ausführungen bezogen sich auf die Eigenschaften von OSS und die Unterschiede zu proprietärer Software. Da eines der Ziele dieser Arbeit die Untersuchung der Nachfragerseite (warum entscheidet sich ein Unternehmen für OSS?) ist, wird im folgenden Schritt der allgemeine IT-Markt und dann die relevanten OSS-Marktsegmente vorgestellt.

Für die Beschreibung des Marktes der Informationstechnologie im allgemeinen hat Hoch (1999) das folgende Modell entwickelt:

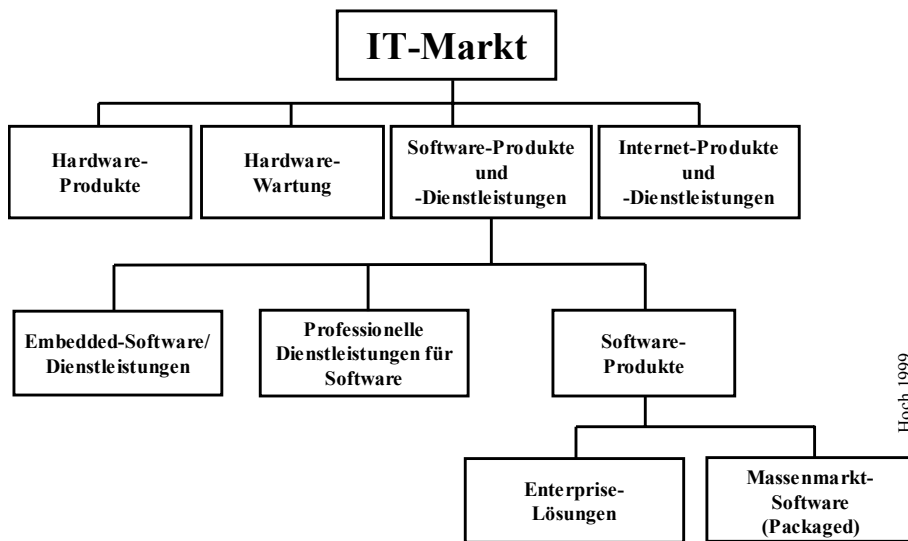


Abbildung 4: Struktur des IT-Marktes
Quelle: Hoch (1999)

OSS hat sich in vielen, aber nicht allen Segmenten des IT-Marktes entwickelt. So ist das Hardware-Segment nicht relevant, da es sich bei OSS im ursprünglichen Sinn per Definition um Software handelt.¹⁷ Die weitere Untersuchung der relevanten OSS-Teilmärkte wird in Produktmärkte, den Dienstleistungsmarkt und den Internetmarkt unterteilt.

Das Volumen des gesamten OSS-Marktes wird von der Unternehmensberatung WR Hambrecht (2000) für das Jahr 2000 mit \$2 Milliarden angegeben und in 2003 auf eine Größe von \$11 Milliarden geschätzt (Rosenberg 2000). Diese Schätzung umfaßt Hardware, Software, Dienstleistungen, online-Werbung, aber nicht den Embedded-Markt.

¹⁷ Es gab in den letzten Jahren zwar Ideen, das Prinzip von offenen Quellen auch auf Hardware, z. B. auf CPUs zu übertragen (Pomerantz 2000), aber diese Ansätze sind bisher nie über das theoretische Stadium hinausgekommen und haben deshalb keine Relevanz im Sinne von in die Praxis umgesetzten analysierbaren Geschäftsmodellen.

2.6.1 OSS-Produktmarkt

Der OSS-Produktmarkt ist nicht homogen, denn es gibt unterschiedliche Einsatzzwecke wie z. B. den Einsatz von OSS als Server- oder als Embedded-System. Deshalb wird der Produktmarkt im folgenden in neun Teilsegmente unterteilt:

	Server	Client	Embedded
Betriebssysteme	Linux-Distribution +	Linux-Distribution -	Linux-Distribution o
Applikationen	Server-Applikationen +	Desktop-Applikationen -	Embedded-Applikationen o
Appliances¹⁸	Server Appliances o	Thin Clients, Set Top Boxen o	PDA's, MP3-Player, Smartphones o

„+“ steht für hohe Relevanz, „o“ für mittlere Relevanz und „-“ für niedrige Relevanz.

Abbildung 5: OSS-Produktmarktsegmente und deren Relevanz im kommerziellen Bereich

Die Übersicht teilt sich dabei in den Spalten nach den Zielplattformen (Server, Client, Embedded) und in den Zeilen in den Umfang des Produktes (nur Betriebssystem, nur Applikation, Appliances). „Relevanz“ steht in der Abbildung für die Einschätzung der kommerziellen Bedeutung von Open Source/Linux (IDA 2001, IDC 2000 u.a.).

Der Software-Produktmarkt unterscheidet sich generell in einigen Eigenschaften von anderen Märkten. Hoch (1999) hat eine Reihe von Merkmalen definiert:

- **Niedrige Eintrittsbarrieren:** Die Markeintrittsbarrieren sind im Software-Markt gering. Softwareentwicklung ist wissensorientiert und deshalb leicht kopierbar.
- **Niedrige Investitionen:** Im Softwaremarkt sind keine Fabriken, keine Grundstücke und keine Maschinen notwendig. Für die ersten Schritte reichen Büroausstattung, Internetzugang, Hard- und Software und Softwareentwickler.
- **Schnelle Innovationszyklen:** Da die Eintrittsbarrieren niedrig sind und wenig Kapital benötigt wird, streben viele Anbieter auf den Markt. Diese setzen auf Innovationen und treiben bestehende Unternehmen wiederum zu neuen Innovationen.

¹⁸ Appliances sind Betriebssystem/Applikationen/Hardware-Kombinationen (z. B. vorinstallierte Email-Server).

- **Niedrige Grenzkosten:** Der erste Datenträger einer Software kann Millionen von Euro kosten, jeder weitere nur noch Cents. In der Softwareindustrie existieren praktisch keine Produktionskosten. Der gesamte Aufwand fällt in der Entwicklung und der Vermarktung der Software an. Deshalb wird Software in möglichst großen Stückzahlen für möglichst große Märkte produziert.
- **Kampf um die Marktführerschaft:** Softwareunternehmen profitieren überdurchschnittlich stark von hohen Marktanteilen. Dieses Phänomen ist unter dem Namen „law of increasing returns“ bekannt (Zerdick et. al 1999, S. 157). Produkte, die hohe Marktanteile haben und weiter Marktanteile gewinnen, wachsen wiederum überdurchschnittlich stark. Produkte mit fallenden Marktanteilen verlieren besonders viele Marktanteile. Der Grund für dieses Phänomen ist, daß der Nutzen für die Anwender bei hohen Marktanteilen größer wird (niedrigere Schulungskosten, Gewöhnung, einfachere Bedienung, Bevorzugung erfolgreicher Software, höhere Wechselkosten) und dies wiederum positives feedback für weiteres Wachstum erzeugt.

Im folgenden werden die Segmente des OSS-Produktmarktes weiter untersucht.

2.6.1.1 Der Markt für Betriebssysteme

Der Markt für Betriebssysteme kann in die drei Zielplattformen Server, Client und Embedded unterschieden werden.

Die Entwicklung im *Server*-Marktsegment kann mit Hilfe dieser Tabelle veranschaulicht werden:

	1997	1998	1999
Linux	6,8	15,8	25
Windows NT	36,0	38,3	38
Netware	26,4	22,8	19
Unix	20,7	18,8	15
OS/2	6,3	3,0	-
Sonstige	3,8	1,3	3

Tabelle 2: Marktanteile der Server-Betriebssysteme in Prozent
Quelle: Rosenberg 2000/IDC 2000

An dieser Übersicht ist neben dem imposanten Wachstum von Linux eine weitere Tatsache bemerkenswert. Linux ist das einzig wachsende Betriebssystem überhaupt und vollzieht dieses Wachstum insbesondere auf Kosten der Unix-Betriebssysteme. Der oftmals beschriebene Kampf „David Linux“ gegen „Goliath Microsoft“ findet offensichtlich in der Realität nicht statt. Im oberen Serversegment (große Festplattensysteme, große Nutzerzahlen) dominieren dagegen Unix-Betriebssysteme wie z. B. Solaris von Sun und AIX von IBM. Andere wichtige Unix-Serversysteme mit offenem Quellcode sind FreeBSD und OpenBSD, die vor allem im Bereich Sicherheit und Internetserver eingesetzt werden.

Die Gartner Group schätzt, daß der weltweite Linux Server-Markt von 2001 bis 2005 von \$2,5 Milliarden auf \$9,1 Milliarden wächst (IDA/Unisys 2001). Das Markt-

forschungsunternehmen Techconsult (2001) beurteilt die Branchenverbreitung von Linux bis 2003.¹⁹ Danach haben im kommerziellen Bereich die Energie- und Wasserversorger und die Telekommunikationsunternehmen mit 25 % den höchsten Linux-Einsatzgrad vorzuweisen.

Client-Systeme sind typische Endanwendersysteme. Auf ihnen laufen z. B. Office-Tools, Email-Systeme und Webbrowser. Der Client-Bereich wird von Microsoft und dem Betriebssystem Windows dominiert. Das Konkurrenzsystem Apple spielt nur eine untergeordnete Rolle vor allem im Bereich Multimedia, Design und Druckvorstufe. Der Marktanteil von Linux im Client-Bereich liegt bei ca. 4 Prozent (IDC 2000). Andere OSS-Systeme spielen auf dem Desktop keine Rolle.

Embedded-Systeme sind Rechnersysteme, die für den Benutzer unsichtbar verbaut sind. Sie werden in der Automobilindustrie, in kleinen tragbaren Geräten und in der Automatisierungstechnik verwendet. Typischerweise sind die Kunden nicht Endkunden, sondern Anbieter von Automatisierungs- oder Industrielösungen, die ein Betriebssystem kaufen und in eigene Lösungen integrieren. Der Embedded-Markt hatte 1999 ein Volumen von \$942 Millionen und wird im Jahr 2003 auf \$1.6 Milliarden geschätzt (Cook 2000). OSS ist in der Embedded-Industrie erst seit einigen Jahren vertreten. Allerdings wird OSS aufgrund der Flexibilität, Offenheit, den Preisvorteilen und der Stabilität eine große Zukunft prophezeit (s. Cook 2000, VDC 2000, EDC 2000), aber es existieren noch Schwächen z. B. im Echtzeitbereich.

Neben einer quantitativen Betrachtung könnten die Betriebssysteme auch nach der Vielfalt der Zielsysteme unterschieden werden. Der Autor hat dafür das folgende Schaubild entwickelt (Quellen: Wiehr et al. 2001, Wheeler 2001b, IDA/Unisys 2001, Rosenberg 2000). Mainframe-Systeme sind Server im obersten Leistungs- und Preissegment.

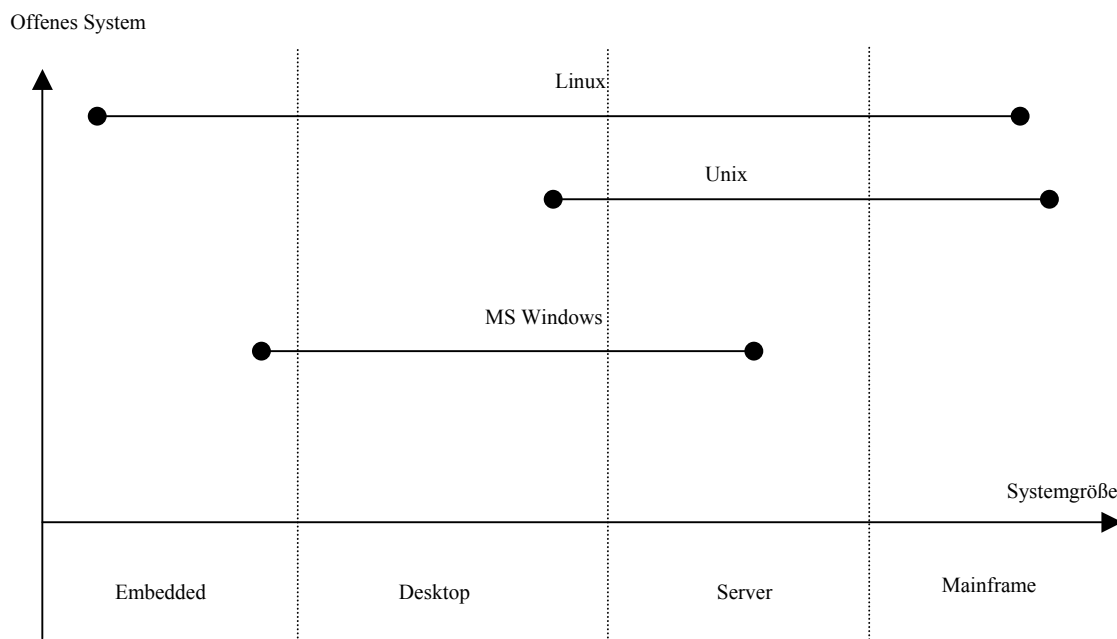


Abbildung 6: Linux ist das Betriebssystem mit der größten Anwendungsbreite

¹⁹ Für weitere Marktschätzungen s. IDA/Unisys 2001, IDC 2001 u. 2001, Weiss 2001, Howe 2000.

2.6.1.2 Der Markt für Applikationen

Auch der Markt für Applikationen kann in die drei Zielplattformen Server, Client und Embedded unterschieden werden:

Applikationen im *Serverbereich* werden für den Betrieb von zentralen Diensten benötigt. Beispiele sind

- Betriebswirtschaftliche Standardsoftware (Buchhaltung)
- Internet/Intranet-Server (Email, Webserver)
- System-Management-Software (Verwaltung von Computersystemen)
- Workflow-Software (Steuerung von Arbeitsprozessen)
- Datenbank-Software

Der Markt für Server-Applikationen ist sehr fragmentiert. Je nach Art der Applikation und der Zielgruppe existieren unterschiedlichste Produkte von einer großen Anzahl von Anbietern. Bekannte Anbieter sind z. B. SAP (Betriebswirtschaftliche Standardsoftware, auch als proprietäre/kommerzielle Lösung für Linux verfügbar), Oracle (Datenbank-Software, auch als proprietäre/kommerzielle Lösung für Linux verfügbar), Microsoft (Exchange Email Server) und Lotus/IBM (Workflow--Software).

OSS hat den Schwerpunkt im Bereich der IT-Infrastrukturlösungen wie z. B. Internet/Intranet-Server, Sicherheitssoftware, Datenbankserver. Es existieren zwar vereinzelt Unternehmenslösungen wie z. B. ERP-Systeme oder Workflowsysteme, allerdings sind diese in der Regel nicht Open Source, sondern werden als proprietäre/kommerzielle Software für den Linux-Server-Markt angeboten.

Applikationen im *Client*-Bereich werden für Endanwender entwickelt. Auch dieser Markt ist sehr fragmentiert. Je nach Art der Applikation und der Zielgruppe existieren unterschiedlichste Produkte von einer sehr großen Anzahl von Anbietern. Bekannter Vertreter ist z. B. die Office-Software von Microsoft (Word, PowerPoint, Excel). Daneben existieren z. B. Produktivitätstools für Projektverwaltung, Management Information Systeme oder Spiele. OSS spielt in diesem Marktsegment aufgrund des geringen Marktanteils keine bedeutende Rolle. Es existieren zwar Produkte im Client-Bereich (z. B. Staroffice/Openoffice als Bürosoftware, GIMP für die Grafikbearbeitung), aber der Markt bleibt eine Nische für technisch orientierte Anwender, denn die Bedienung und der Austausch mit kommerziellen Produkten bleiben schwierig (s. Kapitel 3.2, Grenzen des kommerziellen Einsatz von OSS).

Der Markt für *Embedded-Applikationen* ist im Gegensatz zum Server- und Client-Bereich kein Massenmarkt. Typische Kunden sind Industrieunternehmen, die eigene Embedded-Lösungen entwickeln. Es gibt eine überschaubare Anzahl von Anbietern wie z. B. Windriver, Lynx, Motorola und QNX, die neben dem eigentlichen Betriebssystem auch Applikationen wie z. B. Entwicklungsumgebungen, Managementsoftware und ähnliches anbieten. OSS-Anbieter sind z. B. Lineo (USA) und Sysgo (Deutschland).

Die folgende Abbildung faßt die Bedeutung von Linux als Applikationsplattform im Vergleich zu klassischen Unix-Systemen zusammen. Die Anzahl der unter Linux laufenden Applikationen übersteigt jede andere Unix-Plattform:

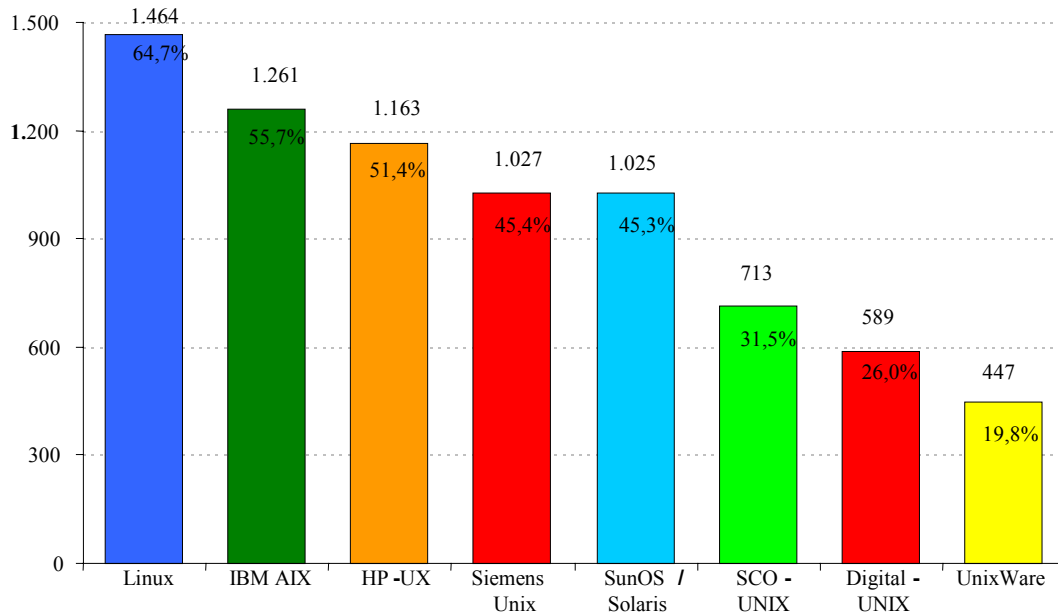


Abbildung 7: Anzahl der Programme für die jeweilige Plattform
 Quelle: nomina GmbH 2001, www.nomina.de

Die folgende Grafik zeigt das Wachstum der Anzahl an Programmen für Linux, gegliedert in verschiedene Applikationskategorien. Die Anzahl der Programme ist in den letzten drei Jahren stetig gestiegen, die größte Anzahl und das größte Wachstum verzeichnen dabei die branchenübergreifenden Applikationen:

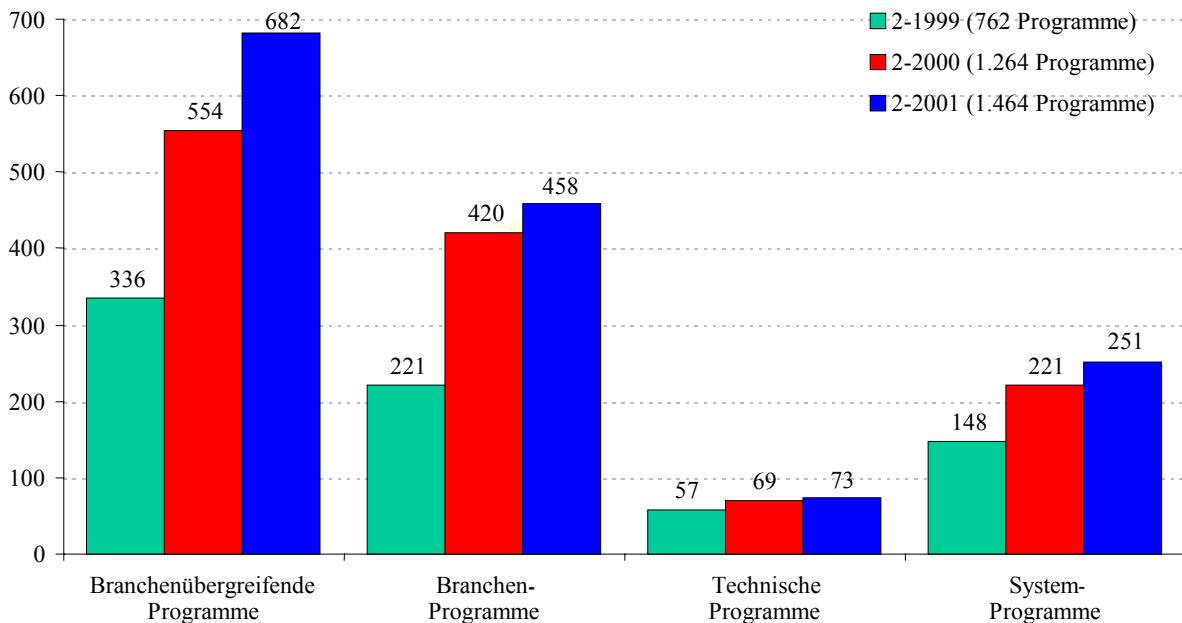


Abbildung 8: Linux-Programme nach Kategorien
 Quelle: nomina GmbH 2001, www.nomina.de

2.6.1.3 Der Markt für Appliances

Appliances sind Geräte, die aus einer Hardware/Software/Betriebssystem-Kombination bestehen. Je nach Einsatzzweck sind es Server- (z. B. Email-Server, Firewall-Server, Kommunikationsserver) oder Client-Geräte (Set top-Boxen, Thin Clients). Appliances integrieren mehrere Funktionen in einem Gerät und werden als einfach zu bedienende Komplettlösung vermarktet.

Technisch gesehen sind Appliances Standardgeräte, bestehend aus Standard (PC-)Hardware und aus Standard-Betriebssystemen (Windows, Linux). Meistens sind sie optisch besonders ansprechend gestaltet.

Appliances richten sich an Kunden, die einen konkreten Lösungsbedarf haben. Dieser Bedarf könnte zwar auch durch eine individuell konfigurierte Lösung gedeckt werden. Die Appliances stellen aber als vorinstallierte und vorkonfigurierte Variante eine bequeme und unter Umständen kostengünstigere Alternative zu dieser individuell konfigurierten Lösung dar.

Beispiele für Appliances im Server-Bereich (Zielgruppe kleine und mittlere Unternehmen):

- **Email-Appliances** sind Geräte, die den Austausch von Emails zwischen Unternehmen und dem Internet organisieren. Internetadressen von Mitarbeitern können eingerichtet und verwaltet werden.
- **Firewall-Appliances** sind Sicherheitssysteme, die Schutz vor Zugriffen aus dem Internet bieten und den Internetverkehr in einem Unternehmen überwachen.
- **Kommunikationsserver** sind Geräte, die mehrere Funktionen kombinieren und einen Allzweck-Kommunikationscharakter haben. Beispielsweise ermöglichen sie den Internetzugang, Emailaustausch, Proxy-Speicherung, Firewall-Schutz, Fax-Versand und –Empfang.

Beispiele für Appliances im Client-Bereich (Zielgruppe Unternehmen und Privatpersonen):

- **Thin Clients** sind einfache Arbeitsplatzrechnersysteme. Oft bestehen sie aus reduzierten Standard-PC-Komponenten (kein Diskettenlaufwerk, keine Festplatte). Sie ermöglichen den einfachen und bequemen Einsatz am Arbeitsplatz, ohne daß der Anwender mit komplexen PC-Eigenschaften überfordert wird. Sie sind zentral verwaltbar und bieten dadurch einen Kostenvorteil. Zielgruppe sind Unternehmen jeder Größe.
- **Set-top-Boxen** sind für den Privateinsatz konzipiert und ermöglichen die Verbindung eines Fernsehers oder anderer Heimelektronikgeräte mit digitalen Diensten. Ein typisches Beispiel ist die Anbindung eines Fernsehers an digitales Pay TV.

IDA/Unisys (2001) urteilen über den Appliance-Markt: „Our opinion is that from now on to 2004, the low-end dedicated Linux servers will grow up faster, to represent 30 to 40% of this market in units” (S. 24).

Rosenberg (2000) zitiert aus internen Microsoft-Papieren: „While the memo believes that Linux is a more immediate threat to Unix, it also perceives a near-term threat to NT-server revenues because hardware-manufacturers will be able to put out inexpensive special purpose servers using OSS” (S. 80).

Auch Analyst Weiss (2001) von der Gartner Group schätzt, daß Server Appliances der wichtigste Treiber für die Kommerzialisierung von Linux in den Jahren 1998 und 1999 war.

2.6.2 OSS-Dienstleistungsmarkt

Nachdem die Segmente des OSS-Produktmarktes untersucht wurden, folgt nun die Beschreibung des OSS-Dienstleistungsmarktes.

Hoch (1999) unterscheidet den Dienstleistungsmarkt in „Professional Services“ und „Enterprise Solutions“. Der Unterschied zwischen den beiden Segmenten liegt im Grad der „Produktisierung“ und der Verkaufszahl. Professional Services sind hochindividuelle Dienstleistungen, die nur einmal bzw. wenige Male in einer bestimmten Form für einen Kunden erbracht werden (Hoch 1999: 1-9 mal). Enterprise Solutions hingegen sind Dienstleistungen mit Produktcharakter. Sie sind zu einem gewissen Grad standardisiert und können für eine größere Anzahl von Kunden verkauft werden (Hoch 1999: zwischen 10 und 100.000 mal).

Genau so wie der Produktmarkt, kann auch der Dienstleistungsmarkt weiter segmentiert werden. Dienstleistungen können nach der technischen Ebene (z. B. Betriebssystem, betriebs-systemnahe Applikationen, Embedded-Systeme, Server, Clients), den Wertschöpfungs-abschnitten (z. B. Beratung, Studien, Implementation, Support) und der Zielgruppe (z. B. kleine Unternehmen, -große Unternehmen/Enterprise, Finanzunternehmen) unterschieden werden.

Der Dienstleistungsmarkt hat besondere Eigenschaften, die ihn vom Produktmarkt unterscheiden (angelehnt an Hoch 1999):

- **Hohe und konstante Grenzkosten:** Im Gegensatz zu den Produkt-Geschäftsmodellen fallen bei Dienstleistungen bei jedem Erstellungsvorgang nahezu identische variable Kosten an, da jedesmal Personen involviert sind. McKinsey beschreibt, daß diese Infrastrukturkosten bei Dienstleistungsunternehmen mehr als viermal höher sind als bei Software-Produktunternehmen (Hoch 1999). Dies ist der Grund, warum das „law of increasing returns“, also die zunehmende Profitabilität bei steigendem Produktabsatz, bei Dienstleistungsunternehmen nicht gilt.
- **Niedrige Fixkosten:** Die Kosten, um ein Dienstleistungsunternehmen zu gründen, sind niedriger als bei einem Produkt-Geschäftsmodell. Bei der Produktentwicklung muß eine Entwicklungszeit ohne Umsätze finanziert werden. Für ein

Dienstleistungsmodell reichen wenige Mitarbeiter und eine einfache Büroinfrastruktur aus.

- **Hoher Wettbewerb/Fragmentierung:** Niedrige Fixkosten machen den Dienstleistungsmarkt sehr wettbewerbsintensiv. Da Dienstleistungsunternehmen aber keine Möglichkeit haben, überproportional von Markterfolgen durch das „law of increasing returns“ zu profitieren, ist der Dienstleistungsmarkt geographisch fragmentiert (jedes Jahr werden laut Hoch in den USA mehr als 10.000 IT-Dienstleistungsunternehmen neu gegründet) und die Anbieter sind relativ klein.
- **Optimierung der Auslastungsquote:** Software-Unternehmen können Erfolg anhand der Marktanteile ihrer Produkte messen. Dienstleistungsunternehmen müssen Erfolg an der (internen) Größe des Auslastungsgrads messen. Das Personal verursacht Fixkosten, egal ob Dienstleistungsprojekte verkauft werden oder nicht. Nur wenn die Umsätze der verkauften Projekte höher sind als die Personalkosten und die sonstigen Kosten, werden Gewinne erzielt.
- **Verkauf vor Fertigstellung:** Software-Produkte werden aufgrund von definier- und vergleichbaren Eigenschaften verkauft. Dienstleistungsprojekte werden verkauft, ohne daß der Kunde sicher sein kann, die versprochene Leistung auch zu erhalten. Deshalb sind Vertrauen und eine enge Kundenbeziehung notwendig.

IDC beschreibt die Größe des Marktes für Linux-Dienstleistungen wie folgt (Monaco 2000):

	2000	2001	2002	2003	2004
Umsatzvolumen in Mio \$	28,3	56,6	105,0	177,0	285,0
Wachstum in %		97,9 %	87,5 %	68,6 %	61,0 %

Tabelle 3: Der Markt für Linux-Dienstleistungen

Gartner schätzt den Markt für Linux-Dienstleistungen bezogen auf das Server-Segment auf \$43 Millionen in 1999 und \$429,4 Millionen in 2004 (Gertz 2001).

2.6.3 OSS-Internetmarkt

Mit dem Internet-Boom in den letzten Jahren haben sich auch kommerziellen Anwendungen im Internet entwickelt²⁰. Einige wichtige Internet-Marktsegmente sind:

- E-Commerce (Online-Handel mit Produkten)
- Application Service Providing (Online-Software-Vermietung)
- Marktplätze (Auktionen, Matching-Funktion)
- Content (Portale, Communities, Content-Verkauf)
- Infrastruktur (Internet Service Provider, Telekommunikationsanbieter, Netzbetreiber)

²⁰ Für eine umfangreiche Auseinandersetzung mit Internet-Geschäftsmodellen vgl. Krüger (2001) und Zerdick (1999).

- Mobile Business

In vielen dieser Märkte spielt OSS eine Rolle, allerdings primär bei der technischen Umsetzung (z. B. OSS als Webserver-Software, Perl als dynamische Web-Programmiersprache, OSS-Sicherheitslösungen als Schutz der Infrastruktur). Diese technologische Rolle ist für die Marktanalyse allerdings weniger interessant, da OSS dort nur Mittel zum Zweck ist.

Eine weiteres Marktsegment sind Internetseiten zum Thema OSS. In dieser Arbeit werden auch diese deshalb nicht weiter betrachtet, weil erstens werbefinanzierte Geschäftsmodelle als nicht tragfähig gelten (Engdegard 2001) und zweitens das Geschäftsmodell ein übliches Content-Geschäftsmodell ist und sich nicht vom Geschäftsmodell anderer Internet-Webseiten unterscheidet. Ob auf einer Webseite Information über Open Source, Zeitungsmeldungen oder Gesundheitstips angeboten werden, spielt für die Erlösmechanismen keine Rolle.

Das einzige Internet-Marktsegment, in der OSS eine spezifische Rolle spielt, ist das *Marktplatzmodell*. Internet-Marktplätze betreiben ein „matching“ (Zusammenführen) von verschiedenen Interessengruppen.

Im Falle von OSS sind dies Entwickler, Anwender und Dienstleister rund um OSS. Zusätzlich können sich auch Werbeanbieter am Marktplatz beteiligen. Marktplätze unterstützen diese unterschiedlichen Interessengruppen und nehmen dabei eine neutrale Vermittlerfunktion ein. Im Falle von OSS nennt man die Betreiber solcher Marktplätze auch *Mediatoren*. Beispiele für Mediatoren im OSS-Bereich sind Berlios (www.berlios.de) und Sourceforge (www.sourceforge.net).

Neben OSS-Marktplätzen gibt es auch Marktplätze für kommerzielle/proprietäre Software. Ein deutscher Anbieter ist das Unternehmen nomina, das einen Marktplatz unter www.software-marktplatz.de betreibt. Der Marktplatz richtet sich an die Anbieter kommerzieller/proprietärer Software und an potentielle Kunden. Der Marktplatz umfaßt 7.000 Produkthersteller und über 10.000 Programme. Weitere Anbieter sind die Marktplätze www.internetcomponent.com und www.componentsource.com (8.500 Softwarekomponenten, Kunden in 110 Ländern). Dort werden Programme bzw. Teilprogramme (Komponenten) zum Thema Internet, Internet-Programmierung, Internet-Server und ähnliches angeboten.

Der Vorteil eines Marktplatzes liegt darin, daß die Kontaktkosten der Teilnehmer reduziert werden. Die Anwender sparen Zeit bei der Recherche nach einer bestimmten Software. Der Entwickler hat den Vorteil, ohne Streuverluste potentielle Kunden zu erreichen. Der Vorteil für Dienstleister liegt wiederum darin, das Angebot möglichst vielen potentiellen OSS-Nutzern zu präsentieren.

Die folgende Abbildung visualisiert das Zusammenspiel der Teilnehmer an einem Marktplatz:

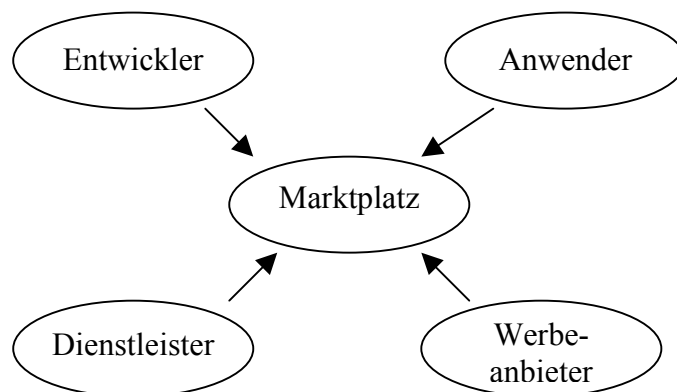


Abbildung 9: Zusammenspiel von Marktplatz-Teilnehmern

Abschließend hier noch einmal die wichtigsten Inhalte dieses Kapitels:

OSS ist Software mit offenem, modifizierbarem Quellcode. Sie kann dabei kommerziell oder nicht-kommerziell sein. OSS ist ein relativ altes Konzept und geht auf die Anfänge der Computerindustrie zurück. Die Entwicklung und Nutzung von OSS ist ein dynamischer Prozeß. Dabei kann eine Person gleichzeitig z. B. Nutzer, Entwickler und Tester sein.

Es gibt wesentliche Unterschiede zwischen OSS und proprietärer Software: OSS verändert die klassische Software-Wertschöpfungskette. Der Nutzer hat einen höheren Freiheitsgrad und das Innovationsmonopol der Softwarehersteller wird aufgebrochen. Die Dienstleistung erhält einen höheren Stellenwert. Der OSS-Entwicklungsprozeß ist dezentralisiert, eigenmotiviert, offen und dynamisch. Es gibt eine Kultur permanenter Fehlersuche und Verbesserung. Dadurch entsteht (oftmals) hochwertigere Software. OSS liegen offene Lizenzen zugrunde. Im Gegensatz zu proprietärer Software existieren keine Schutzmechanismen, die Software wird zum Gemeingut.

OSS kann in den allgemeinen IT-Markt eingeordnet werden: OSS ist dabei im Produkt-, Dienstleistungs- und Internet-Marktsegment vertreten. Der Produktbereich unterteilt sich dabei in Betriebssystem (z. B. Linux-Distribution), Applikationen (z. B. ERP-Software) und Appliances (z. B. Email-Server) für Server, Client und Embedded-Systeme. Im Dienstleistungsmarkt wird professionelle Unterstützung beim Einsatz von OSS angeboten. Auf einem OSS-Marktplatz wird ein Matching von Entwicklern, Nutzern, Dienstleistern und Werbeanbietern betrieben, dabei werden die Kontaktkosten gesenkt.

3 Nachfragerseite: Kommerzieller Einsatz von OSS

Die ersten Kapitel haben sich mit den grundsätzlichen Eigenschaften von OSS und den relevanten Märkten beschäftigt. Das folgende Kapitel beschreibt den Einsatz von OSS auf Nachfragerseite, also den Unternehmen und Institutionen, die sich für den Einsatz OSS als IT-Technologie entscheiden. Dabei wird beschrieben, welche Rolle OSS im kommerziellen Einsatz spielt und welche Vor- und Nachteile für und gegen diesen Einsatz sprechen.

Rosenberg (2000) beschreibt eine Studie des Marktforschungsunternehmens Datapro von 1998, in der folgende Haupteinsatzgebiete von Linux beschrieben werden:

Einsatzgebiet	Anteil
Webserver	33,3
Applikationsserver	9,9
Wissenschaftliche/technische Anwendung	14,8
Vernetzte Workstation	9,9
Desktop	6,2
Enterprise	9,9
Sonstige	16

Tabelle 4: Linux-Einsatzgebiete
Quelle: Rosenberg (2000)/Datapro (1998)

Über diese Einsatzgebiete hinaus sind noch weitere zu nennen, für die der Einsatz von OSS empfohlen wird (vgl. BMWi/innominate 2000):

- **Intranetbereich:** Ein Intranet dient zur internen Vernetzung in einem Unternehmen. OSS eignet sich besonders gut aufgrund der standardisierten Protokolle und der Kompatibilität mit den Internet-Technologien.
- **Virtual Private Networks: (VPNs):** VPNs verbindet Niederlassungen eines Unternehmens abgesichert über das Internet. OSS eignet sich besonders aufgrund der erhöhten Sicherheit.
- **Internetbereich:** Wenn ein Unternehmensnetzwerk an das Internet angebunden werden soll (z. B. Email, Webserver) eignet sich OSS gut wegen der offenen Internetstandards und des großen Softwareangebots.
- **Heterogene Netze:** Bei Verwendung unterschiedlicher Computersysteme und Netzwerke eignet sich OSS sehr gut aufgrund der vielfältigen Protokolle und offenen Schnittstellen.
- **Datenarchivierung:** Wenn viele Dokumente langfristig archiviert werden sollen, eignet sich OSS besonders gut aufgrund der vielen herstellerunabhängig unterstützten Datenformate.

Auch die vollständige Umstellung auf Linux/OSS ist möglich: Klever (1999) beschreibt den Einsatz von Linux in der „tageszeitung“. Dort wird Linux seit 1995 eingesetzt. Begonnen wurde mit Linux als ISDN-Router, Webserver, Mail- und Newsserver. Heute laufen alle EDV-Systeme unter Linux (140 Arbeitsplätze, darunter Entwicklung, Administration, Redaktion und Archiv).

3.1 Vorteile des kommerziellen Einsatzes von OSS

Im Vergleich zu proprietärer Software ist OSS sicherer, schneller, offener, flexibler, stabiler bei geringeren Kosten. Im folgenden werden diese Thesen näher begründet.

3.1.1 Sicherheit

Im Zuge der fortschreitenden Vernetzung wird die Sicherheit von IT-Systemen immer wichtiger. Zur IT-Sicherheit gehören sowohl technische als auch nicht-technische Aspekte. Die nichttechnischen Aspekte sind dabei nicht zu vernachlässigen, denn oftmals sind IT-Systeme zwar sicher ausgelegt, werden aber durch den „Faktor Mensch“ unterlaufen (z. B. Notieren von Passwörtern oder Sabotage durch Mitarbeiter). Im folgenden werden ausschließlich die (software)technischen Eigenschaften der Sicherheit behandelt, auf eine ausführlichere Behandlung aller Aspekte von IT-Sicherheit wird verzichtet.

Sicherheitssysteme haben den Zweck, Rechnersysteme und Netzwerke vor unerlaubtem Zugriff zu schützen. Um diese Aufgabe zu erfüllen, müssen Securitysysteme ausgereift, aktuell und fehlerfrei sein. Diese Eigenschaften gelten prinzipiell für alle Security-Systeme auf allen Plattformen. Es ist also grundsätzlich möglich, ein sicheres oder ein unsicheres System sowohl auf Linux- als auch z. B. auf Unix-Systemen zu implementieren²¹.

In die Diskussion von Security tritt aber im Kontext von OSS noch eine ganz andere Dimension hinzu: Bei OSS geht es nicht nur um die Qualität einer konkreten Sicherheitsanwendung, sondern um prinzipbedingte Vorteile. Quade (2001) führt folgende Argumentation, warum Open Source *prinzipbedingt*, unabhängig von der Güte einer konkreten Anwendung, vorteilhaft ist:

- OSS ist sicherer aufgrund des Qualitätsmanagements rund um den Entwicklungsprozeß. Bei der öffentlichen Entwicklung von OSS wird ein „Security Code Auditing“ durch Dritte durchgeführt. Nicht nur ein Autor oder ein lokales Autorenteam ist an der Erstellung einer Software beteiligt, sondern durch globale Teamentwicklung werden Fehler vermieden. Dieses Prinzip ist nur möglich, weil die Software im Quellcode vorliegt.
- OSS ist sicherer aufgrund von Aktivierungsmöglichkeiten zusätzlicher Sicherheitsmechanismen. Weil OSS offen ist, können Dritte eine Software verändern und verbessern. Im Security-Bereich gibt es beispielsweise Möglichkeiten, bestehende

²¹ Die Verwendung von Windows als Plattform für Security-Systeme wird von Sicherheitsexperten dagegen eher kritisch betrachtet.

Softwarepakete anderer Autoren durch grundlegende Veränderungen an deren Quelltext sicherer zu machen.²²

- OSS ist sicherer, weil es auf bestimmte Anwendungsgebiete optimiert werden kann. So existieren eine Reihe von speziellen Linux- und BSD-Versionen, die sich für den Einsatz in Hochsicherheitssystemen ganz besonders auszeichnen.
- OSS ist sicherer aufgrund der Offenlegung des Codes. Die Architektur, die Funktion und die Fehlerfreiheit einer OSS können auditiert werden, d. h. aufgrund des Mehraugenprinzips ist eine Kontrolle durch Dritte möglich. Es ist dabei nicht unbedingt notwendig, daß der Endkunde selber den Quellcode prüft, denn der *Prozeß* ist sicherer.

Köhntopp (2000) ergänzt diese Argumente noch durch folgende Aspekte:

- OSS ist sicherer, weil aufgrund des offenen Quellcodes dort „Trojaner“²³ schwieriger zu verbergen sind.
- OSS ist sicherer, weil im Falle einer Sicherheitslücke bei Open Source schneller durch Gegenmaßnahmen (Patches oder anderes) reagiert werden kann und reagiert wird.
- OSS ist sicherer, weil keine Abhängigkeit von einem einzelnen Hersteller besteht.

Kritiker von OSS argumentieren, daß OSS gerade aufgrund der Offenheit der Quellen besonders angreifbar wäre. Der Angreifer kenne die Funktionsweise, geschlossene Software wäre geschützter aufgrund der Verborgenheit der Funktionen. Köhntopp (2000) widerlegt diese Argumentation (S.8):

„Die Offenlegung des Quellcodes und der Designkriterien wird für kryptographische Algorithmen seit langem als eine notwendige Bedingung für eine Sicherheitsuntersuchung angesehen. [...] Daß ‚Security by Obscurity‘, also Sicherheit durch Verschleierung, ein höchst unzuverlässiges und fragwürdiges Prinzip ist, zeigen auch heute noch immer wieder Beispiele, in denen sich die Geheimhaltung des Quellcodes doch nicht gewährleisten ließ oder sicherheitsrelevante Fehler durch Zufall, gezielte Angriffe auf das ausführbare Programm oder Reverse-Engineering gefunden wurden [...].“

Auch Whitlock (2001) widerlegt solche Argumente dadurch, daß Sicherheitssoftware gerade dann sicherer sei, wenn der Mechanismus offengelegt wäre. Historische Beispiele aus der Militärwelt würden zeigen, daß Verschlüsselung sicherer ist, je offener die Verschlüsselungsverfahren gemacht werden.

Köhntopp (2000) zeigt zwar auch einige Grenzen der OSS-Sicherheit auf und führt aus, daß der Open Source Entwicklungsprozeß kein „Allheilmittel“ ist. Er kommt aber wie die meisten

²² Technisch erfolgt dies durch einen „Patch“ (die Veränderung fremder Quellcodes durch eigenen Quellcode), der sich dabei auf das gesamte Betriebssystem und alle beteiligten Komponenten auswirken kann.

²³ Trojaner sind bösartige Funktionen, die in Anwendungen eingeschleust werden.

Sicherheitsexperten insgesamt zu dem Schluß, daß Open Source die Sicherheit durchaus erhöht:

„Die Offenlegung des Quellcodes ist ein wichtiges Hilfsmittel, um Sicherheit und Vertrauenswürdigkeit von Systemen zu erreichen. Das offene und kooperative Software-Entwicklungsmodell von Open-Source-Projekten hat gezeigt, daß man auf diese Weise robuste und verlässliche Programme erstellen kann. Allerdings ist dies kein Automatismus, sondern erfordert Sorgfalt im gesamten Entwicklungsprozess sowie bei der Evaluierung durch Experten.“ (Köhntopp 2000, S. 11).

Insgesamt kann also festgehalten werden, daß OSS Software in der Regel sicherer ist, aber die Offenlegung des Quellcodes dabei nicht das einzige Kriterium ist.

3.1.2 Performance

OSS ist nach Ansicht vieler Analysten (im Server-Bereich) schneller als proprietäre Produkte. Kaven (2001) hat einen Vergleichstest zwischen Linux und Windows 2000 auf drei Hardwareplattformen durchgeführt. Getestet wurde der Datendurchsatz bei der Anwendung als Dateiserver (nach Rosenberg (2000) die am zweithäufigsten genutzte Anwendung im kommerziellen Bereich).

Die Grafik zeigt, daß Samba (das OSS-Projekt, um Windows-Clients an einen Dateiserver anzubinden) Windows-Server im Datendurchsatz schlägt. Ähnlich positive Ergebnisse wurden auch für den Parameter „Reaktionszeit“ erzielt.

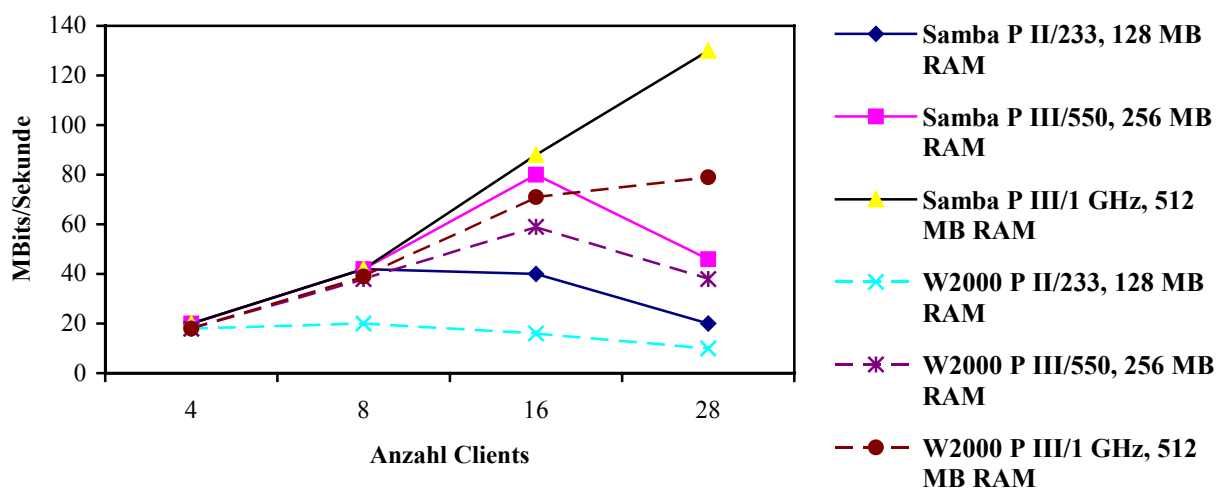


Abbildung 10: Datendurchsatz von Linux/Samba im Vergleich mit Windows 2000
Quelle: Kaven 2001

Der oben durchgeführte Performance-Test bezieht sich auf einen Vergleich zwischen Linux und Samba auf der einen und Windows 2000 mit dem integrierten Dateiserver auf der anderen Seite auf verschiedenen Rechnersystemen.

Um allgemeingültige Aussagen zur Performance von OSS gegenüber proprietärer Software zu machen, müßte man alle gängigen Betriebssysteme (Linux, BSD, Unix, Windows 2000) und alle gängigen Anwendungen (Dateidienste, Webserver, Email etc.) in einer Matrixform vergleichen. Diese Vorgehensweise kann im Rahmen dieser Arbeit nicht geleistet werden. Es sei an dieser Stelle lediglich darauf hingewiesen, daß in der Literatur beschriebene Performancevergleiche das Bild ergeben, daß OSS gegenüber Windows-Systemen in der Regel Leistungsvorteile bietet (Weiss 2000, Diedrich 2000, IDC 2000, IDA/Unisys 2001, Wheeler 2001). Diese variieren je nach Anwendung und Hardwareausstattung von vergleichbarer Performance bis hin zu deutlichen Performancevorteilen.

Weiss (2000) betont allerdings auch die schnell fortschreitende Entwicklung der Windows-Systeme, so daß Windows aufholen wird (S.4):

„Linux, while making technical progress, will still face stiff challenges at the high end as Unix and Windows 2000 continue to advance. By 2003, Linux will achieve in most categories an acceptable rating, but few excellent ratings. To achieve the high-end functionality of Unix and Windows 2000, Linux will need more support from server and software vendors because the OSS community will not be organized or experienced enough for the fast pace of server advances.”

Der Performancevergleich zwischen Unix und Open Source/Linux-Systemen fällt in der Regel deutlich zugunsten von Unix aus. Unix wird allgemein als besser skalierbar und für High-End-Anwendungen geeignet eingeschätzt (Weiss 2000). Den Leistungsvorteil erkaufte man aber auch durch höhere Kosten bei Lizenzen und Hardware.

3.1.3 Offenheit/Flexibilität

Hersteller von proprietärer Software bedienen eine bestimmte Zielgruppe. Für diese Zielgruppe erstellen sie ein Produkt, das eine gewisse Individualität (für eine bestimmte Zielgruppe entwickelt), aber gleichzeitig auch einen gewissen Grad an Standardisierung hat. Je individueller ein Produkt ist, desto teurer ist es (weil es nur an eine begrenzte Anzahl an Kunden verkauft werden kann). Je standardisierter ein Produkt ist, desto preiswerter ist es (da die Software öfter verkauft werden kann). Der Kunde muß folglich eine Abwägung zwischen den beiden Parametern Individualität und Preis treffen.

Des Weiteren muß Software in der Regel konfiguriert, also auf die konkreten Einsatzzwecke beim Kunden angepaßt werden. Die Konfigurierbarkeit von proprietärer Software ist dabei auf den Rahmen beschränkt, den der Hersteller vorgibt. Entscheidend ist hierbei die Mächtigkeit der Schnittstellen und die Anpaßbarkeit (z. B. die Verbindung zu anderen Softwaresystemen oder das Ein- und Ausgabeformat von Daten).

Manchmal ist der vom Hersteller vorgegebene Anpassungsrahmen nicht ausreichend. So kann z. B. das Format eines alten Buchhaltungssystems nicht lesbar sein. Oder ein Netzwerk-

protokoll ist für eine Software nicht geeignet. Ein besonders unangenehmer Fall liegt dann vor, wenn ein Kunde ein Update einer Software einspielen muß (z. B. aus Sicherheitsgründen), dort aber eine bestimmte Funktion in einer Softwareversion nicht *mehr* vorhanden ist. Oder ein Softwareanbieter wird gekauft oder insolvent und deswegen wird die Pflege der Software eingestellt. In all diesen Fällen ist der Kunde auf die vom Hersteller vorgesehenen Möglichkeiten beschränkt. Eine Anpassung oder Erweiterung darüber hinaus ist nicht möglich.

Anders bei OSS: Stellt der Anwender fest, daß bestimmte Aspekte der Software den Anforderungen nicht (mehr) genügen, kann er die Software beliebig erweitern (lassen).

In einer Studie von IDA/Unisys über den Einsatz von OSS in der europäischen Kommission wurden Systemadministratoren über Gründe für den Einsatz von OSS befragt. Das Argument der „Interoperabilität“, ein technischer Begriff für die Flexibilität und Offenheit von OSS, wurde dabei mit 80% als wichtigstes Argument benannt:

„With more than 80% attributed to these criteria, the interoperability and respect of standards of the OSS are the main reasons for using OSS in the Public Sector. The main strength of OSS is that it is ‘constructed for interoperability’ and ‘closely associated to open standards’. OSS is considered to better respect standards because no proprietary standards are used to ‘protect’ the vendor captive market and it is in the interest of everybody to achieve the best interoperability. The permanent research for public common standards makes OSS more convenient for long-term interoperability.”
(IDA/Unisys 2000, S. 15)

Dieser Aspekt stellt einen wesentlichen Vorteil von OSS gegenüber proprietärer Software dar. Allerdings muß erwähnt werden, daß eine Veränderung einer installierten Software, sei sie OSS oder proprietär, zwar ein aktuelles Problem löst, aber auch ein zukünftiges Problem schafft: Jede Abweichung vom Standardsystem erzwingt bei der Umstellung auf neue Softwareversionen eine erneute Anpassung und damit erneuten Zeit- und Kostenaufwand. Dies gilt auch für individuelle Abweichungen von Standard-OSS-Installationen, wenn sich zwischen zwei Softwareversionen wesentliche Teile der Software verändern (z. B. Datenstrukturen, Format, Schnittstellen).²⁴

3.1.4 Stabilität

Ein weiteres wichtiges Argument für den Einsatz von OSS ist die hohe Stabilität. Diese ist ein kritischer Faktor für den Betrieb von IT-Systemen, denn mangelnde Stabilität äußert sich in Abstürzen, Ausfallzeiten und erhöhtem Wartungsaufwand.

IDA/Unisys (2000) urteilt: „OSS is generally of better software quality and a higher reliability is obtained in many cases” (S. 16). Als Argumente werden angeführt, daß OSS in der Regel

²⁴ Allerdings können allgemeine Anpassungen/Änderungen, wenn sie auch für andere Anwender nutzbar sind, in das OSS-Projekt einfließen. Der Urheber der Änderungen wird zum „mit-Autor“ der Software. Dadurch fallen aber Zusatzkosten für das Unternehmen an und die Innovationen werden auch für Wettbewerber nutzbar.

besser getestet und kommentiert wird und der Entwicklungsprozeß im Gegensatz zu kommerziellen Produkten ohne Zeitdruck erfolgt.

In einer Untersuchung des Marktforschungsunternehmens TNS Emnid wurden Linux-Nutzer nach dem Grund des tatsächlichen bzw. geplanten Einsatzes von Linux befragt (Edlbauer 2001). 46% der Benutzer gaben die hohe Stabilität von Linux als einen wichtigen Grund an. Die Umfrage richtete sich allerdings vornehmlich an Heimbenutzer.

Ein weiteres Beispiel für die Stabilität von OSS liefert das Technikmagazin Zdnet (zitiert in Wheeler 2001). Dieses hat einen zehnmonatigen Vergleichstest zwischen Windows NT 4.0 und Linux auf identischer Hardware durchgeführt. Dabei wurde festgestellt, daß Windows NT im Schnitt alle sechs Wochen abgestürzt ist und die Wiederanlaufzeit im Schnitt 30 Minuten betrug. Das Linux-System ist im gleichen Zeitraum nicht ein einziges Mal abgestürzt.

3.1.5 Preisvorteil

OSS ist in der Regel kostenfrei erhältlich. Für den Nutzer entfällt damit der Kaufpreis für Softwarelizenzen für Betriebssystem und Applikationen.

Allerdings ist der Kaufpreis als Maßstab im kommerziellen Bereich nicht ausreichend. Eine Software muß nicht nur erworben, sondern auch installiert, konfiguriert, geschult und gewartet werden. Das Marktforschungsunternehmen IDC definiert den Begriff „Total Cost of Ownership“ (TCO) als umfassendes Kostenkonzept wie folgt (Gillen 2001, gesponsort vom Linux-Distributor Red Hat):

Das TCO-Modell kann unterteilt werden in drei Phasen: *Beschaffung*, *Betrieb* und *Außerstandsetzung* (procurement, use, disposition). Dabei enthält Beschaffung neben den Kosten für Hard- und Software auch die Kosten für die Finanzierungsplanung (Leasing, Kauf), die Erhebung der Anforderungen und die Lieferkosten. Der Betrieb beinhaltet Inbetriebnahme, Schulung, technischen Support und Disposition. Die Außerstandsetzung beinhaltet den physischen Abtransport der Hardware und die Überspielung von Daten und Applikationen auf Nachfolgesysteme.

Weitere Kosten für den Betrieb von Rechnersystemen wie z. B. externe Kommunikationskosten, Medien, Strom und Platzbedarf werden im folgenden ausgeklammert, da sich verschiedene IT-Systeme in diesen Parametern nicht wesentlich unterscheiden.

Die TCO-Untersuchung von IDC/Red Hat untersucht die Kosten von Linux auf einer PC-Plattform im Vergleich mit Unix-Systemen auf RISC-Prozessoren (Unix-Workstations). Dabei wurden zwei Einsatzzwecke getrennt untersucht. Erstens die Verwendung als Internet/Intranet/Extranet-Plattform und zweitens als „Collaborative“-Plattform (u.a. Nutzung gemeinsamer Verzeichnisse, Email, Kalender).

IDC kommt zu dem Ergebnis, daß Linux als Plattform deutliches Einsparpotential gegenüber Unix-Systemen ermöglicht:

„Linux on the Intel platform has emerged as a viable alternative to RISC/Unix for enterprise computing. For enterprises with the right

mix of requirements and skill, Linux offers tremendous potential to lower costs associated with supporting application workloads.” (IDA/Unisys 2000, S.1)

Die Einsparung bezieht sich dabei vor allem auf die Kostenbereiche Hard-/Software („harte Kosten“) und den Bereich Personalaufwand („weiche Kosten“). Die ausgewiesenen Kennzahlen beziehen sich dabei auf die Kosten, die pro Jahr und Anwender auf der Basis 1.000 unterstützter Anwender anfallen. Im Internet/Intranet/Extranet-Bereich unterbietet Linux die Unix-Systeme mit 377 US-Dollar gegenüber 684 US-Dollar. Noch erheblicher fällt der Unterschied im Collaborative Computing aus. Hier schlägt laut IDC ein Linux-User mit 255 Dollar zu Buche, während unter Unix pro Anwender 1.407 Dollar anfallen.

Begründet wird das Einsparpotential im Bereich der harten Kosten mit niedrigeren Preisen für die Anschaffung der Software und der niedrigeren Hardwarekosten. Die Einsparungen im Bereich Personal resultieren vor allem aus der niedrigeren Komplexität der unter Linux verwendeten Applikationen.

Ein Vergleich zwischen Linux und Windows-Systemen in ähnlich detaillierter Form ist nicht bekannt. Da die Hardwareanforderungen für Windows-Systeme im Vergleich zu Unix-Systemen weniger hoch sind und gerade die Bedienbarkeit von Windows in der Regel einen großen Vorteil darstellt, wird ein Vergleich zwischen Linux und Windows vermutlich weniger deutlich zugunsten von Linux ausfallen.

Diese These wird indirekt bestätigt in der Diskussion über die Verwendung von Linux im Bundestag:

„In der Nutzwert- und Wirtschaftlichkeitsbetrachtung [über den Einsatz von Linux als Alternative zu Windows-Servern im Bundestag durch die Unternehmensberatung Infora, RL] taucht nur noch die Alternative ‚Windows in allen Bereichen‘ sowie die auf dem Server-Sektor um Linux angereicherte Variante auf. Bei den Auswirkungen auf den Haushalt weichen diese beiden Planspiele laut Infora ‚nicht gravierend voneinander ab.‘“ (Krempf 2001, S.1).

Die Umstellungskosten von Windows auf Linux werden dabei mit 9,5 Millionen Euro veranschlagt.

Der Bundesrechnungshof kommt zu dem Schluß, daß der Einsatz von Linux gegenüber Windows in der gesamten Bundesverwaltung ein Einsparpotential von „bis zu 200 Millionen Mark“ bringt (Krempf 2002). Der Bayerische Oberste Rechnungshof beziffert das Einsparpotential bei der Verwendung von Linux gegenüber Microsoft-Produkten auf eine Summe von „bis zu 18 Millionen Euro“ für die bayerischen Ämter (Krempf 2002, S.1).

Ein Vergleich von anderen Open Source-Betriebssystemen wie z. B. FreeBSD oder OpenBSD mit Windows- oder Unix-Systemen ist nicht bekannt. Der Grund hierfür dürfte in der geringeren Verbreitung dieser Systeme im Vergleich zu Linux liegen.

3.2 Grenzen des kommerziellen Einsatzes von OSS

Im vorigen Abschnitt wurden Gründe *für* den Einsatz von OSS im kommerziellen Bereich genannt. Der folgende Abschnitt beschreibt Argumente *gegen* den Einsatz von OSS.

OSS hat im Vergleich zu proprietärer Software ein geringeres Angebot an Applikationen, unzureichende Fähigkeiten im Desktop-Bereich, bietet aus Entscheidersicht unter Umständen mangelnden Investitionsschutz und hat Mängel in der Administrierbarkeit. Diese Thesen werden im folgenden näher untersucht.

3.2.1 Mangel an Applikationen

Entscheidungen für Betriebssysteme werden in den seltensten Fällen als Selbstzweck getroffen. In der Regel ist die Entscheidung für ein Betriebssystem sekundär und leitet sich aus der Entscheidung für bestimmte Applikationen ab (z. B. Buchhaltungssoftware, Internetserver, Office-Produkt).

Das Angebot von Applikationen ist deshalb ein kritischer Faktor für den Erfolg eines Betriebssystems. Wie in Kapitel 2.6.1.2 beschrieben, hat OSS und speziell Linux durchaus ein reichhaltiges Angebot an Applikationen. Allerdings zeichnen sich diese durch bestimmte Charakteristika aus:

- **Applikationen haben Infrastrukturcharakter:** Typische Applikationen für den Einsatz von Linux im Netzwerk sind z. B. Firewalls, Webserver, Newsserver, Emailserver etc.²⁵ Diese sind eher Basislösungen, als daß sie eine spezielle Lösung im Sinne einer Unternehmens- oder Verwaltungsanwendung darstellen.
- **Mangel an Unternehmens- und Verwaltungsapplikationen:** Es gibt zwar eine Reihe von „prominenten“ Linux-Unterstützern aus diesem Bereich (z. B. SAP oder Oracle), dennoch gibt es eine Reihe von Aufgaben, die unter Linux bisher nur am Rande adressiert wurden. Beispiele hierfür sind z. B. Management Information Systems, CRM-Systeme, Projektmanagement-, Collaborative- und Workflowsysteme.
- **Mangel an Clientapplikationen:** Die wichtigsten OSS-Applikationen sind Serverapplikationen. Echte „Killerapplikationen“ im Desktop-Bereich wie z. B. eine Portierung von Microsoft Office oder komfortable Produktivitäts-, Groupware-, Kalender- und Workflow-Applikationen aus dem Windows-Bereich fehlen. Alternative Produkte erzeugen Probleme mit dem Datenaustausch (Filter- und Konvertierungsproblematik).

Rosenberg (2000) nennt folgende Client-Applikationen als die am dringendsten benötigten:

- Personal Information Manager (Terminplanung kombiniert mit Kontaktverwaltung)

²⁵ S. Tabelle 4: Linux-Einsatzgebiete, S. 28

- Customer Relationship Software (Software zur Kundenpflege und Vertriebsunterstützung)
- Decision Support Software (Software zur Entscheidungsunterstützung).

3.2.2 OSS auf dem Desktop

OSS hat seine Ursprünge in einem technischen Milieu. Bei der Entwicklung von OSS werden technische Maßstäbe angelegt, weniger anwenderorientierte Kriterien wie z. B. Benutzerfreundlichkeit und Bedienbarkeit.

Deshalb gibt es trotz vieler Verbesserungen in den letzten Jahren immer noch Probleme mit Inkompatibilitäten und mangelnder Bedienbarkeit. Für technisch versierte Personen mag dies selten ein Problem darstellen; für an Windows gewohnte Benutzer kann es eine große Hürde bedeuten.

Trotz der Tatsache, daß in den letzten Jahren einige Initiativen wie z. B. KDE und Gnome für die Verbesserung der Qualität von OSS (insbesondere Linux) auf dem Desktop gestartet wurden, stagniert der Marktanteil von Linux deshalb bei wenigen Prozentpunkten (IDC 2000).

Weitere Gründe für die mangelnde Verbreitung von OSS auf dem Desktop sind die üblicherweise mit Windows vorinstallierten PC-Systeme, die bevorzugte Entwicklung von Windows-Treibern durch Hardwarehersteller, das größere Angebot von Applikationen für Windows, die bessere und einfachere Bedienbarkeit von Windows und das Fehlen von de-facto-Standardapplikationen wie z. B. Microsoft Office.

Deshalb zeigen nur wenige Unternehmen oder Behörden Bereitschaft, Linux auf dem Desktop einzuführen. Auch in der Diskussion über den Einsatz von Linux in der deutschen Verwaltung und im Bundestag zeigt sich die Tendenz zu einer gemischten Umgebung von Linux auf dem Server und Windows auf den Arbeitsplätzen:

„Die Prüfteams geben vor allem als Client-Betriebssystem Windows XP gegenüber Linux [...] aus 'technischer Sicht' den Vorzug. Beim Test gängiger Bürotätigkeiten zeigten sich mit dem Open-Source-Produkt StarOffice 6 zwar keine schwerwiegenden Mängel; die Kompatibilität mit unter Windows erstellten Dateien sei größtenteils gewährleistet. Die IT-Experten der Bundestagsverwaltung bemängeln aber, daß das anscheinend bei den Abgeordneten beliebte Arbeiten mit Copy & Paste 'nicht vollständig unterstützt' werde und die Zwischenablage-Funktion nicht durchgängig zu nutzen sei.“ (Krempel 2002, S.1)

Diedrich (2001) führt als weiteren Punkt für das Scheitern von Linux auf dem Desktop die Insolvenz von Linux-Desktop-Unternehmen an. So mußte das mit \$11 Mio. finanzierte Unternehmen Eazel (unter anderem von ehemaligen Apple-Gründern unterstützt), seine Tätigkeit 2001 aufgrund mangelnder Akzeptanz einstellen (Hall 2001). Das Marktforschungsunternehmen IDC wird von Galli zitiert: „...while the server market remains compelling for Linux, I see no promising future for Linux on the desktop through 2005.“ (2001, S. 1).

Auch der Grafikspezialist Corel scheiterte mit dem Versuch, eine eigene Distribution für Linux-Desktops zu entwickeln und trennte sich mit Verlusten von dem Linux-Entwicklungsbereich (Galli 2001).

3.2.3 Investitionsschutz

Der Open Source-Entwicklungsprozeß zeichnet sich durch hohe Parallelität und Komplexität aus. Der Kernel und die einzelnen Softwarekomponenten werden mit hoher Frequenz upgedatet. Auf der einen Seite stellt dies einen Vorteil dar (Qualität, Sicherheit), auf der anderen Seite kann dies für Unternehmen auch Unsicherheit und höheren Kostenaufwand bedeuten.

Das Fehlen einer Koordinationsstelle im Sinne des „bottom-up“-Prozesses der OSS-Entwicklung bedeutet, daß für die Unternehmen kein zentraler Ansprechpartner „auf Augenhöhe“ existiert. Auch wenn zunehmend Unternehmen wie IBM, HP, Compaq und viele kleine und mittlere Unternehmen OSS-Kompetenz aufbauen, kann für Entscheider das Gefühl eines „Abenteuers“ verbleiben.

Dieser Vertrauensaspekt äußert sich auch in Haftungsfragen. Da es keinen Verkäufer und Lizenzgeber gibt, kann keine definierte Instanz im Schadensfall angesprochen werden. Dieses Argument verliert allerdings an Bedeutung, je mehr garantierte Serviceleistungen in Form von Wartungsverträgen und Service Level Agreements (s. Geschäftsmodell OSS-Dienstleister) angeboten werden. Außerdem wird auch bei proprietärer Software regelmäßig jede Haftung durch die Lizenzbestimmungen ausgeschlossen.²⁶

3.2.4 Administrierbarkeit

OSS-Betriebssysteme und insbesondere Linux sind aufgrund der Abstammung von Unix sehr mächtige Werkzeuge. Das Bedienungskonzept und die Mächtigkeit der zahllosen Unix-Administrations- und Verwaltungstools sind für ausgebildete und erfahrene Administratoren sehr hilfreich.

Auf der anderen Seite sind die hinter Unix, Linux und den meisten anderen OSS-Betriebssystemen liegenden Konzepte jahrzehntealt und haben sich in dieser Zeit wenig verändert.²⁷ Die Bedienung ist für ungeübte und ungeschulte Benutzer schwierig. So hat auch eine repräsentative Befragung des Marktforschungsunternehmens Emnid ergeben (Emnid 2001), daß (private) Linux-Benutzer die „umständliche Installation“ und die „umständliche Bedienung“ mit 18% bzw. 16% als die meistgenannten Nachteile von Linux einschätzen.

²⁶ Für eine weitergehende Auseinandersetzung mit Haftungsfragen bei OSS und proprietärer Software vgl. BMWi/innominate 2001.

²⁷ Viele der Linux-Innovationen im oberen Server-Bereich orientieren sich an klassischen Unix-Konzepten.

Viele der Innovationen von Microsoft dagegen kommen aus dem Bereich einfache Bedienung und sinnvolle Reduktion der Komplexität. Das sind wichtige Gründe für die Entwicklung von Windows zum de-facto-Standard auf dem Desktop.

Nadeau (1999) illustriert diesen Sachverhalt wie folgt:

„The difference here is, in every release cycle Microsoft always listens to its *most ignorant customers*. This is the key to dumbing down each release cycle of software for further assaulting the non-PC population. Linux and OS/2 developers, on the other hand, tend to listen to their *smartest customers*.” (Nadeau 1999, S.3)

Die Entwicklungen von Microsoft im Serverbereich (Windows NT, Windows 2000) verlief in den letzten Jahren vor allem auf Kosten der klassischen Unix-Systeme, die ähnliche Konzepte wie Linux/OSS vertreten. Es bleibt deshalb eine wichtige Herausforderung für Linux und OSS, die mit der Mächtigkeit einhergehende Komplexität durch geeignete Benutzerkonzepte auf das notwendige Maß zu reduzieren.

Abschließend hier noch einmal die wichtigsten Inhalte dieses Kapitels:

OSS hat sich fest in Unternehmen und Verwaltungen etabliert. OSS ist dabei am stärksten im Bereich der Basistechnologien/Infrastrukturen auf Servern vertreten. Typische Einsatzszenarien sind z. B. der Einsatz im Intranet, als Virtual Private Network, als Internetserver, in heterogene Netzen und zur Datenarchivierung in kleinen und mittleren Unternehmen und in Abteilungen. OSS und speziell Linux konkurriert dabei hauptsächlich mit klassischen Unix-Systemen, nur bedingt mit Windows.

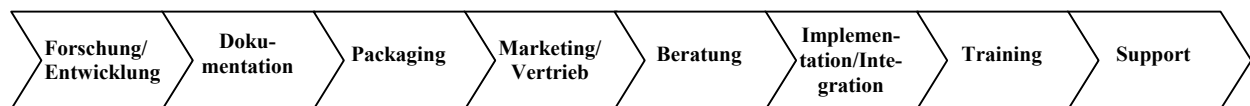
Stärken von OSS im kommerziellen Bereich sind die Sicherheit, die Geschwindigkeit, die Offenheit und Stabilität und der Preisvorteil. Die Schwächen von OSS sind der Mangel an Applikationen, das Desktop-Einsatzgebiet, mangelndes Vertrauen/Investitionssicherheit und die Administrierbarkeit.

4 Anbieterseite: Kommerzielle Open Source Geschäftsmodelle

Das vorherige Kapitel hat sich mit dem Einsatz von OSS auf der Nachfragerseite beschäftigt. Auf dieser Grundlage wird nun die Anbieterseite untersucht, also die Unternehmen, die ihren Geschäftszweck auf OSS stützen und Produkte und Dienstleistungen an die Nachfrager verkaufen. Welche Möglichkeiten gibt es, mit OSS Geld zu verdienen? Wie unterscheiden sich verschiedene OSS-Geschäftsmodelle? Welche OSS-Geschäftsmodelle sind tragfähig? Wie haben sich die OSS-Geschäftsmodelle verändert?

4.1 Wertschöpfungsprozesse bei IT-Unternehmen

Die Prozesse in einem Unternehmen können anhand einer Wertschöpfungskette gegliedert werden (vgl. Kapitel 2.3.1). Wie sehen die Schritte der Wertschöpfungskette aus Sicht der Anbieter genau aus?



**Abbildung 11: Allgemeine Software Value Chain
(angelehnt an Cimetiere o.J. und Zerdick et al. 1999)**

Forschung und Entwicklung ist der Abschnitt der Wertschöpfungskette, in der Software entwickelt wird. Basierend auf einem Entwicklungsplan verfassen Softwareautoren, meist im Team, den Quelltext der Software. Durch Compiler wird der Quelltext in Maschinencode übersetzt und dadurch ausführbar. Der Entwicklungsprozeß kann mit Entwicklungswerkzeugen technisch unterstützt werden. Dadurch müssen Teams nicht räumlich verbunden sein.

Bei der *Dokumentation* wird Information zur Software und zum Einsatz der Software angefertigt. Dokumentation entsteht dabei erstens im Quellcode.²⁸ Zweitens wird Dokumentation über den Betrieb der Software in Form von Handbüchern (selten bei OSS) oder elektronischen Hilfsdokumenten erstellt.

Packaging bezeichnet den Vorgang, in dem die Einzelteile eines Produktes (Software, Dokumentation usw.) in ein Paket zusammenfasst werden. Bei kommerzieller Software ist dies üblicherweise ein Regalprodukt (commercial off the shelf), bei OSS kann dies ein Softwarepaket sein, das im Internet bereitgestellt wird.

Im Bereich *Marketing/Vertrieb* sind die Vermarktungs- und Absatzaktivitäten eines Unternehmens zusammengefaßt. Im Marketing definiert das Unternehmen, welche Produkte wie an wen verkauft werden. Marketingstrategien werden nach den „4P“ unterschieden:

²⁸ Bei OSS ist es üblich, den Quellcode ausführlich zu dokumentieren, so daß er auch von Dritten nachvollzogen werden kann.

Produkt (welches Produkt, Gestaltung, Umfang), Price (Preis des Produktes), Place (wo wird das Produkt verkauft) und Promotion (wie wird das Produkt beworben). Von besonderer Bedeutung sind Standards. Wer eine Software als Standard im Markt durchsetzt, hat einen hohen Wettbewerbsvorteil. Im Desktop-Bereich ist Microsoft diese Standardsetzung gelungen.

Bei Massenprodukten wird ein Partnermodell mit indirektem Vertrieb über Händler oder andere Hersteller (Bundling) verfolgt. Bei individuelleren Produkten mit hohem Dienstleistungsanteil werden Produkte über Dienstleistungspartner vertrieben.

Beratung beschreibt die Unterstützung des Kunden vor der eigentlichen Software-implementation. Dazu gehört das Anfertigen von Studien, Analysen und Konzepten und eventuell die Anpassung der Unternehmensprozesse an die implementierte Software.

Implementation/Integration ist der Abschnitt, bei dem die Software vor Ort installiert wird. Wenn bereits IT-Systeme vorhanden sind, wird das neue System auf die Zusammenarbeit mit diesen Systemen konfiguriert und Schnittstellen angepaßt.

Training beschreibt die Schulung des Kunden durch den Dienstleister. Schulung kann sich dabei erstens auf die technischen Aspekte der Software beziehen, dabei wird das IT-Fachpersonal des Kunden geschult. Zweitens kann sich Schulung auf Nutzung und Anwendung der Software beziehen, dabei werden die Anwender der Software geschult.

Im Bereich *Support* werden die installierten IT-Systeme im laufenden Betrieb betreut. Dies kann im einfachsten Fall die Unterstützung der Mitarbeiter vor Ort bedeuten, aber auch bis hin zum vollständigen Betrieb der IT-Systeme durch den Dienstleister (outsourcing) reichen.

Ein Unternehmen muß nicht alle Teile der Wertschöpfungskette selber abdecken. Viele Softwareunternehmen beschränken sich auf Forschung, Entwicklung, Dokumentation und Packaging. Marketing und Vertrieb werden von Partnerunternehmen übernommen. Auch die Integration/Implementation kann im Rahmen von Partnerkonzepten ausgelagert werden (s. Einführung der Value Chain in Kapitel 2.3.1).

4.2 Modell zur Analyse von Geschäftsmodellen

Bevor mehrere Geschäftsmodelle verglichen werden, soll zunächst das in der vorliegenden Arbeit zugrundeliegende Verständnis des Begriffs „Geschäftsmodell“ erläutert werden. In dieser Arbeit wird auf ein Modell der Unternehmensberatung Mercer zurückgegriffen (Mercer 2001):



Abbildung 12: Business Design Modell

Die *Marktpositionierung* beschreibt die Beziehung eines Unternehmens zum Markt. Insbesondere werden hier die Kundensegmente, das eigene Leistungsangebot und die Differenzierung zu den Mitbewerbern beschrieben. Folgende Fragen werden dort beantwortet:

- Was ist das Produkt?
- Welchen Preis hat das Produkt?
- Wer sind die Kunden?
- Was ist die Marketing- und Vertriebsstrategie?

Das *Gewinnmuster* beschreibt das Modell, mit dem das Unternehmen aus den Beziehungen zu den Kunden Umsätze erzielt:

- Wie verdient das Unternehmen Geld?

Der Abschnitt *Ressourcenfokus* beschäftigt sich mit der Frage, welche Teile der Wertschöpfungskette das Unternehmen selber durchführt (und welche nicht) und welche Ressourcen hierzu benötigt werden:

- Was macht das Unternehmen selber?
- Was lagert das Unternehmen an Partner aus?
- Welche Partner hat das Unternehmen?

Strategische Absicherung sind die Maßnahmen, die das Unternehmen trifft, um sich gegen Wettbewerber abzusichern. Hier werden Markteintrittsbarrieren, Alleinstellungsmerkmale und die Nachhaltigkeit des Geschäftsmodells beschrieben:

- Warum kaufen die Kunden bei diesem Anbieter?
- Wer sind Wettbewerber?
- Was sind Bedrohungspotentiale?
- Wie schützt sich das Unternehmen vor Nachahmern?

Der Aspekt *Organisation, Mitarbeiter und Kultur* beschreibt den Kern des Unternehmens, mit dem die vorher beschriebenen Maßnahmen geplant und umgesetzt werden:

- Welche Anforderungen stellt das Unternehmen an die Mitarbeiter?
- Welche Organisationsform und Kultur hat das Unternehmen?

Aus Sicht des Autors eignet sich dieses Modell deshalb gut für die Analyse von OSS-Geschäftsmodellen, weil es die Umwelt des Unternehmens einbezieht. Andere Definitionen von Geschäftsmodellen beziehen sich in erster Linie auf ein einzelnes Unternehmen oder einen Unternehmenstyp. Sie beschränken sich auf die Beschreibung der Prozesse, die das Unternehmen im Rahmen seiner Geschäftstätigkeit ausführt („Binnenperspektive“). Marktorientierte Geschäftsmodelldefinitionen wie z. B. das Modell von Mercer haben den Vorteil, daß sie das Unternehmen in ein Spannungsfeld zwischen Kunden und Wettbewerber setzen und damit neben der Binnenbetrachtung auch die Außensicht auf den Kunden beachtet.²⁹ Das Modell ist dadurch erstens sehr praxisnah und zweitens ermöglicht es, die Wettbewerbszusammenhänge innerhalb der OSS-Welt und zu den etablierten Mitbewerbern zu beschreiben.

4.3 Produkt-Geschäftsmodelle

Der OSS-Produktmarkt wurde im Kapitel 2 in neun Segmente unterteilt. Da sich Anbieter nicht immer nur auf ein Marktsegment beschränken, werden diese Segmente im folgenden zu Unternehmenstypen zugeordnet.

- Die Marktsegmente Betriebssystem für Server, Client und Embedded werden vom Unternehmenstyp „OSS-Distributor“ angesprochen.
- Die Marktsegmente Applikationen für Server, Client und Embedded werden vom Unternehmenstyp „OSS-Applikations-Anbieter“ angesprochen.
- Die Marktsegmente Appliances für Server und Client werden vom Unternehmenstyp „OSS-Appliance-Hersteller“ angesprochen.

²⁹ Eine ausführliche Begründung, warum erfolgreiche Unternehmen den Kunden in den Mittelpunkt der Wertschöpfung stellen müssen, liefert die Unternehmensberatung Mercer in Slywotzky et al. (1997).

- Das Segment Appliances für Embedded-Systeme wird keinem eigenen Typ zugeordnet, sondern im Teil „Sonstige Geschäftsmodelle“ behandelt.

	Server-Linux-Distribution	Desktop-Linux-Distribution	Embedded-Linux-Distribution	OSS-Distributoren
	Server-Applikationen	Desktop-Applikationen	Embedded-Applikationen	OSS-Applikations-Anbieter
OSS-Appliance-Hersteller	Server Appliances	Thin Clients, Set Top Boxen	PDA's, MP3-Player, Smartphones	Sonstige-Geschäftsmodelle

Abbildung 13: Zuordnung von Unternehmenstypen zu Marktsegmenten

Es ist wichtig zu betonen, daß die Überführung in Unternehmenstypen durch den Autor eine idealisierte Darstellung ist. Viele Geschäftsmodelle kommen nicht in Reinform vor, sondern sind gemischte Modelle. So sind z. B. viele OSS-Distributoren gleichzeitig auch OSS-Dienstleister. Mehr zu Mischformen und Transformationen der Geschäftsmodelle in der Zusammenfassung am Ende dieses Kapitels.

4.3.1 Geschäftsmodell OSS-Distributor

OSS-Distributionen fassen OSS-Komponenten auf Datenträgern zusammen und machen sie durch Installations- und Administrationsroutinen für Endkunden als Komplettprodukt nutzbar. Das Geschäftsmodell der Distributoren besteht aus Entwicklung, Vermarktung, Vertrieb und Support dieser Distributionen.

Das OSS-Distributions-Geschäftsmodell unterscheidet sich in einigen Eigenschaften von anderen OSS-Geschäftsmodellen.

- Es ist das älteste OSS-Geschäftsmodell (SuSE-Gründung 1992).
- Es ist ein „pures“ OSS-Geschäftsmodell in dem Sinne, daß es ohne OSS nicht existieren würde.
- Es ist das Geschäftsmodell mit der größten Bedeutung im OSS-Markt. Dies läßt sich daran festmachen, daß einige der Anbieter börsennotiert sind, teilweise zweistellige Euro-Millionen Umsätze erzielt werden und einige hundert Mitarbeiter beschäftigt werden.

Im folgenden werden Linux-Distributionen als Beispiel für OSS-Distributionen untersucht (andere OSS-Distributionen sind z. B. OpenBSD oder FreeBSD). Die wichtigsten Linux-Distributoren sind SuSE (Deutschland), Red Hat (USA), Turbolinux (USA/Japan), Caldera (USA) und Mandrake (Frankreich, Schwerpunkt auf Desktop-Linux).

Der überwiegende Teil einer Linux-Distribution besteht aus frei verfügbaren OSS-Komponenten. Rosenberg (2000) schätzt beispielsweise, daß 87% einer Red Hat Linux-Distribution nicht von Red Hat stammt. Nur wenige Komponenten wie z. B.

Installationsroutinen, Administrationsoberflächen oder die Hardwareerkennung werden selber entwickelt. Manche Linux-Distributoren geben diese Kernkomponenten nicht als OSS frei,³⁰ sondern bieten sie nur als proprietäre Software an. In diesem Fall unterliegt die Distribution nicht mehr vollständig der GPL-Lizenz und darf nicht mehr ohne Einschränkungen kopiert werden.

Zusätzlich zu eigenen Entwicklungen werden von den Distributoren auch fremde OSS-Projekte gefördert. So fördert z. B. Red Hat das Entwicklungsprojekt „GNOME“³¹. Diese unabhängigen Entwicklungen kommen allerdings nicht nur Red Hat, sondern auch Mitbewerbern zugute.

Zusätzlich zum Produktverkauf wird ein Herstellersupport (Telefon, Email) angeboten, der für einen bestimmten Zeitraum kostenlos ist. Das Angebot wird in der Regel auf einen definierten Umfang (Installationsprobleme) begrenzt. Darüber hinaus bieten manche Distributoren eine Wissensdatenbank mit Problemlösungen an, die kostenfrei bzw. durch Registrierung genutzt werden kann.

Alle darüber hinaus gehenden Leistungen werden als professionelle Dienstleistungen (im Kapitel 4.4 als „OSS-Dienstleister“ beschrieben) angeboten.

4.3.1.1 Marktpositionierung

Die Distributoren bewegen sich im Markt der Computer-Betriebssysteme für Server, Clients, und Embedded Systems (s. Kapitel 2.6.1.1). Zielgruppen der Distributoren sind sowohl Unternehmen (dort die IT-Entscheider und -Administratoren) als auch Privatkunden.

Distributionen werden in Releases zusammengefaßt und mit einer Versionsnummer versehen. Sie werden dabei nach verschiedenen Versionen (z. B. „Red Hat 7.2“), Hardwareplattformen (z. B. Intel, Itanium, Embedded) und Zielkunden (Red Hat Linux und Red Hat Linux Professional) unterschieden. Releases werden im Abstand von wenigen (bei Red Hat und SuSE: drei) Monaten produziert und auf Datenträgern wie CD und DVD gepreßt und/oder auf Internetservern abgelegt.

Viele Distributoren bieten ein Standard-Produkt für Endkunden und eine Professional-Version für Unternehmenskunden an. Die Professional-Version unterscheidet sich dabei technisch nur wenig, ist aber mit zusätzlichen Angeboten auf den Unternehmensmarkt zugeschnitten (24h-Telefonhotline, Schulungsangebot). Teilweise wird auch kommerzielle und/oder proprietäre Software mitgeliefert.

Die folgenden Tabellen zeigen exemplarisch die Preisgestaltung der Distributionen und der Zusatzprodukte:

³⁰ Zum Beispiel die Installations- und Verwaltungssoftware „yast“ von SuSE Linux.

³¹ Ziel dieses Projekts ist die Entwicklung einer einfachen und komfortablen Oberfläche, s. S. 14 ff.

	SuSE Linux	Red Hat Linux
Standard	DM 97.60	DM 126.90
Professional	DM 154.31	DM 449.00
Itanium-Prozessor	DM 1474.70	DM 1379.00
SPARC-Prozessor	DM 369.00	-
Alpha-Prozessor	DM 98.00	-

Tabelle 5: Preise für Linux-Distributionen
(Quelle: Homepages der Distributoren, Stand 12/2001)

	Preis	Beschreibung
SuSE eMail Server II	DM 579.00	E-Mail-Lösung für Unternehmen
SuSE Linux Entertainment Pack	DM 111.00	Spielesammlung
Red Hat High Availability Server	DM 5300.00	Linux- Hochverfügbarkeitslösung
Red Hat Linux für Oracle 8i	DM 3700.00	Datenbankplattform für Oracle unter Linux

Tabelle 6: Preise für Zusatzprodukte der Linux-Distributoren
(Quelle: Homepages der Distributoren, Stand 12/2001)

OSS-Distributoren verfolgen eine Branding-Strategie (Aufbau einer hochwertigen Marke). Bob Young, Gründer von Red Hat, bezeichnet die Red Hat Distribution als „commodity“ und vergleicht die Strategie seines Unternehmens mit der eines Ketchupherstellers (Dibona 1999):

„Ketchup is nothing more than flavored tomato paste. Something that looks and tastes a lot like Heinz Ketchup can be made in your kitchen sink without so much as bending a copyright rule. It is effectively all freely-redistributable objects: tomatoes, vinegar, salt, and spices. [...] Heinz has 80% of the ketchup market because they have been able to define the taste of ketchup in the mind of ketchup consumers. Now the Heinz Ketchup brand is so effective that as consumers we think that ketchup that will not come out of the bottle is somehow better than ketchup that pours easily! This was Red Hat's opportunity: to offer convenience, to offer quality, and most importantly to help define, in the minds of our customers, what an operating system can be. At Red Hat, if we do a good job of supplying and supporting a consistently high-quality product, we have a great opportunity to establish a brand that Linux OS customers simply prefer.” (Dibona 1999, Kapitel „Giving it away“, Abschnitt „We Are in the Commodity Product Business“).

Distributoren positionieren das Unternehmen gleichermaßen bei Endkunden, Unternehmenskunden und der Open Source Community. Die Linux-Distributoren setzen dabei auf einen Werbemix aus Produkt- (Zeitschriften und Internet) und Unternehmenswerbung (Messeauftritte und Imageanzeigen) und auf PR-Arbeit.

Ein wichtiges Marketinginstrument ist Sponsoring. Da sich OSS-Distributoren in einem Spannungsfeld zwischen der nicht monetär orientierten Open Source Community und der eigenen kommerziellen Ausrichtung befinden, geben die Unternehmen durch Sponsoring etwas an die Linux-Community „zurück“. Beispiele sind gesponserte Anwendertreffen (Linux User Groups), Messen, Kongresse und die Unterstützung von OSS-Entwicklern.

Die Linux-Distributoren setzen in der Regel auf mindestens drei Distributionskanäle. Erstens den Direktverkauf über die Internetseiten des Distributors, zweitens den Vertrieb über Presse- und Einzelhandelskanäle (Buchhändler, Zeitschriften etc.) und drittens Partnerschaften mit Computerherstellern. SuSE und Red Hat kooperieren z. B. mit IBM und Compaq (vorinstallierte Linux-Computersysteme) sowie kleinen und mittleren IT-Dienstleistern, die Distributionen im Rahmen von Projekten einsetzen. Die Partner können dabei auch Dienstleistungen von den Distributoren beziehen (1st level support, Wartungsverträge) und sind dann gleichzeitig auch Kunden.

4.3.1.2 Gewinnmuster

Das Gewinnmodell der Distributoren besteht aus dem Verkauf von Linux-Distributionen auf Datenträgern über Partner an Endkunden. Dabei wird ein einmaliger Kaufpreis erzielt. Es fallen keine weiteren Gebühren oder Erlöse an. Die Distributoren bieten regelmäßig neue Releases an (bei SuSE ca. vier mal pro Jahr). Manche Distributoren (z. B. Red Hat) legen die Distribution vollständig zum download bereit. In diesem Fall wird mit dem Produkt überhaupt kein Umsatz erzielt.

Weiterhin bieten die Distributoren zusätzliche Dienstleistungen wie Support, Wartungsverträge und u.U. Integration/Implementation an.

Der Distributor SuSE ist im Moment (Stand Anfang 2002) nicht profitabel, plant aber den Eintritt in die Gewinnzone für Ende 2002: „Nach umfassenden Umstrukturierungen im vergangenen Jahr soll das Unternehmen im zweiten Halbjahr 2002 profitabel sein [...]. Vor dem allgemeinen Linuxboom war SuSE bereits profitabel, doch war es dem Unternehmen nicht gelungen, im Zuge seiner Expansion auch nachhaltige Gewinne zu erwirtschaften.“ (N.N. Linux-Enterprise 2002). Red Hat hat im Geschäftsjahr 2001 einen Verlust von \$86,7 Millionen erwirtschaftet (Quelle: www.redhat.com). Mandrake hat Anfang 2002 gewarnt, daß die wirtschaftliche Existenz bedroht ist, wenn sich nicht ausreichend zahlende Benutzer für das Support-Angebot „Mandrake Club“ finden (Quelle: www.linux-mandrake.com/en/mdkfuture.php3).

4.3.1.3 Ressourcenfokus

Das Geschäftsmodell der Distributoren umfaßt aus Sicht des Autors die Bereiche Forschung/Entwicklung, Dokumentation, Packaging, Marketing/Vertrieb und Support aus der Wertschöpfungskette.

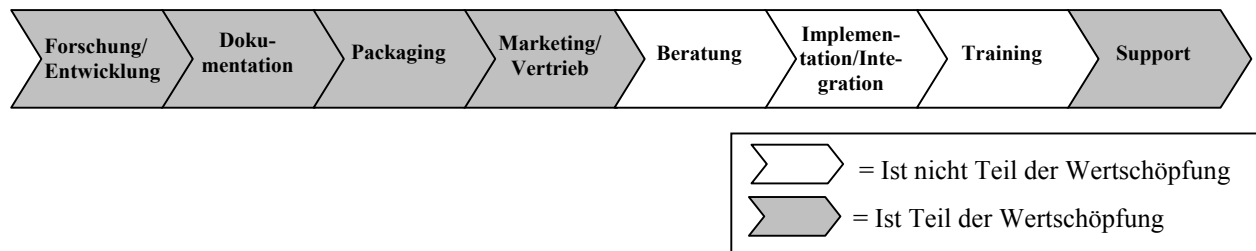


Abbildung 14: Ressourcenfokus der Distributoren

Partner der Distributoren sind z. B.

- Entwicklungspartner, die Software zuliefern (OSS-Entwickler oder Unternehmen)
- Marketing- und Vertriebspartner (Buchhändler, Hardwarehersteller)
- Dienstleistungspartner (z. B. große IT-Anbieter und regionale Systemhäuser, Dienstleister und Integratoren)

4.3.1.4 Strategische Absicherung

OSS-Distributoren haben im Gegensatz zu Herstellern proprietärer Software keine Möglichkeit, ihre Software zu schützen (s. Kapitel 2.3.3). Die Software auf den Distributionen steht unter der GPL und ist deshalb frei kopierbar. Teilweise dürfen sogar die gesamten Distributionen frei kopiert werden (Red Hat; SuSE hingegen schützt sich vor dem Kopieren der gesamten CD, indem nicht alle Komponenten unter der GPL stehen).

Die Distributoren setzen deshalb zur strategischen Absicherung auf die Etablierung einer globalen Marke und eines Standards mit möglichst hoher Verbreitung. Die Markteintrittsbarrieren für Mitbewerber sind dabei relativ gering, denn die OSS-Komponenten der Produkte sind für jeden Mitbewerber frei verfügbar.

Die Käufer der Linux-Distributionen verfügen über keine hohe Macht. Die Zielgruppen sind stark fragmentiert (geographisch, unterschiedliche Einsatzzwecke von privat bis kommerziell) und es existieren viele sehr kleine Einkaufsvolumina.

Strebt ein Kunde einen Systemwechsel auf eine andere Betriebssystemtechnologie (Windows, Unix oder z. B. SuSE zu Red Hat) an, entstehen Umstellungskosten. Dadurch werden die Kunden an die eigene Distribution gebunden.

Da die Preise für die Linux-Distributionen niedrig - im Vergleich mit kommerziellen Produkten sogar sehr niedrig sind - kann man von einer relativ hohen Preissensitivität der Kunden ausgehen, insbesondere im Privatbereich. Proprietäre Produkte können deshalb nicht über Preisstrategien, sondern nur über Leistungsdifferenzierung in den Markt eingeführt werden.

Mögliche Bedrohungen für die Produkte der Distributoren sind alternative Bezugswege für die Distribution. Bedeutsam ist hierbei vor allem der Download der Distribution aus dem Internet, da hier kein Umsatz generiert wird. Im Moment kann diese, von den Distributoren sogar geförderte, Möglichkeit mit der Marken- und Standardisierungsstrategie begründet werden. Langfristig nimmt diese Bedrohung zu, da immer mehr Anwender über Hochgeschwindigkeitszugänge verfügen und das Internet als Vertriebskanal an Bedeutung gewinnt.

Eine weitere Bedrohung der Distributionen sind vorinstallierte Computersysteme. Bei wachsendem Marktanteil ist davon auszugehen, daß die Zahl der vorinstallierten OSS/Linux-Computersysteme zunehmen wird. Wenn die Distribution komplett unter der GPL steht - wie im Falle von Red Hat - muß der Computerhersteller dabei keine Lizenzabgaben oder sonstige Abgaben an den Distributor zahlen, der Umsatz für den Distributor fällt also in kompletter Höhe weg.

Eine theoretische Bedrohung der Marktstellung der Distributoren liegt im Verhältnis zu den Entwicklern. Sollte aus irgendeinem Grund - beispielsweise durch die Verletzung der GPL oder dem „Missbrauch“ einer Marktposition - ein Linux-Distributor den Unmut der Open Source Community auf sich ziehen, wäre ein Boykottaufruf der OSS-Entwickler denkbar. Es ist allerdings kein ernsthafter Fall bekannt.³² Selbst wenn es dazu käme, muß das wechselseitige Abhängigkeitsverhältnis der Distributoren und Entwickler berücksichtigt werden. Obwohl die Entwickler kein Geld bekommen, sind sie dennoch auf die Verbreitung der Software angewiesen, denn sonst macht die Entwicklung nur begrenzt Sinn.

4.3.1.5 Organisation, Mitarbeiter und Kultur

Im Rahmen der Brandingstrategie bauen die Distributoren eine interationale vor-Ort-Präsenz auf. Dabei ist Red Hat am weitesten fortgeschritten. SuSE und Mandrake haben einen europäischen, Turbolinux einen asiatischen Schwerpunkt.

Die Organisation der Distributoren teilt sich zum einen in zentrale Softwareentwicklung und zum anderen in Dienstleistungen, die an den jeweiligen Unternehmensstandorten angeboten werden. Für die Softwareentwicklung benötigen die Distributoren qualifiziertes Entwicklungspersonal. Für den Dienstleistungsbereich werden u. a. Berater, Hotlinemitarbeiter und Systemadministratoren benötigt.

³² Ein deutscher Physikstudent ruft im Internet dazu auf, mit Hilfe einer Vereinskonstruktion 25% der Aktien von SuSE zu übernehmen, weil dem Autor „sowohl das Konzept, als auch die Umsetzung der Firma SuSE GmbH AG in Bezug auf die Umsetzung des freien Betriebssystem Linux nicht paßt“ (www.physik.tu-cottbus.de/users/heinold/suse/). Das Projekt klingt nicht nur abenteuerlich, sondern ist auch seit Anfang 2000 nicht aktualisiert worden.

Einige Distributoren unterstützen ihre Branding-Strategie dadurch, daß sie prominente OSS-Entwickler als Mitarbeiter verpflichten. Dadurch entsteht aus Sicht der Kunden eine Qualitätsaufwertung, denn die Technologie wird „aus erster Hand“ geliefert.

4.3.2 Geschäftsmodell OSS-Applikations-Anbieter

Das vorige Geschäftsmodell hat sich mit den Distributionen beschäftigt. Die Wertschöpfung der Distributoren besteht vor allem darin, fremde Software zu packagen und zu vertreiben. Dieses Kapitel beschäftigt sich mit dem Fall, daß ein Unternehmen eigene Software entwickelt und unter eine OSS-Lizenz stellt.

Bei der Betrachtung von OSS-Applikations-Anbietern können aus Sicht des Autors drei Fälle unterschieden werden:

- Fall 1: Ein Unternehmen gibt eine Software, die es zu einem früheren Zeitpunkt proprietär entwickelt hat, ab einem bestimmten Zeitpunkt im Quellcode frei.
- Fall 2: Ein Unternehmen beginnt, eine Software ab einem bestimmten Zeitpunkt unter einer OSS-Lizenz zu entwickeln.
- Fall 3: Ein Unternehmen „übernimmt“ zu einem bestimmten Zeitpunkt ein bis dato existierendes OSS-Projekt und betreut dieses ab diesem Zeitpunkt kommerziell.

Im ersten Fall kehrt das Unternehmen den klassischen OSS-Entwicklungsprozeß um und „konfrontiert“ die OSS-Welt mit einem fertigen OSS-Produkt (Beispiel: Netscape mit dem Netscape-Browser, genannt Mozilla).

Der zweite Fall verläuft analog zum klassischen OSS-Entwicklungsprozeß. Der einzige Unterschied besteht darin, daß keine bzw. nur wenige unabhängige Personen an dem Projekt arbeiten, da ein einzelnes Unternehmen den Prozeß dominiert (Beispiel: Red Hat/GNOME).

Im dritten Fall wechselt der Charakter des OSS-Modells zu diesem Zeitpunkt von frei auf kommerziell.³³

4.3.2.1 Marktpositionierung

Die Anbieter dieses Geschäftsmodells bewegen sich im gesamten Markt für Software-Applikationen. Dieser umfaßt alle Arten von Betriebssystemen, Hardwareplattformen und Anwendungszwecken (s. Kapitel 2.6.1.2).

Meistens werden OSS-Applikationen für OSS-Betriebssysteme (Linux) entwickelt. Theoretisch ist es aber auch denkbar, daß ein Unternehmen OSS-Applikationen für Windows-Betriebssysteme freigibt.³⁴

³³ Gründe für diesen Fall können z. B. sein, daß der oder die Hauptentwickler das Projekt nicht mehr weiterbetreuen können oder wollen oder daß das Projekt so erfolgreich ist, daß es zu viel Ressourcen beansprucht.

³⁴ Eine Reihe von OSS-Softwaretools aus der OSS-Community sind interessanterweise auch für proprietäre Systeme verfügbar, z. B. Apache und Perl für Windows.

Für die Betrachtung des Geschäftsmodells ist es egal, um welche Art von Software es sich genau handelt. Es könnte sich z. B. um eine kleine Zusatzsoftware handeln oder um eine komplette Office-Suite. Der zugrundeliegende Mechanismus des Geschäftsmodells ist davon nicht betroffen. Je nach Art der Software variiert entsprechend auch die Zielgruppe.

Die Vermarktung der Software findet meist mit Hilfe der OSS-Community statt. Je „glaubwürdiger“ und akzeptierter das Gewinnmodell ist, desto mehr wird die Community die Software bekannt machen und nutzen. Dies geschieht z. B. durch Webseiten, in Newsgroups, auf Messen und in Zeitschriften. Die Software wird in der Regel auf Projekt-Homepages im Internet oder bei OSS-Mediatoren (s. Kapitel 4.5) bereitgestellt.

Der Vertrieb der Produkte erfolgt über das Internet, in seltenen Fällen können Datenträger und Handbücher bestellt werden (z. B. bei Staroffice).

4.3.2.2 Gewinnmuster

Beim Geschäftsmodell OSS-Applikations-Anbieter können folgende Gewinnmuster unterschieden werden:

Verschenken der Software³⁵

In diesem Fall gibt der Anbieter die Software komplett unter einer OSS-Lizenz frei. Dem Anbieter ist es praktisch nicht mehr möglich, mit dem Verkauf der Software Erlöse zu erzielen (laut GPL zwar möglich, aber nicht praktikabel, s. Kapitel 2). Zwar behält der Anbieter oft noch eine Schlüsselfunktion (z. B. die Homepage des OSS-Projekts), aber in der Konsequenz befindet er sich auf der gleichen Stufe wie jeder andere Entwickler (und Mitbewerber!). Ein Beispiel für dieses Modell ist der Anbieter Sun mit dem OSS-Office-Produkt OpenOffice, das ursprünglich proprietär und kommerziell war.³⁶

Diese Variante ist die radikalste Ausprägung des Geschäftsmodells. Der Anbieter verschenkt Software ohne primäre Gegenleistung und hat u. U. sogar noch laufende Kosten (Koordination des Entwicklungsprozesses, Infrastruktur, Personalkosten etc.). Für den Anbieter kann sich diese Variante wirtschaftlich nur lohnen, wenn er eine Überleitung zu Sekundärgeschäftsmodellen vornehmen kann.

Eine andere Motivation für das Verschenken der Software kann darin liegen, daß sich ein Anbieter von dem Produkt keinen kommerziellen Erfolg mehr verspricht. Dieser Schluß liegt z. B. beim Netscape-Browser/Projekt Mozilla und beim Office-Paket Sun Staroffice/Projekt OpenOffice nahe: Beide Unternehmen haben den Wettbewerb gegen Microsoft verloren und machen „aus der Not eine Tugend“. Anstatt einen bereits verlorenen Kampf gegen den Marktführer zu führen, attackieren sie ihn mit einer freien Variante. Mögliche Vorteile sind der Imagegewinn und die Schwächung der dominanten Position von Microsoft, die ja wiederholt zum Eintritt in neue Märkte genutzt wurde.

³⁵ Für eine allgemeine Analyse (unabhängig von OSS), warum es für Unternehmen sinnvoll sein kann, Produkte zu verschenken vgl. Busa (1999).

³⁶ In der Zwischenzeit wurde das Sun-eigene Produkt StarOffice, dem OpenOffice (Lizenz: LGPL) zugrunde liegt, wieder kommerziell, wird also nicht mehr kostenlos verteilt.

Lizensierung nach Zeit

Dieses Modell stellt eine Variante des oben genannten Modells dar. Der wesentliche Unterschied ist, daß der Anbieter die Software in Versionen unterteilt und diese unterschiedlich behandelt. Aktuelle Versionen der Software sind kommerziell (nicht unbedingt proprietär). Ab einem bestimmten Zeitpunkt (feste Zeitspanne oder bei Erscheinen einer neuen Version) wird die Version unter einer OSS-Lizenz freigegeben.

Mit diesem Verfahren geht der Hersteller einen Mittelweg: Die Investitionen in die Entwicklung sind für einen Zeitraum geschützt und können wirtschaftlich genutzt werden. Die Herausforderung dieses Modells besteht erstens darin, zahlende Kunden zu finden, obwohl der Kunde ab dem Freigabezeitpunkt die Software auch unentgeltlich nutzen könnte. Die zweite Herausforderung besteht darin, die OSS-Entwicklung „in Schwung“ zu bringen, denn wegen des kommerziellen Kompromisses (der ja primär dem Anbieter nutzt), könnten unabhängige OSS-Entwickler kein Interesse an der Mitarbeit haben.

Lizensierung nach Zielgruppe

Diese Variante differenziert nicht nach Zeitpunkt, sondern nach Zielgruppe. Der Anbieter unterteilt die Kunden beispielsweise nach Privatkunden und Unternehmen (Variante: kleine Unternehmen/große Unternehmen). Je nach Segment wird die Software dann kostenfrei oder kommerziell angeboten, in beiden Fällen ist sie OSS. Diese Variante ähnelt dem Shareware-Modell, denn auch dort wird teilweise nach Zielgruppe unterschieden (allerdings wird bei Shareware kein Quellcode mitgeliefert). Ein Beispiel für diese Variante ist der deutsche E-Commerce-Anbieter IntraDAT (mittlerweile insolvent) mit dem Produkt „vshop“, der dieses Modell praktiziert hat:

„Der Bereich E-Commerce ist einer der vielversprechendsten in der IT-Branche. Open Source heißt für uns und unsere Kunden: Stabilität und Flexibilität bei hoher Entwicklungsgeschwindigkeit; ein neuer Standard der Software-Entwicklung, der nicht auf herkömmlichem Weg erreicht werden kann.“ (Quelle: www.intradat.de).

Eine andere Zielgruppendifferenzierung findet bei der freien Datenbank MySQL (www.mysql.com) statt. Sie ist nur dann kostenpflichtig, wenn ein Dritter das Produkt verkauft oder in ein eigenes Produkt integriert.

Lizensierung nach Leistungsumfang

In dieser Variante differenziert der Anbieter das Produkt in verschiedene Versionen, die unterschiedlichen Leistungsumfang besitzen. Eine Basiskomponente wird als OSS freigegeben. Diese Basiskomponente ist einsatzfähig und hat auch einen Nutzen. Wenn die Ansprüche des Benutzers wachsen, wird eine professionelle, kostenpflichtige Version angeboten.

Dieses Modell operiert mit einem „Lockeffekt“. Der Vorteil für den Anbieter ist die kostenlose Werbung, der Vorteil für den Anwender die Testmöglichkeit. Der dahinterliegende Mechanismus ist ebenfalls aus der Shareware-Welt bekannt.

Lizensierung nach Zielplattform

Einige Anbieter lizenzieren Softwareprodukte nur auf bestimmten Plattformen als OSS. Beispielsweise ist die Softwarebibliothek QT des Unternehmens Trolltech kommerziell auf Windows-Systemen, aber kostenfrei für Unix- und Linux-Systemen erhältlich. Allerdings darf mit Produkten, die auf der freien Version basieren, kein Geld verdient werden.

Lizensierung nach Komponenten

Eine weitere Variante ist die Offenlegung einer wirtschaftlich weniger bedeutsamen Komponente, von der der Hersteller strategisch profitiert. Der Anbieter SAP hat beispielsweise die Datenbank SAP DB als Komponente seines ERP-Systems freigegeben und erhofft sich von der Freigabe einen Innovationsschub:

“SAP believes that times have changed, that databases are becoming part of the basic technology infrastructure, and as such, they need not be proprietary or complex. SAP sees that the time is right to drive open-source development of database technology and contribute to this effort by making SAP DB Open Source. Open-source development is revolutionizing the way software is created, as shown by the success of Linux, and SAP - which is already providing the first comprehensive e-business solution on Linux - wants to encourage this development.”
(Quelle: www.sapdb.org).

Ein andere Form dieser Variante ist die Freigabe einer Clientsoftware, während das gleichzeitig notwendige Server-Produkt kommerziell und/oder proprietär bleibt.

Proprietäre Software für OSS

Einige Anbieter bewegen sich zwar im OSS-Markt, bieten aber selber keine OSS-Software an. So verkauft z. B. der Anbieter Covalent proprietäre Zusatzsoftware zum Webserver Apache. Der Anbieter ArsDigita hat sogar seine ursprüngliche OSS-Software zurückgezogen und bietet diese nunmehr proprietär an. Diese Geschäftsmodelle spielen in der weiteren Analyse keine Rolle, da es sich nicht um OSS-Geschäftsmodelle handelt. Im Grunde sind die Anbieter normale Softwareunternehmen, die kommerzielle, proprietäre Software für OSS-Betriebssysteme anbieten (s. auch die Beschreibung der Applikationen für OSS in Kapitel 2.6.1.2).

Überleitung zu sekundären Gewinnmodellen

Häufig beschränken sich die Anbieter von freier Software nicht auf das Softwaregeschäft, sondern bieten zusätzlich Sekundärleistungen an. Diese finden sich in den späteren Abschnitten der Software-Wertschöpfungskette (z. B. Hersteller-Support für die Software, Wartungsverträge, Dokumentation und Training). Rosenberg (2000) vergleicht diese Vorgehensweise mit dem Rasierproduktehersteller Gillette. Dessen Rasierer werden zu niedrigen Preisen angeboten bzw. sogar verschenkt (Primärprodukt). Der langfristige Ertrag

wird mit dem wiederholten Kauf der Rasierklingen (Sekundärprodukt) erzielt. Aus Sicht des Autors hinkt dieser Vergleich allerdings etwas, da die Käufer der Gillette-Produkte aufgrund der proprietären Eigenschaften gezwungen sind, Gillette-Klingen zu verwenden (nur Original-Klingen passen). Genau solch ein systembedingter Zwang fehlt aber bei OSS!

Ein Vertreter dieses Modells ist der Softwareanbieter Zope Corporations mit ihrem Applikationsserver Zope. Die Software ist Open Source und der Hersteller bietet ein umfangreiches Business-Service-Angebot für die Software. Der Gründer von Zope Corporations:

„Going open source will increase our user base by a factor of 100 within three months. Wider brand and stronger identity leads to more consulting and increased valuation on our company. [...] The exit plan isn't about the golden eggs (the intellectual property) laid last year. It is about the golden goose and tomorrow's golden eggs. The shelf life of eggs these days is shrinking dramatically, and the value of an egg that no one knows about is tiny. Give the eggs away as a testament to the value of the goose and a prediction of eggs to come.” (Everit 2001, S. 1)

Die Freigabe als OSS ist folglich ein Kompetenz- und Glaubwürdigkeitsbeleg für die Sekundärangebote.

4.3.2.3 Ressourcenfokus

Folgende Abbildung des Autors zeigt die Teilung in einen primären, nicht-kommerziellen Abschnitt und einen sekundären, kommerziellen Abschnitt der Wertschöpfungskette.

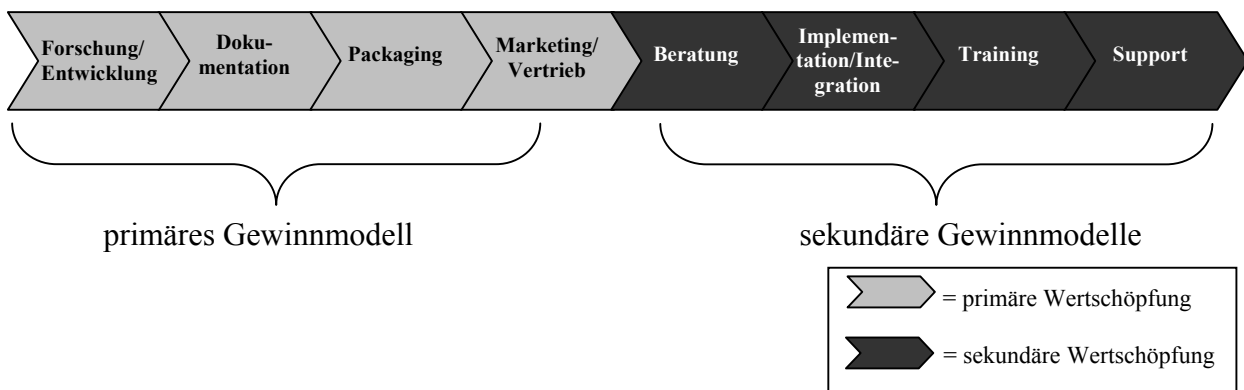


Abbildung 15: Wertschöpfungskette der OSS-Anbieter

Die OSS-Anbieter haben in der Regel keine Partner, da die Software direkt über das Internet vertrieben wird. Eventuelle Dienstleistungspartner würden mit dem sekundären Gewinnmodell konkurrieren.

4.3.2.4 Strategische Absicherung

Die strategische Absicherung der Anbieter erfolgt über zwei Faktoren: Preis und Standardsetzung. Der Preis ist ein starker Absicherungsfaktor, denn er kann in diesem Modell nicht unterboten werden. Für einige Mitbewerber wird es nicht möglich sein, die Software kostenlos freizugeben, z. B. weil der Anbieter zu klein ist oder weil die Software einen strategischen Faktor darstellt. Die Standardsetzung ist der zweite Absicherungsfaktor. Gelingt es einem Anbieter, mit einem OSS-Produkt einen Industriestandard zu setzen, schließt er Mitbewerber aus dem Wettbewerb aus. Im Sinne der Netzwerkeffekte verstärken sich einmal gesetzte Standards noch weiter, und wenn die Software noch dazu umsonst zu haben ist, werden die Kunden kaum zu Mitbewerbern wechseln.

Die Sekundärgewinnmodelle werden über das primäre Gewinnmodell abgesichert: Die Freigabe der Software belegt Kompetenz und Glaubwürdigkeit.

Die Käufer besitzen in der Regel wenig Macht, da sie wenig Druck aufbauen können, insbesondere wenn der Anbieter das einzige Unternehmen aus dem Marktsegment ist, der das Produkt verschenkt.

Mögliche Bedrohungen sind andere OSS- oder proprietäre Produkte. Für letztere bleibt nur die Leistungsdifferenzierung der eigenen Produkte z. B. durch bessere Qualität, mehr Leistungsumfang oder Support.

4.3.2.5 Organisation, Mitarbeiter und Kultur

Diese Faktoren unterscheiden sich nicht von den in der Softwarebranche üblichen Faktoren. Das Zentrum der Unternehmenstätigkeit ist die Softwareentwicklung. Es ist keine überregionale Organisation notwendig.

4.3.3 Geschäftsmodell OSS-Appliance-Hersteller

Appliances sind Geräte, die aus einer Hardware/Software/Betriebssystem-Kombination bestehen. Die Anbieter entwickeln zusätzlich zum OSS-Betriebssystemkern eigene Applikationen für die Schnittstelle zum Benutzer (Administration, Bedienoberfläche, Updatefunktionen und ähnliches)³⁷. Diese Applikationen sind meist proprietär, was aber nicht der GPL widerspricht, da die Software zwar zusammen aufgespielt ist, aber nicht logisch/technisch aufeinander aufbaut.

Der Support der Appliances wird in der Regel über Partner abgewickelt. Der Partner betreibt beispielsweise einen telefonischen Support und/oder eine Vor-Ort-Unterstützung (first level). Der Partner kann wiederum Support beim Hersteller einkaufen (second- oder third-level Wartungsvertrag), wenn es um komplexere Probleme geht.

³⁷ Einige wenige Appliance-Anbieter entwickeln über die Benutzerschnittstelle hinaus auch basistechnische Komponenten (z. B. der deutsche Anbieter Astaro mit einer eigenen Firewall-Software für die angebotene Appliance).

Beispiele für Appliance-Anbieter:

- Server-Appliances: Cobalt/Sun, IBM Whistle, VA Software, eSoft
- Thin Clients: IBM NetVista, Linware, Wyse
- Set top boxen: Nokia, PersonalTV, TiVo

4.3.3.1 Marktpositionierung

Die meisten Appliance-Anbieter haben ein ganzes Portfolio von Appliances. Dieses gliedert sich nach Einsatzzweck (verschiedene Aufgaben) und nach Zielgruppe (Standard oder Professional).

Zielgruppe der Appliances sind Unternehmen oder Privatpersonen, die die angebotenen Funktionen benötigen, diese aber nicht selber installieren wollen oder können.

Appliances konkurrieren mit individuell konfigurierten Softwarelösungen. Bei diesen Lösungen fällt neben den Kosten für Hardware, Betriebssystem und Applikationen auch Arbeitsaufwand an. Deshalb sind Appliances in der Regel preiswerter (EUR 500,- bis EUR 2500,-). Manche Typen von Appliances werden dennoch in einem relativ hohen Preissegment angeboten. Diese zielen weniger auf den Massenmarkt, sondern werden aufgrund der Spezialisierung auf eine bestimmte Funktion als besonders hochwertige Lösung positioniert (Beispiel: Firewall-Appliances, Preise zwischen EUR 1000,- und EUR 5000,-).

Appliances werden in der Regel über Partner vertrieben. Die Promotionaktivitäten werden deshalb primär im Umfeld der Partner durchgeführt (gemeinsame Messeauftritte und Anzeigenschaltungen). Die Vermarktung der Produkte steht im Vordergrund (im Gegensatz zur stärkeren Vermarktung des *Herstellers* wie z. B. bei den Distributor-Geschäftsmodellen).

Im Privatbereich werden Appliances über den Einzelhandel (Media Markt, Red Zac), im Unternehmensbereich über Partner (Distributoren, kleine IT-Systemhäuser, IT-Händler) vertrieben.

4.3.3.2 Gewinnmuster

Das Gewinnmodell besteht aus dem Verkauf der Appliance-Produkte über Partner an Endkunden. Hinzu kommen Erlöse aus Wartungsverträgen und Supportanfragen der Partner und evtl. Updates³⁸. Zusätzliche Erlöse können aus Partnerschaften mit Softwarelieferanten kommen, deren Produkte in die Appliances integriert werden (Vertriebsprovision).

³⁸ Bei TV-Appliances (digitaler Videorekorder) existieren auch Subskriptionsmodelle, bei denen eine Art „intelligente digitale TV-Zeitschrift“ abonniert wird.

4.3.3.3 Ressourcenfokus

Das Geschäftsmodelle der Appliance-Hersteller schließt aus Sicht des Autors die Teile Forschung/Entwicklung, Dokumentation, Packaging und Marketing/Vertrieb ein:

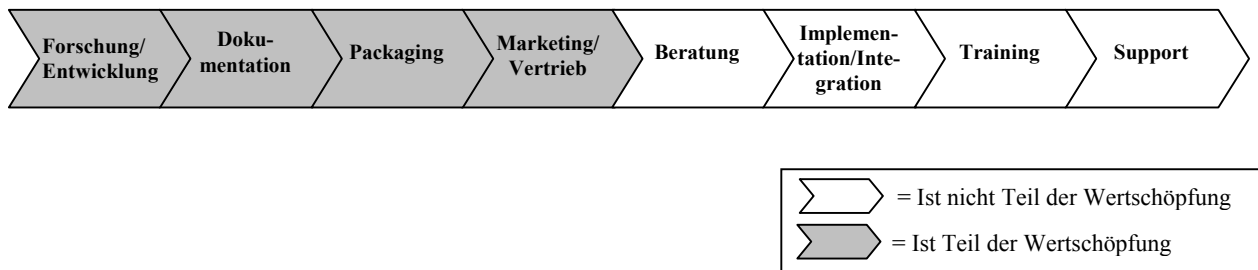


Abbildung 16: Wertschöpfungskette der Appliance-Anbieter

Partner der Appliance-Hersteller sind z. B.

- Entwicklungspartner, die Software zuliefern (OSS-Entwickler oder Unternehmen)
- Marketing- und Vertriebspartner (Großhändler, Hardwarehersteller, Einzelhändler)
- Dienstleistungspartner (kleinere Systemhäuser und Integratoren)

4.3.3.4 Strategische Absicherung

Die strategische Absicherung erfolgt über den Preis und die installierte Basis. Appliances sind Massenprodukte. Große Volumina sind entscheidend. Eine große installierte Basis sichert wiederkehrende Umsätze über Updates, neue Appliances und Partner-Wartungsverträge.

Für Appliances gelten ähnliche Bedingungen wie bei den OSS-Distributionen. Der Appliance-Markt ist relativ preissensitiv, da es sich um einen Massenmarkt handelt. Bessere Einkaufskonditionen für die Hardware durch größere Volumina ermöglichen einen Marktvorteil. Ebenso entscheidend ist die Qualität der Vertriebskanäle. Je mehr (und bessere) Vertriebspartner ein Hersteller hat, desto besser sind seine Absatzchancen. Hier sind große Unternehmen mit bestehenden Handelsbeziehungen im Vorteil, da größerer Vertriebsdruck erzeugt werden kann.

Bedrohungen für OSS-Appliances sind:

- proprietäre Appliances (Windows-Appliances)
- individuell konfigurierte Softwarelösungen (internes Projekt mit OSS- oder proprietärer Software)
- Dienstleistungsangebote von Systemhäusern und Integratoren

Diese Appliance-Alternative ermöglichen alle ähnliche Leistungen. Im ersten Fall sind diese wahrscheinlich teurer, da der Kunde neben der Hardware auch Lizenzen für die proprietäre

Software erwerben muß. Im zweiten Fall erarbeitet der Kunde eine individuelle Lösung, wobei Projektkosten anfallen. Im dritten Fall wird die Leistung durch eine individuelle Lösung eines Dienstleisters erbracht, bei der wiederum Projektkosten entstehen.³⁹

Auch Microsoft ist ein Wettbewerber, da mit der zunehmenden Integration von Komponenten in das Betriebssystem ebenfalls Funktionsbreite und Bedienkomfort erhöht werden.

Ähnlich wie bei den Distributoren existiert das theoretische Risiko eines Boykotts durch die OSS-Community (Kapitel 4.3.1.4), es sind aber keine Fälle bekannt.

Im Hardwarebereich existiert kaum Druck durch die Zulieferer, da die Hardware standardisiert ist. Ausnahme könnten Spezialkomponenten wie Decoder-Chips bei Pay-TV-Komponenten sein.

4.3.3.5 Organisation, Mitarbeiter und Kultur

Die Prozesse entsprechen auch hier den üblichen Softwareentwicklungsprozessen. Es ist keine globale Organisation notwendig, da der Vertrieb und die Vor-Ort-Installation durch Partner vorgenommen wird.

4.4 Dienstleistungs-Geschäftsmodell

Die bisher beschriebenen Geschäftsmodelle sind Produkt-Geschäftsmodelle. Bei ihnen liegt der Schwerpunkt auf der Entwicklung eines Softwareproduktes, der Freigabe unter einer OSS-Lizenz sowie dem Vertrieb und dem Vermarkten des Produktes.

Bei einem Dienstleistungs-Geschäftsmodell wird kein eigenes Produkt entwickelt, sondern es werden Dienstleistungen für existierende OSS-Produkte angeboten (Beratung, Integration, Installation, Support, Schulungen).

Das OSS-Dienstleistungsgeschäftsmodell hat sich zum „kleinsten gemeinsamen Nenner“ der OSS-Geschäftsmodelle entwickelt. Fast alle Geschäftsmodelle rund um OSS haben (auch) einen Dienstleistungsanteil. Unterschiedlich ist vor allem die Angebotstiefe und -breite: Sie kann vom einfachen Email-Support bis zur kompletten Dienstleistungspalette reichen.

Von allen OSS-Geschäftsmodellen ist das Dienstleistungsmodell das unumstrittenste, weil es herkömmlichen Geschäftsmodellen sehr ähnlich ist. Vergleicht man klassische IT-Dienstleister rund um proprietäre Software wie z. B. Accenture oder IBM Global Services mit OSS-Dienstleistern, sieht man faktisch wenige Unterschiede. In der Konsequenz ist es egal, ob man Beratung für betriebswirtschaftliche Standardsoftware oder OSS-basierte Firewalls anbietet. Die dahinterliegenden Prozesse und Mechanismen sind nahezu identisch.

³⁹ Im privaten Bereich existieren praktisch keine Dienstleistungsangebote, da die Kunden zu preissensitiv sind (vermutlich würde kein normaler Privatkunde einen Dienstleister beauftragen, eine Lösung für einen individuellen digitalen Videorekorder zu erarbeiten).

Ein Unterschied zwischen proprietären- und OSS-Dienstleistern besteht allerdings in der fehlenden Herstellerbeziehung. Viele proprietäre Dienstleister sind Partner von großen Softwareunternehmen. Diese binden die Dienstleister in Form von Partnerprogrammen an sich (Unterstützung in Form von Schulungen, Marketing und ermäßigten Lizenzen). Dies ermöglicht ein Zusatzgeschäft durch Softwarelizenzhandel. Ein typisches Beispiel für dieses Geschäft sind die Partner von großen ERP-Herstellern wie SAP, die eng mit dem Hersteller zusammenarbeiten und einen Teil ihres Umsatzes mit Lizenzhandel erzielen⁴⁰. Dieses Zusatzgeschäft fehlt OSS-Dienstleistern komplett. Natürlich könnten auch sie theoretisch nach dem o.g. Muster Partner von proprietären Softwareanbietern werden, allerdings verlassen sie damit den Markt der reinen OSS.

Die OSS-Dienstleister nehmen zusätzlich eine Qualitätssicherungsfunktion für die OSS-Komponenten ein. Dabei muß der Dienstleister testen, korrigieren und mit OSS-Entwicklern kommunizieren, was die Kosten zusätzlich erhöht.

IDA/Unisys (2000) kategorisiert die OSS-Dienstleister in vier Gruppen:

- Distributoren: Deren primäres Geschäftsmodell ist der Vertrieb einer OSS-Distribution. Zusätzlich bieten sie als sekundäres Geschäftsmodell auch Support für ihre Distribution an. In zunehmenden Maße bewegen sich die Distributoren dabei aus dem Endkunden-Markt in den Markt für Unternehmenskunden, denen sie u.a. jährliche Wartungsverträge mit garantierten Leistungen (z. B. definierte Reaktionszeiten) und allgemeine Dienstleistungen bei der Installation und Implementation anbieten.
- Große Hardwarehersteller: Diese verkaufen Hardwaresysteme mit vorinstallierten OSS-Distributionen (hauptsächlich Linux), aber vertreiben keine eigene OSS-Distribution. Hardwarehersteller haben einen großen Kundenkreis, den sie u.a. mit ihren proprietären Unix-Systemen gewonnen haben. Von der Erfahrung und Reputation in diesem Bereich können sie auch im OSS-Bereich profitieren. Hauptzielgruppe der Unternehmen sind mittlere bis große Unternehmen und Verwaltungen. Beispiele sind u.a. IBM, Compaq, HP.
- Globale Systemintegratoren: Sie bieten weltweit IT-Dienstleistungen im proprietären-, zunehmend aber auch im OSS-Bereich an. Sie fokussieren sich mit ihrer internationalen Organisationsstruktur auf große, komplexe IT-Projekte in bestehenden IT-Infrastrukturen und bieten dort die komplette Palette von IT-Dienstleistungen an. Teilweise binden diese Unternehmen Subunternehmen als Dienstleister mit ein. Beispiele sind IBM, Unisys, Accenture, CSC, Cap Gemini, PWC.
- Spezialisierte OSS-Dienstleister: Diese Unternehmen konzentrieren sich vollständig auf Dienstleistungen für OSS-Komponenten. Sie sind in der Regel kleinere Unternehmen aus dem Startup-Bereich. Sie zeichnen sich durch hohe Dynamik und eine offene Unternehmenskultur aus, sind aber auch weniger erfahren im Management großer IT-Projekte. Diese Unternehmen positionieren sich als herstellerunabhängig und unterstützen in der Regel mehrere OSS-Distributionen (z. B. mehrere Linux-Distributionen und Free BSD). Beispiele für spezialisierte OSS-Dienstleister sind Mind (Belgien), Alcove (Frankreich, Deutschland), Arkane, Atrid, Aurora, Linbox etc. (Frankreich), Thiesen, Abas Software, Sernet, Linux Partner (Deutschland).

⁴⁰ Viele SAP-Partnerunternehmen sind förmlich „mit SAP groß geworden“.

4.4.1.1 Marktpositionierung

OSS-Dienstleister bewegen sich im Markt für IT-Dienstleistungen. Zielgruppe sind in der Regel IT-Entscheider in Unternehmen, keine Privatpersonen.

Dienstleister bieten kundennahe Wertschöpfungsschritte wie Beratung, Analyse, Konzeption, Integration, Netzwerkadministration, Sicherheitskonzepte, Implementation, Support, remote Management (Wartung von Systemen über Telefonleitung oder das Internet) und Training an.

Diese Angebote können teilweise auch einen Produktcharakter besitzen (z. B. bestimmte Supportangebote, die in Form von Wartungsverträgen zusammengefaßt und mit einer garantierten Reaktions- und Behebungszeit kombiniert werden). Diese Quasi-Produkte haben für den Dienstleister den Vorteil, daß regelmäßige, planbare Umsätze erzielt werden.

IDA/Unisys (2000) unterteilt den Markt für Support-Produkte in vier Kategorien:

- Installationssupport: Die Distributoren gewähren beim Kauf einer Distribution für eine bestimmte Zeit kostenlosen Installationssupport. Mandrake bietet z. B. 60 Tage Expert-Support plus 30 Tage Telefonsupport und online-Support durch eine Web-Datenbank an. Weiterhin bietet Mandrake Unterstützung für Dienstleistungspartner. SuSE differenziert das Supportangebot nach dem erworbenen Distributionsprodukt (personal: 60 Tage Support ab Verkaufsdatum, professional: 90 Tage). Wenn eine neue Distribution erscheint, werden Fragen zu einer älteren Version nur noch begrenzt entgegengenommen⁴¹. Weiterhin bietet SuSE eine umfangreiche webbasierte Supportdatenbank an.
- Supportpakete: Distributoren bieten Support-Pakete in Form von „incident packs“ und „call packs“ an, in denen eine Anzahl von Vorfällen oder Anrufen zusammengefaßt sind. Die Leistungen werden nach und nach abgerufen.
- Wartungsverträge: In diesen wird für eine gewisse Zeitspanne (3, 6, 9 oder 12 Monate) eine bestimmte Leistung durch den Anbieter garantiert. Die Bandbreite reicht dabei von einfacher telefonischer Erreichbarkeit während üblicher Büroarbeitszeiten bis hin zu voller 24x7-Unterstützung mit garantierten Reaktionszeiten. Wartungsverträge werden vor allem an Unternehmens- und Verwaltungskunden verkauft.
- Integrationsleistungen: Allgemeine OSS-Dienstleister bieten über die einfache Hilfestellung hinaus auch komplexere Unterstützung für OSS an. Diese beinhaltet z. B. strategische Beratung, Entwicklung und Integration in bestehende Systemumgebungen.

Die Vermarktung von Dienstleistungen unterscheidet sich von der der Produkthersteller. Da Vertrauen und Kompetenz im Vordergrund stehen, sind Anzeigen weniger geeignet. Dienstleister setzen auf Kompetenzdarstellung im Rahmen von Konferenzen, Messen oder in Fachartikeln und „Mund-zu-Mund-Propaganda“.

⁴¹ Dies stellt eine (schwache) strategische Absicherung dar, weil es die Kunden zu Updates zwingt.

Dienstleister haben in der Regel einen Direktvertrieb mit direkter Kundenansprache durch Vertriebsmitarbeiter. Der Aktionsradius ist deswegen auf die Unternehmensstandorte begrenzt. Es gibt keinen Multiplikatoreffekt durch andere Vertriebspartner (Ausnahme: Subunternehmeraufträge).

4.4.1.2 Gewinnmuster

Das Gewinnmuster im Dienstleistungsbereich besteht aus dem Verkauf von Personensätzen (überlicherweise Stunden-, Tages- oder Monatssätze). Dabei ist der Auslastungsgrad der Mitarbeiter entscheidend, denn die Fixkosten (Personalkosten, Infrastrukturkosten usw.) fallen unabhängig von der Auslastung an.

4.4.1.3 Ressourcenfokus

Aus Sicht des Autos können die Wertschöpfungsschritte der OSS-Dienstleister wie folgt beschrieben werden:

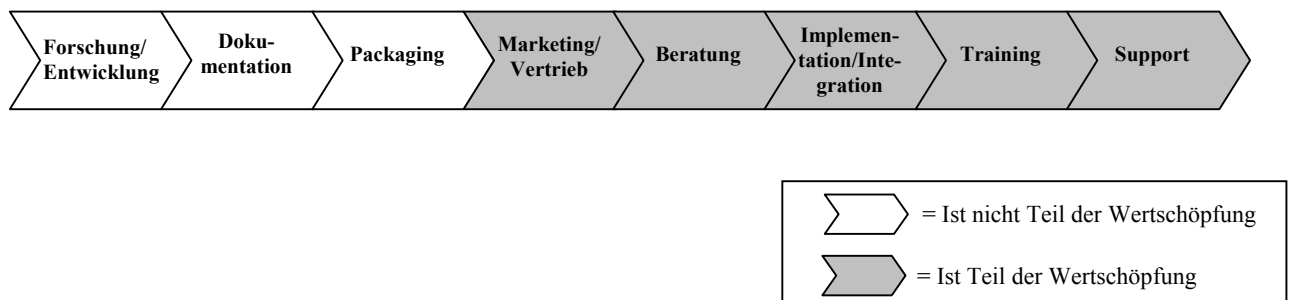


Abbildung 17: Der Ressourcenfokus der OSS-Dienstleister

Mögliche Partner der Dienstleister sind:

- Technologie/Entwicklungspartner (Unternehmen bzw. OSS-Entwickler, die Technologie zuliefern)
- Dienstleistungspartner (Unternehmen, die der Dienstleister als Subunternehmer beschäftigt oder die den Dienstleister als Subunternehmer beschäftigen)
- Vertriebspartner (Unternehmen, die Dienstleistungen und Quasi-Produkte wie Wartungsverträge vermitteln)

4.4.1.4 Strategische Absicherung

OSS-Dienstleister befinden sich in einem stark fragmentierten Markt mit vielen Anbietern. Die Beziehung Kunde-Anbieter basiert auf Vertrauen und Kompetenz. Vertrauen und Kompetenz sind allerdings nicht schützbar, sondern können von jedem Anbieter aufgebaut werden. OSS-Dienstleister können versuchen, durch Beschäftigung- und Unterstützung von prominenten

Open Source-Entwicklern oder der Förderung von OSS-Projekten eine besondere Kompetenz im OSS-Bereich aufbauen, die sie von anderen Wettbewerbern unterscheidet.

Die Markteintrittsbarrieren im Dienstleistungsbereich sind sehr gering. Eindringlinge sind Dienstleister von proprietären Betriebssystemplattformen (Unix, Windows) auf der einen Seite und IT- und Strategieberater wie accenture oder McKinsey auf der anderen Seite. Wächst der OSS-Markt weiter, werden auch diese den OSS-Dienstleistungsmarkt betreten. Etablierte Dienstleister haben dabei den Vorteil, daß sie über weltweite Erfahrung und bestehende Kundenbeziehungen verfügen.

Die Kunden haben im Dienstleistungsbereich eine relativ große Macht. Im Gegensatz zum Produktgeschäft haben Dienstleister weniger Kunden mit höheren Umsätzen pro Kunde. Die Dienstleister gehen oft in Vorleistung, bevor Umsätze erzielt werden. Bei Auseinandersetzungen über die Arbeitsleistung kann es sein, daß ein Kunde nicht bezahlen will (oder z. B. insolvent wird) und dadurch Mehraufwand oder Ausfälle entstehen.

Mögliche Bedrohungen für die OSS-Dienstleister sind die immer leistungsfähigeren Betriebssysteme und Softwareangebote. Dienstleistungen können nur verkauft werden, wenn der Kunde eine Leistung nicht selber erbringen kann oder will. Wird Software immer einfacher bedienbar, so daß sie ohne Spezialwissen eingerichtet werden kann, sinkt der Bedarf an fremder Dienstleistung (Microsoft-Strategie).

Andere Bedrohungen sind Appliances (Kapitel 2.6.1.3 und 4.3.3), die Teile der Wertschöpfungskette in einfach zu bedienende Komplettprodukte zusammenfaßt und damit in Konkurrenz zu den Dienstleistungen treten.

Für die Marktmacht der OSS-Zulieferer gilt das gleiche wie bei den Distributoren. Theoretisch sind Boykottmaßnahmen der OSS-Entwickler denkbar, in der Praxis sind solche Fälle nicht bekannt.

4.4.1.5 Organisation, Mitarbeiter und Kultur

Die Organisation der OSS-Dienstleister ist auf Dienstleistungsprozesse für Kunden ausgerichtet (Beratung, Systemadministration, Projekcontrolling etc.). Die Kunden stehen im Vordergrund, deshalb unterscheidet sich die Kultur von Softwareentwicklungsunternehmen. Im Dienstleistungsbereich benötigen die Mitarbeiter neben Fachwissen auch „soft skills“.

Der Grad der Auslastung ist für die Profitabilität entscheidend. Deshalb müssen die Prozesse straff organisiert und durch geeignete Maßnahmen wie z. B. Zeiterfassung und Projektmanagement kalkulierbar, überprüfbar und abrechenbar sein.

4.5 Mediator-Geschäftsmodell

Mediatoren bringen verschiedene Interessensgruppen im Umfeld von OSS über einen Marktplatz zusammen (Entwickler, Nutzer, Dienstleister, Werbetreibende).

Der bekannteste OSS-Mediator ist Sourceforge, ein Tochterunternehmen des Appliance-Herstellers VA Software. Sourceforge stellt eine technische Infrastruktur zur Verfügung, mit der Entwickler Software schreiben, ablegen, verwalten und kompilieren (in Maschinencode übersetzen) und die Entwickler miteinander kommunizieren können. Nutzer von OSS können über Projekt-Homepages Softwarepakete, Anleitungen und Dokumentationen downloaden.

Sourceforge hat 36,858 registrierte Projekte und 387,687 Nutzer (Quelle: sourceforge.net, Stand Ende März 2002).

Eine Variante des Geschäftsmodells stellte das aufgrund mangelnden wirtschaftlichen Erfolgs mittlerweile geschlossene Internetportal sourceexchange.com dar. Hier lag der Schwerpunkt auf der Vermittlung zwischen Sponsoren und Open Source-Entwicklern in Form einer „Auftragsentwicklung“ von OSS durch den Marktplatz.⁴²

Andere Anbieter sind Collab.net (kommerziell) und SourceFubar.net, Tigris.org, Savannah.gnu.org und BerliOS.de (nichtkommerziell).

4.5.1.1 Marktpositionierung

Mediatoren bewegen sich in unterschiedlichen Teilmärkten. Der primäre Markt ist der für werbefinanzierte Software-Marktplätze und -portale, der sekundäre der für IT-Dienstleistungen und Applikationen.

Die Zielgruppe der Mediatoren sind Entwickler, Nutzer, Dienstleister und Werbetreibende.

Das Angebot für Entwickler besteht aus Verwaltung, Hosting und Vermarktung von OSS-Projekten. Das Angebot für Nutzer ist die zentrale Anlaufstelle für Recherche und Auswahl von OSS-Softwarekomponenten. Das Angebot für Sponsoren und Werbetreibende ist zielgruppenspezifische Werbung, das Angebot für OSS-Dienstleister die Vermarktung und der Vertrieb der Dienstleistungen.

Die Nutzung des Mediators ist für Entwickler und die Nutzer kostenfrei. Sponsoren, Werbetreibende und Dienstleister zahlen teilweise für die Teilnahme am Marktplatz.

Marktplätze werden vor allem über das Internet vermarktet (Bannerschaltung und Mund-zu-Mund-Propaganda).

⁴² Rasch (2001) beschreibt das Instrument einer Finanzanleihe für die Entwicklung von OSS („completion bond“). Die Softwareentwicklung wird erst dann gestartet, wenn sich vorher ausreichend Zeichner für die Anleihe gefunden haben. Rasch zufolge würde dies das Finanzierungsproblem von Open Source lösen. Technisch könnte diese Idee mit Hilfe eines Mediator gelöst werden, allerdings ist dem Autor keine praktische Realisation bekannt.

4.5.1.2 Gewinnmuster

Das primäre Gewinnmodell der Mediatoren besteht darin, einen Marktplatz bzw. ein Portal zu betreiben und damit Erlöse zu erzielen.

Einnahmequellen der Mediatoren sind erstens der Verkauf von Bannern an Werbetreibende, zweitens Erlöse aus dem Vertrieb evtl. Commerce-Produkte (Bücher, Datenträger) und drittens Einnahmen durch die OSS-Dienstleister. Subskriptionsmodelle in Form von kostenpflichtigen Abonnements sind theoretisch auch denkbar, aufgrund des hohen Anteils an fremdem Content von OSS-Entwicklern aber wohl nur schwer durchsetzbar.⁴³

Da kein Geld zwischen Nutzern und Entwicklern fließt, ist kein Provisionsmodell möglich. Bei dem mittlerweile geschlossenen SourceXchange stand über die Hostingfunktionen hinaus das Organisieren von Sponsoren/Entwickler-Beziehungen im Vordergrund. Im diesen Fall fand ein Geldtransfer zwischen diesen beiden Gruppen statt. Der Portalbetreiber hätte ähnlich wie bei Auktionshäusern eine prozentuale Beteiligung an diesen Geldströmen erlösen können.

Die Erwartungen an werbefinanzierte Internet-Portale haben sich generell nicht erfüllt. Die Preise für Werbebanner sind gesunken und die erwirtschafteten Deckungsbeiträge sind zu gering, um damit ein Portal zu finanzieren (Engdegard 2001). Sourceforge und die Mutter VA Software haben mittlerweile das Geschäftsmodell grundlegend verändert und ein sekundäres Gewinnmodell entwickelt. Die hinter dem Portal liegende Software wird als kommerzielles und proprietäres (!) Softwareprodukt zum Management von Software-Entwicklungsprojekten verkauft. Der Marktplatz wird zum „showcase“ für die Software.

“The VA Linux business model has dramatically evolved. The Company is now primarily focused on its SourceForge application software business. With the new model, the Company expects revenue in the first quarter of fiscal 2002, excluding August hardware shipments, to fall in the range of \$3 to \$4 million, with a net loss from operations in the range of \$10 to \$13 million. Cash usage is expected to exceed the net loss by approximately \$6 million in this quarter, due to payments related to previously accrued restructuring charges.” (Quelle: www.vasoftware.com)

4.5.1.3 Ressourcenfokus

Der Ressourcenschwerpunkt des Portalbetreibers liegt in Auswahl, Anpassung, Installation, dem Betrieb und der Wartung des Marktplatzes. Bei den sekundären Geschäftsmodellen kommen Dienstleistungen und Softwareentwicklung hinzu.

⁴³ Es sei denn, die OSS-Entwickler würden finanziell davon profitieren, was aber vermutlich zu einem „Glaubenskampf“ unter den OSS-Entwicklern führen würde.

4.5.1.4 Strategische Absicherung

Die strategische Absicherung der Anbieter erfolgt über den Netzwerk- und Lock-in-Effekt⁴⁴: Ein erfolgreicher Marktplatz gewinnt aufgrund der hohen Anzahl an Teilnehmern zunehmende Attraktivität für neue Teilnehmer, wodurch der Marktplatz wiederum erfolgreicher wird. Je mehr Nutzer und Projekte der Anbieter hat, desto mehr positives feedback wird erzeugt (Bequemlichkeit, großes Angebot, Standardisierung). Dies wird wiederum eine steigende Benutzerzahl zur Folge haben. Die Wechselbereitschaft zu anderen Anbietern ist gering, da dort eine weitaus geringere Anzahl von OSS-Projekten und Nutzern zu finden ist. Anbieter können darüber hinaus keinen Preisangriff führen, da das Produkt ohnehin kostenfrei ist. Die Markteintrittsbarrieren für andere Mediatoren sind dadurch relativ hoch.

Die Nutzer des Portals haben eine relativ geringe Macht, da das Produkt für sie kostenfrei ist und deshalb kaum Druck auf den Anbieter aufgebaut werden kann.

Mögliche Bedrohungen für das Mediatorenmodell sind eigene OSS-Projekthomepages, verwaltet durch die OSS-Entwickler (bevor die Mediatoren aufkamen, war das üblich). Da dies allerdings Aufwand und Kosten verursacht und funktionell nicht an das Mediatoren-Angebot heranreicht, ist es fraglich, ob die Entwickler diesen Aufwand auf sich nehmen.

Bei den Distributor- und Appliance-Geschäftsmodellen wurden bereits mögliche Boykott- oder imageschädigende Maßnahmen der OSS-Community beschrieben. Bisher sind dort jedoch keine erfolgt. Im Zusammenhang mit dem Mediatorenmodell hingegen gab es solche Aufrufe: Der Wechsel des Geschäftsmodells von Sourceforge und die Kommerzialisierung und Proprietarisierung der dahinterliegenden Softwaretechnologie hat Unmut bei den OSS-Entwicklern erzeugt. Die Lobbyorganisation „Free Software Foundation“, die insbesondere die Freiheitsaspekte der OSS vertritt, hat mit einem Boykottaufruf reagiert:

“Over the past few months the SourceForge development facility, which hosts a large number of Free Software projects, has changed its policies. Features for exporting a project from SourceForge have been removed. The implementation used to be exclusively Free Software but is now based on non-free software. Finally, VA Linux[...] has become rather underhand in their attempts to grasp exclusive control of contributors' work. SourceForge did a lot of good for the Free Software community, but it's now time to break free. [...] SourceForge is free as in free beer because it was designed this way. It was a very expensive and ephemeral gift to the Free Software community. We could resent VA Linux for such a poisoned gift. [...] . We've finished our beer, it's time to win our freedom.” (Dachary 2001)

Wie sich der Aufruf auf Sourceforge und die freien Pendanten auswirkt, bleibt abzuwarten (die von der FSF empfohlene Alternative Savannah hatte einige Monate später (Anfang 2002), nur 493 Projekte und 3981 Benutzer aufgeführt (Quelle: savannah.gnu.org).

⁴⁴ s. Zerdick 1999

4.5.1.5 Organisation, Mitarbeiter und Kultur

Die Organisation des Unternehmens ist auf den Marktplatzbetrieb bzw. auf Softwareentwicklung ausgerichtet. Es ist keine globale Organisation und keine vor-Ort-Dienstleistung durch Personen notwendig.

4.6 Sonstige Geschäftsmodelle

Der folgende Abschnitt beschäftigt sich mit sonstigen OSS-Geschäftsmodellen. Sie sind an dieser Stelle aufgeführt, weil sie nur am Rande mit OSS zu tun haben oder Kombinationen bzw. Variationen der oben beschriebenen Modelle darstellen.

Embedded OSS

Geschäftsmodelle für Embedded-OSS sind im wesentlichen Mischungen aus dem Distributions- und dem Dienstleistungsmodell und werden deshalb an dieser Stelle nur kurz skizziert. Die Distributionen sind speziell für den Einsatz auf kleine Hardwareumgebungen (kleine CPU, wenig Speicher, Echtzeitumgebung) ausgelegt und optimiert. Cook (2000) beurteilt Linux aus folgenden Gründen positiv im Vergleich zu den typischerweise verwendeten „Eigenbau“-Betriebssystemen („home grown“): Linux ist standardisiert und Open Source. Es ist modular aufgebaut, kann auf spezifische Hardwareumgebungen angepasst werden, ist preiswert und die Kompetenzen auf Entwicklerseite sind zahlreich vorhanden. Als nachteilig sieht er den vergleichsweise großen Umfang des Linux-Kernels, der relativ große Anforderungen an die Hardware stellt. Weiterhin bemängelt er die fehlende Echtzeitfähigkeiten, Treiber und Herstellersupport.

Die meisten Embedded-Distributionen werden unter OSS-Lizenzen vertrieben. Zusätzlich bieten die meisten Hersteller umfangreiche Services im Embedded-Bereich wie z. B. die Anpassung der Distribution auf spezielle Zielsysteme oder die Wartung von installierten Systemen an.

Beispiele für Embedded-Unternehmen sind:

	System	Kostenfrei	Modifikation möglich?	Freie Verteilung der Modifikation
Lineo	Linux	System ja, Tools nein	Ja	Lizenzabgaben, wenn proprietär
Montavista	Linux	Ja	Ja	Ja
Lynx	Linux	Nein	Ja	Ja
Red Hat	Anderes	Ja	Ja	Ja
FreeBSD	BSD	Ja	Ja	Ja
BSDI	BSD	Nein	Ja	Lizenzabgaben,

Tabelle 7: Embedded-Unternehmen und deren Embedded-Applikationen

Quelle: Rosenberg (2000)

Ein wichtiger Unterschied zu den OSS-Dienstleistungs- und -Distributionsgeschäftsmodellen besteht darin, daß die Kunden von Embedded-Unternehmen in der Regel keine Endanwender sind. Kunden können z. B. Hardware-Produkthersteller im Bereich Mobilfunk, Elektronik oder Automatisierungstechnik sein. Das Geschäftsmodell der Embedded-Unternehmen ist ein Business-to-Business (B2B)-Geschäftsmodell.

Vorinstallierte Hardware

Im Zuge des wachsenden OSS-Marktes haben einige Hardwarehersteller begonnen, ihre Hardwaresysteme auch mit Linux vorzinstallieren (Beispiele: Dell, IBM und Compaq). Das Angebot besteht aus PC-Systemen (Server, selten Desktop) kombiniert mit üblichen Distributionen. Aus OSS-Sicht handelt es sich um ein typisches PC-Hardware-Geschäftsmodell, es findet keine Wertschöpfung im OSS-Softwarebereich statt.

Große integrierte IT-Anbieter

Die großen Anbieter der IT-Industrie wie z. B. IBM, Compaq, HP und Sun verfolgen integrierte OSS-Strategien. Die meisten von ihnen bieten OSS-Dienstleistungen (Schulung, Support, Beratung), OSS-Applikationen und vorinstallierte Hardware an. Allein IBM hat angekündigt, in den Jahren bis 2003 \$2 Mrd. in Linux zu investieren. Die großen integrierten IT-Anbieter eint die Gegnerschaft zu Microsoft und sie opfern offensichtlich einen Teil ihrer Unix-Kunden zugunsten von Linux. Sun tut sich mit dieser Strategie etwas schwerer und hat kein so weitgehendes Engagement, da Sun stark auf ihr Solaris-Unix als Serversystem setzt.

Auffallend ist die Tatsache, daß keiner der großen IT-Anbieter bisher eine eigene Distribution entwickelt oder einen Distributor übernommen hat. Vielleicht erkennen sie, daß die OSS-Entwicklung ohne sie „natürlich“ funktioniert und ein Eingreifen in den Markt Konsequenzen haben könnte (Boycott der Entwickler, Demotivation, neuer Kampf der Unix-Anbieter).

Embedded-Appliances

In der Unterhaltungselektronik-Industrie spielt OSS eine immer stärkere Rolle. So sind z. B. die ersten PDAs und MP3-Spieler mit Linux-Betriebssystem auf dem Markt. Allerdings spielt OSS in dem Geschäftsmodell des Anbieters nur eine untergeordnete Rolle. Der Käufer wird einen MP3-Player nach Kriterien wie Preis, Leistung, Bedienung und Größe auswählen und weniger, weil im Hintergrund Linux läuft. Für den Anbieter kann OSS durchaus bedeutend sein, weil auf die Entwicklung eines eigenen Betriebssystems und eigener Applikationen (bzw. die Lizenzierung durch einen Dritten) verzichtet und der Preisvorteil an die Kunden weitergeben werden kann.

Linux-Hotel

Ein Hotelanbieter in Essen vermarktet sein Hotel als voll ausgestattetes Linux-Hotel. Gäste können einen PC-Raum mit Linux-Rechnern benutzen. Motto des Anbieters: „Linux im Linuxhotel zu lernen, ist wie Englisch in England“ (www.linuxhotel.de)

Bücher, Gimmicks

Der Fachverlag O'Reilly gilt als *der* Fachverlag für Bücher zum Thema Open Source. Eine Reihe von bekannten Autoren haben hochwertige Bücher zu verschiedensten Aspekten von OSS veröffentlicht. Der Gründer Tim O'Reilly ist ein prominentes Mitglied der OSS-Community.

Eine Reihe von Anbietern haben den wachsenden OSS-Trend erkannt und bieten alle Arten von Artikeln und Gimmicks rund um OSS an. So gilt der Linux-Pinguin als ein beliebtes Plüschtier-Vorbild, genau wie sein (seltenerer) Kollege, der BSD-Teufel. Weitere Angebote sind diverse Tassen, Gebrauchsutensilien und T-Shirts rund um dem Themenkomplex OSS.

4.7 Zusammenfassung

Der folgende Abschnitt faßt zusammen, welche OSS-Geschäftsmodelle existieren, inwieweit sich die OSS-Geschäftsmodelle bewährt haben und welche Zukunft die OSS-Geschäftsmodelle haben.

Die OSS-Geschäftsmodelle unterliegen teilweise einer Transformation. Das folgende Schaubild des Autors zeigt die OSS-Geschäftsmodelle und deren Veränderungen.

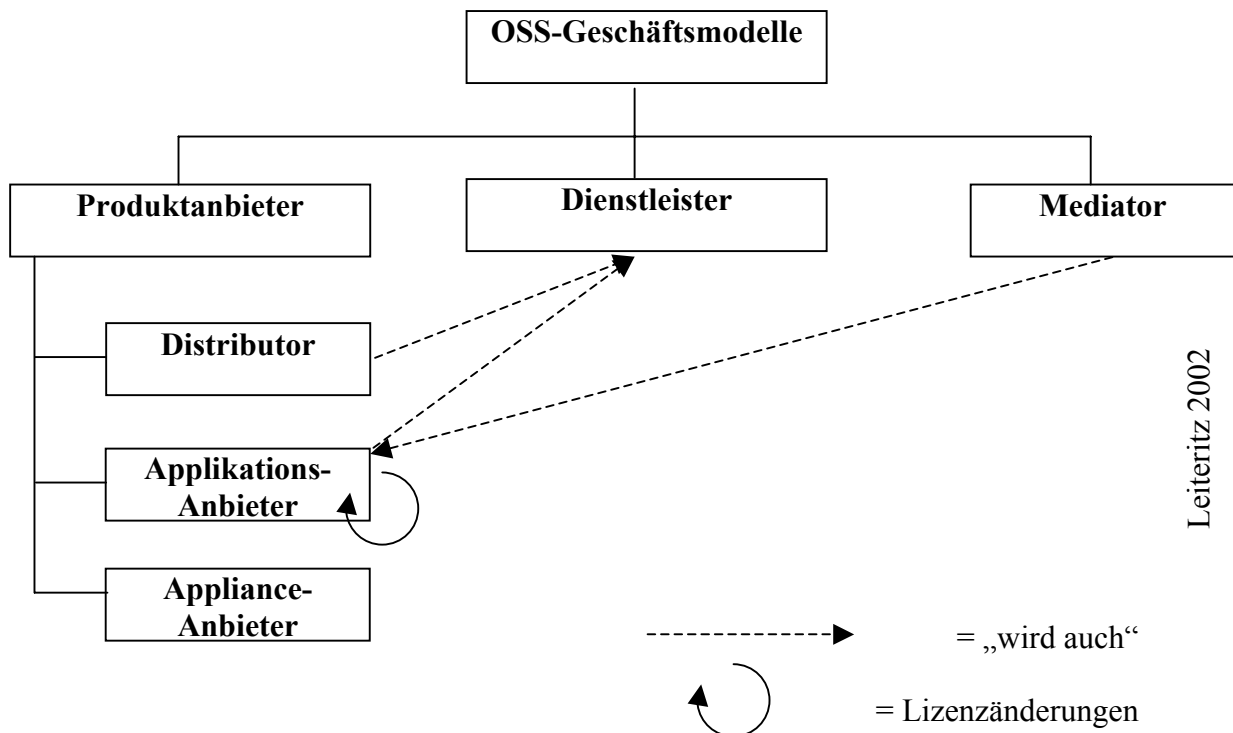


Abbildung 18: OSS-Geschäftsmodelle und deren Transformationen

Es wurde gezeigt, daß einige Geschäftsmodelle kritisch zu beurteilen sind. Aus Sicht des Autors sind das Mediator-, das Distributions- und das OSS-Applikationsmodell als primäre Geschäftsmodelle nicht überlebensfähig. Nur in Verbindung mit sekundären Geschäftsmodellen wie Dienstleistungskomponenten oder mehr oder weniger kommerziellen Softwareangeboten können die Anbieter überleben.

Einige Geschäftsmodelle müssen also als gescheitert bezeichnet werden. Die verbleibenden Geschäftsmodelle müssen ihre Tragfähigkeit beweisen:

- Die OSS-Applikations-Anbieter passen ihre Lizenzen aufgrund veränderter Marktbedingungen an. Teilweise werden OSS-Produkte rekommerzialisiert, einige Anbieter beendeten sogar ihre OSS-Aktivitäten komplett. Praktisch alle OSS-Applikations-Anbieter haben Sekundärgeschäftsmodelle, bei denen sie indirekt aus der Freigabe der Software profitieren.
- Die Distributoren entwickeln sich zunehmend (auch) zu OSS-Dienstleistern, weil der Verkauf der Distributionen offensichtlich nicht kostendeckend möglich ist.
- Die Mediatoren haben ihr ursprüngliches Geschäftsmodell teilweise aufgegeben und verfolgen sekundäre Applikations-Geschäftsmodelle.
- Das Appliance-Geschäftsmodell scheint aus Sicht des Autors prinzipiell überlebensfähig zu sein. Da es sich um Massenprodukte mit relativ niedrigen Margen handelt, müssen Anbieter große Stückzahlen absetzen. Bis heute ist es keinem Anbieter gelungen, signifikante Marktanteile mit OSS-basierten Produkten zu erreichen. Auch die Übernahme des Server-Appliance-Anbieters Cobalt durch Sun hat an dieser Tatsache offensichtlich nichts verändert (vgl. Shankland 2001b).
- OSS-Dienstleistung hat sich zum „kleinsten gemeinsamen Nenner“ der OSS-Geschäftsmodelle entwickelt. Fast jeder Anbieter ist auch Dienstleister. Da sich dieses Geschäftsmodell nur in wenigen Aspekten von „normalen“ IT-Dienstleistungsgeschäftsmodellen unterscheidet, ist dies nicht weiter überraschend. Allerdings hat das Dienstleistungsmodell eigene Herausforderungen (Auslastungsoptimierung, kein „law of increasing returns“). Die herkömmlichen IT-Anbieter werden in den OSS-Dienstleistungsmarkt eindringen und die OSS-Anbieter können sich aufgrund nicht existierender Schutzmechanismen nur mit Hilfe einer Markenstrategie verteidigen. Zusätzlich bleibt für OSS-Dienstleister das Problem, daß sie die Margen nicht durch Handel mit Softwarelizenzen aufbessern können und zusätzlich OSS-Qualitätssicherung übernehmen müssen.

Wie hat sich die kommerzielle Open Source Entwicklung vollzogen und wie wird die Zukunft aussehen?

Die kommerzielle Open Source-Entwicklung gipfelte in einer Euphorie in den Jahren 1999 und 2000. Nach den Highlights wie dem erfolgreichen Börsengang von Red Hat und VA Linux (erfolgreichste Nasdaq-Erstemission aller Zeiten) erreichte der Linux Stock Index im Januar 2000 seinen Höhepunkt.

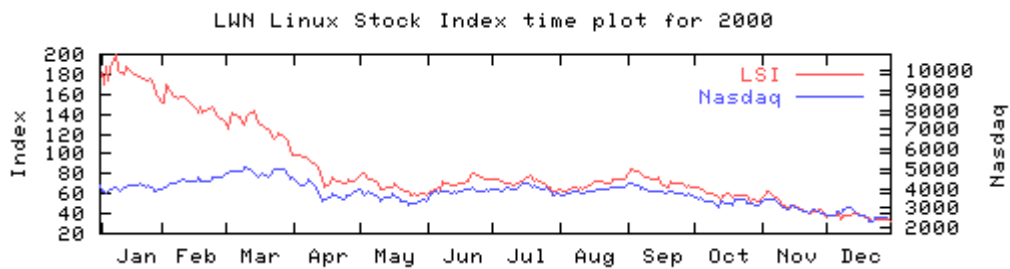


Abbildung 19: Entwicklung des Linux Stock Index 2000 im Vergleich zum Nasdaq Composite Index

LSI obere, Nasdaq untere Kurve (Quelle: www.lwn.net)

In den folgenden Monaten kam es zu einem rapiden Fall der OSS/Linux-Aktienwerte und zu einer Reihe von Massenentlassungen und Insolvenzen bei den Anbietern. Ein wichtiger Grund hierfür ist sicherlich das allgemeine Scheitern der „New Economy“ und der damit verbundenen Börsenübertreibungen. Bedeutet dies, daß die Open Source-Unternehmen die schlechten Börsenzeiten einfach „aussitzen“ können und auf eine Erholung hoffen dürfen?

Wie anhand der einzelnen Geschäftsmodelle beschrieben, wird dies nicht ausreichen. Aus dem noch sehr jungen OSS-Phänomen haben sich in den letzten Jahren eine Reihe von OSS-Geschäftsmodellen herausgebildet. Diese unterscheiden sich allerdings deutlich in ihren Mechanismen, in ihrer Nachhaltigkeit und in ihren Überlebenschancen. Viele der OSS-Geschäftsmodelle befinden sich seit Jahren in einer Transformation (beispielsweise der Anbieter VA Software, der sich vom Softwareunternehmen zum Appliance-Hersteller, dann zum Dienstleister und wieder zum Softwareunternehmen gewandelt hat). Die kommerzielle OSS-Branche befindet sich in einem „kollektiven Selbstfindungsprozeß“.

Im folgenden werden die allgemeinen Faktoren zusammengefaßt, die Einfluß auf die Tragfähigkeit von OSS-Geschäftsmodellen haben.

Open Source als Nischensystem?

Der Erfolg von OSS in den letzten Jahren ist beeindruckend. Dennoch muß man feststellen, daß sich der Erfolg auf bestimmte Marktsegmente konzentriert. Der Server-Markt und die Basistechnologie-Anwendungen rund um Internet und Intranet werden von Open Source erfolgreich abgedeckt. Im Desktop-Bereich und im „mission-critical“-Bereich für Business-Applikationen spielt Open Source dagegen praktisch keine Rolle (Weiss 2001). Sergio zitiert 2001 eine Studie von Goldman Sachs vom November 2001 über die Einsatzplanung von OSS in Fortune 1000 Unternehmen weltweit.

„About 65 percent of executives polled by Goldman Sachs said they have no plans to use Linux at their company next year. Twenty-four percent of respondents said they would use Linux as an addition to their Windows and Unix setups, but not as a replacement. And only 3 percent see Linux as their primary enterprise server system within three years; 60 percent named Windows.” (Sergio 2001, S. 2).

Für die OSS-Anbieter bleibt es folglich eine wichtige Aufgabe, die Entscheider in großen Unternehmen von den Vorteilen des Einsatzes von OSS zu überzeugen.

Markteintritt von Mitbewerbern

Wie oben ausgeführt, ist es für OSS-Anbieter schwer, sich durch wirksame Markteintrittsbarrieren vor Angreifern zu schützen. Es bleibt vor allem der Markenaufbau. Das Wachstum im OSS-Bereich haben aber auch andere Marktspieler erkannt. Für bestehende Anbieter ist Open Source nur eine weitere Plattform, während Infrastruktur, Kundenbeziehungen und Technologien bereits existieren. OSS-Newcomer müssen diese erst aufbauen. Dabei gilt: Je erfolgreicher OSS-Anbieter sind, desto mehr werden sie angegriffen.

Open Source als „bottom-up“-Phänomen

Klassische Softwareentwicklungen erfolgen „top-down“ als Lösung für ein identifiziertes Marktbedürfnis. OSS dagegen wird „bottom-up“ entwickelt: Eine technische Lösung für ein technisches Problem für die Zielgruppe Techniker. OSS entwickelte sich in den letzten Jahren nach diesem Prinzip aus einem fruchtbaren Nährboden in unterschiedlichste Anwendungsfelder, aber ohne Marktorientierung. Für OSS-Unternehmen bleibt die Herausforderung, die OSS-Innovationen in marktfähige Angebote umzusetzen.

Verbreitungs-/Kommerzialisierungs-Paradox

Bei proprietären Software-Geschäftsmodellen gilt: Je größer der Verbreitungsgrad eines Produktes oder einer Produktgattung ist, desto erfolgreicher sind die jeweiligen Anbieter. In der OSS-Welt existiert dieses Gesetz nicht. Die Verbreitung von Open Source-Software im Server-Bereich steigt (oder bleibt zumindest auf hohem Niveau konstant), während die Anbieter von OSS-Geschäftsmodellen wirtschaftliche Probleme haben. Durch die Abwesenheit von Schutzmechanismen können OSS-Unternehmen die Verbreitung von OSS nicht 1:1 an ihren wirtschaftlichen Erfolg koppeln.

Der Gartner-Analyst Weiss (2001) gibt folgende Einschätzung über die Zukunft der Linux-Entwicklung bis 2005. Er unterteilt die kommerzielle Linux-Entwicklung dabei in drei Phasen eines „hype cycles“:

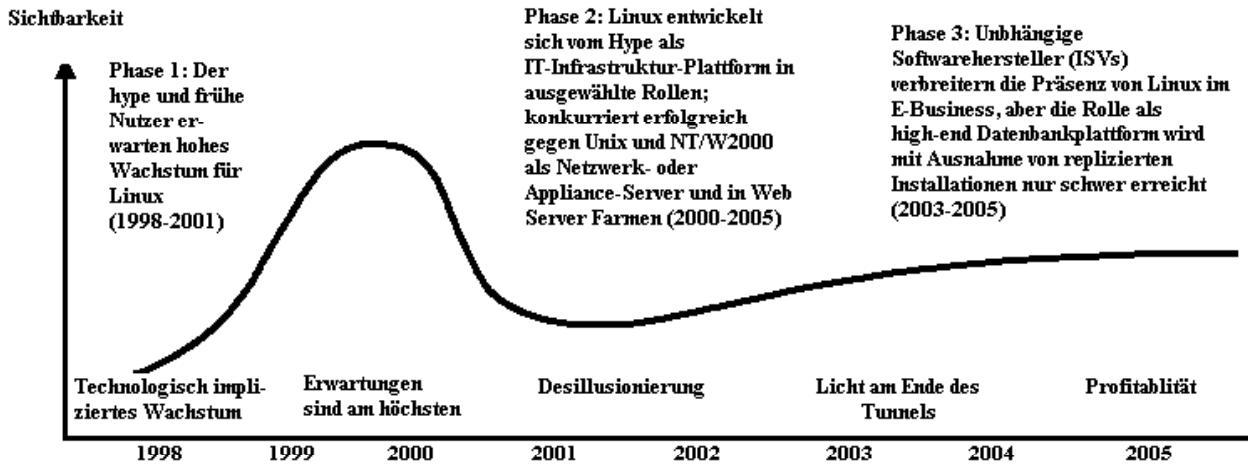


Abbildung 20: Gartner-Prognose über die Entwicklung des Linux-Marktes
Übersetzungen RL

Weiss unterscheidet dabei in den Zeitabschnitten jeweils den wichtigsten Treiber („driver“) und das wichtigste Hemmnis („inhibitor“) für die Entwicklung von Linux:

	Haupttreiber	Haupthemmnis
1998 - 1999	Server-Appliance-Geschäftsmodelle	Fehlends Vertrauen in Linux
2000	Vorteile aus den Lizenzen	fehlender Support
2001 - 2002	Ähnlichkeit zu Unix	fehlende Enterprise-Tauglichkeit
2003	Druck der Hardware-Hersteller	fehlende Fähigkeiten („skills“)
2004 - 2005	Open Source Entwicklung	fehlende Applikationen

Tabelle 8: Gartner-Prognose über die kommerzielle Linux-Entwicklung

5 Die Zukunft von OSS im kommerziellen Bereich

„The Microsoft business model will clash so severely with the new open-source-fueled development and distribution models that the company's market share will shrink for the first time in its history. And eventually, MS will become little more than a 'legacy vendor,' offering support for its antiquated products.”
Carl D. Howe, 8/2000

Ziel dieser Arbeit war es, einen Überblick über den kommerziellen Einsatz von OSS und OSS-Geschäftsmodellen zu geben. Dabei wurde zwischen dem Einsatz auf Nachfrager- und auf Anbieterseite unterschieden.

OSS und insbesondere der prominente Vertreter Linux hat sich auf Nachfragerseite in bestimmten Bereichen durchgesetzt. Linux hat einen Marktanteil von ca. 25% im Serverbereich erobert und ist dort das einzige Betriebssystem mit wachsenden Marktanteilen. Am stärksten vertreten ist OSS dabei im Bereich der Infrastrukturlösungen wie z. B. der Einsatz im Intranet, als Virtual Private Network, als Internetserver und in heterogenen Netzen für kleine und mittlere Unternehmen und Abteilungen. Hauptkonkurrent ist dabei eher Unix als Windows. Die Argumente für den Einsatz von OSS im kommerziellen Bereich sind Sicherheit, Geschwindigkeit, Offenheit und Stabilität und der Preisvorteil. Argumente dagegen sind der Mangel an Applikationen, das Desktop-Einsatzgebiet, mangelndes Vertrauen/Investitionssicherheit und die Administrierbarkeit.

Auch im Embedded-Bereich wird OSS großes Potential zugetraut, allerdings ist dieser Markt aufgrund sehr vielfältiger Endgeräte und des „in den Hintergrund tretens“ der Software sehr komplex. Auf dem Desktop spielt Linux keine Rolle und es ist auch keine positive Tendenz erkennbar. Im High-End-Bereich hat OSS mit einer starken Opposition durch Unix-Systeme zu kämpfen, aber dort ist durch stetigen technischen Fortschritt ein Aufholpotential erkennbar.

Auf Anbieterseite zeigt sich ein gemischtes Bild. Aus der Euphorie der späten 90er Jahre ist Ernüchterung geworden. Einige wenige Anbieter wie die Linux-Distributoren SuSE, Mandrake, Turbolinux und Red Hat haben es geschafft, eine erfolgreiche Marktposition aufzubauen. Von diesen Distributoren hat allein Red Hat eine globale Marke im Businessbereich aufgebaut und durch einen frühzeitigen Börsengang die Kapitalbasis für eine weltweite Expansion gesichert. Aber auch die Distributoren haben ihr Geschäftsmodell angepaßt und setzen zunehmend auf sekundäre Dienstleistungen, die so etwas wie den „kleinsten gemeinsamen Nenner“ der OSS-Geschäftsmodelle darstellen. Die Vertreter anderer Geschäftsmodelle befinden sich in einem Selbstfindungsprozeß mit unklarem Ausgang. Einige Vertreter der Dienstleistungs-, Mediatoren- und Applikationsgeschäftsmodelle mussten ihre Tätigkeit sogar ganz einstellen.

Zusammenfassend kann man festhalten, daß sich die hohen Erwartungen an die Open Source-Geschäftsmodelle nicht erfüllt haben. Es bleibt abzuwarten, welche Unternehmen die Krise der OSS-Geschäftsmodelle überleben. Es zeichnet sich jedoch ab, daß nur eine Mischform der Geschäftsmodelle, verbunden mit einer ausgezeichneten Markenpositionierung, guter Kapitalausstattung und Partnerschaften mit großen IT-Unternehmen, erfolgreich sein wird.

Die OSS-Geschäftsmodelle haben über ihre spezifischen Eigenschaften hinaus mit einer Reihe von Herausforderungen zu kämpfen: Die OSS-Anbieter haben Schwierigkeiten, nachhaltige Markteintrittsbarrieren gegen Mitbewerber zu errichten. OSS ist als „bottom up“-Phänomen technik- und nicht marktorientiert. Die Verbreitung von OSS ist aufgrund nicht existierender Schutzmechanismen nicht 1:1 mit einer positiven wirtschaftlichen Entwicklung der Unternehmen koppelbar.

Allerdings kann auch, unabhängig vom gemischten Erfolg der OSS-Anbieter, festgehalten werden, daß OSS die IT-Industrie verändert hat. Die Vorteile von OSS haben zu einem erhöhten Verbraucherbewußtsein, einer kritischen Auseinandersetzung mit Monopolen und zu einer Hinwendung zu offenen Standards geführt. Insofern sind die Vorteile und die Zukunft von OSS ein Stück weit unabhängig vom kommerziellen Erfolg der Anbieter zu sehen.

Einer gesonderten Erörterung bedarf über diese Arbeit hinaus die Rolle der IT-Industrie in Bezug auf OSS. Insbesondere das Engagement von IBM mit bisher über \$2 Milliarden und einer Vielzahl von integrierten Projekten und Angeboten rund um OSS geht in der Dimension über die in dieser Arbeit vorgestellten Geschäftsmodelle hinaus. Die Vorgehensweise von IBM zeigt die Bedeutung von Linux als „vereinigende Technologie“ der Microsoft-Gegner. Diese Bestrebungen heben Linux und Open Source auf die Ebene eines neuen Standards gegen die proprietäre Softwareentwicklungswelt.

Ein weiterer lohnender Aspekt wäre eine Untersuchung der volkswirtschaftlichen Implikationen von OSS. Zum Beispiel wäre es interessant, zu untersuchen, welche Rolle OSS für den europäischen Wirtschaftsraum spielt und inwieweit Open Source die Innovation in der Softwareindustrie fördert. Darüber hinaus könnte eine interessante Frage sein, welche Auswirkungen OSS auf die IT-Arbeitswelt hat.

6 Abkürzungen

AT&T	American Telephone and Telegraph Company. Amerikanisches Unternehmen für Fernmelde- und Kommunikations-einrichtungen
Bell Labs	Bell Telephone Laboratories. Forschungseinrichtungen von AT&T
BIND	Berkeley Internet Name Daemon
BSD	Berkely System Distribution
BSI	Bundesamt für Sicherheit in der Informationstechnik
Bzw.	beziehungsweise
ca.	circa
DNS	Domain Name System
EDV	Elektronische Datenverarbeitung
etc.	et cetera
ff.	fortfolgende
FSF	Free Software Foundation
GCC	GNU C Compiler
GNU	GNU is not Unix
GPL	GNU General Public License
IBM	International Business Machines
IDC	International Data Corporation
IT	Informationstechnologie
LAN	<i>LAN</i> Local Area Network
OS	Operating System
OSS	Open Source Software
PC	Personal Computer
s.	siehe
S.	Seite
u.a.	unter anderem
u.ä.	und ähnliche
Vgl.	vergleiche
z. B.	zum Beispiel
z. T.	zum Teil

7 Glossar

Appliances: Applikations/Betriebssysteme/Hardwarekombinationen, die für einen konkreten Zweck entwickelt wurden und einfach bedienbar sind.

Applikation: Das ausführbare Programm und alle Dateien, die zur Ausführung eines Programms gebraucht werden.

Beta-Test: Test einer Software, die sich noch im Entwicklungsstadium befindet, aber bereits in einer Version vorliegt, in der Qualität und Funktionsumfang zuverlässig überprüft werden können. Nach Durchführung des Beta-Tests werden die Ergebnisse in die Software eingearbeitet und diese in der Regel anschließend auf den Markt gebracht.

Betriebssystem: Grundlegende Programmsammlung, welche die Nutzung eines Computers erst ermöglicht. Es stellt die Verbindung zwischen Ein- und Ausgabegeräten her und verwaltet Hardware.

Bottom Up: Lösung von abgegrenzten Teilproblemen, anschließend mit deren Hilfe Lösung von übergeordneten Problemen.

Browser: Programm zum Navigieren im WWW.

Bug: Fehler, Programmfehler.

Bugfix: Fehlerbehebung.

Client: Rechner, der von einem anderen Rechner (Server) Programme, Dienste etc. in Anspruch nimmt.

Compiler: Programm, das Quellcode in Maschinensprache konvertiert.

Code: Bezeichnung für Binärcode von Programmen.

CPU: „Central Processing Unit“. Die zentrale Einheit des Computers, die Rechen- und Steueroperationen ausführt.

Desktop: Arbeitsfläche fensterorientierter Benutzeroberflächen.

Download: Der Vorgang des Herunterladens von Daten von einem anderen Rechner über das Internet.

E-Commerce: „Electronic Commerce“. Abwickeln von Geschäftsvorgängen über das Internet.

Email: „Electronic Mail“. Das elektronische Versenden von Daten, Nachrichten und Texten.

FAQ: „Frequently Asked Questions“. Fragen und Antworten von Anwendern in Online-Medien.

Fileserver: Dateiserver.

Freeware: Ursprünglich der Begriff für freie Software, deren Quellcode offengelegt und für jedermann erhältlich ist sowie ohne Einschränkungen verändert, kopiert und weitergegeben werden darf. Im Alltagsgebrauch wird unter Freeware Software verstanden, die entgeltfrei benutzt werden darf, wobei der Quellcode nicht offengelegt wird und Modifikationen nicht erlaubt sind.

FTP: „File Transfer Protocol“. Protokollverfahren im Internet, das den Datenaustausch auch zwischen unterschiedlichen Computersystemen erlaubt.

Grafische Benutzeroberfläche: Bedienungsumgebung einer Anwendung oder eines Programms, die Symbole oder Icons benutzt, um den Benutzer Eingaben machen zu lassen.

Homepage: Innerhalb des WWW die Seite einer Person, Organisation oder eines Unternehmens, die bei Zugriff auf dessen Web Site zuerst erscheint.

Internet: Der weltweite Rechnerverbund, der nahezu universellen Daten- und Informationsaustausch sowie das Versenden von E-Mails ermöglicht.

Intranet: Netzwerk in Unternehmen, das den Mitarbeitern intern die Funktionalitäten des Internet zur Verfügung stellt (Kalender, Email etc.).

IP-Adresse: „Internet-Protocol-Adresse“. Eine Nummer, die einen Rechner im Internet eindeutig identifiziert.

Kernel: Der Teil des Betriebssystems, der u. a. die Systemressourcen verwaltet.

kompilieren: Die Herstellung eines Programms in Maschinensprache durch einen Compiler.

Law of increasing returns: Effekt, daß ein Produkt durch hohe Verbreitung wiederum bevorzugt wird und dadurch ein positives Feedback erzeugt wird.

Lock-In: Zum Lock-In (Einsperrungseffekt) kommt es, wenn ein Vertragspartner größere spezifische Investitionen getätigt hat als der andere. Er ist dann in das Vertragsverhältnis oder in die gewählte Technologie innerhalb gewisser Grenzen „eingesperrt“.

Mainframe: Großrechner.

Minix: Ein zu Lehrzwecken entwickelter, minimal ausgestatteter Unix-Kernel.

MS-DOS: Von der Firma Microsoft entwickeltes DOS-Betriebssystem.

Netzwerk: Struktur aus Hard- und Software, die den Verbund räumlich entfernter Computer ermöglicht.

Newsgroup: Forenplatz für Diskussionen im Internet.

Patch: Gesondertes Programm, das erzeugt wird, wenn sich ausgelieferte Software

als fehlerhaft oder verbesserungswürdig erweist.

PDA: „Personal Digital Assistant“. Kleiner, tragbarer Computer.

Peer Review: Überprüfung durch Fachkollegen.

Performance: Oberbegriff für die Leistungsfähigkeit eines Rechners oder einer Rechnerkonfiguration.

Portabilität: Eigenschaft von Programmen, auf unterschiedlichsten Rechnersystemen lauffähig zu sein, ohne daß an den Programmen Änderungen vorgenommen werden müssen.

Proprietär: Software, die nicht im Quellcode offenliegt.

Public Domain Software: Software, die vom Urheber entgeltfrei zur Verfügung gestellt wird.

Quelltext/-code: Programmtext, der von Menschen in einer Programmiersprache verfaßt wurde und durch Kompilieren von einem Rechner ausführbar wird.

Release: Veröffentlichung von Versionen einer Software.

Server: Rechner, der in Netzwerksystemen anderen Rechnern (Clients) Anwendungen zur Verfügung stellt.

Shell: Der Befehlsinterpreter von Unix Systemen. Dient zur Interpretierung und Ausführung von Kommandos.

Solaris: Auf Unix basierendes Betriebssystem der Firma Sun.

Standard: Eine vom Produzenten stillschweigend oder auf der Grundlage eines formalen Übereinkommens eingehaltene technische Spezifikation.

Top Down: Ausgehend von einem hohen Abstraktionsgrad zunehmende Konkretisierung von oben nach unten, d. h. Aufteilung einer Gesamtaufgabe in Teilaufgaben, die wiederum weiter aufgeteilt werden.

Tool: Kleineres Programm, das die Leistung einer Software erweitert oder verbessert.

Trojaner: Ein unerkannt in ein Computersystem eindringendes Programm, das den befallenen Rechner ausspioniert oder fernsteuert.

Unix: Professionelles Netzwerkbetriebssystem.

Update: Neue Version von Softwarepaketen oder –programmen.

Windows NT/2000/XP: Microsoft-Betriebssysteme für den Einsatz in Netzwerken.

WWW: „World Wide Web“. Dienst des Internet mit grafischer Darstellung von Dokumenten.

8 Literaturverzeichnis

Ardal, Atila: „Open Source – das Beispiel Linux. Ökonomische Analyse und Entwicklungsmodell eines erfolgreichen Betriebssystems“. Diplomarbeit, 2001
ig.cs.tu-berlin.de/da/059/ardal-opensource.pdf

Balzert, Helmut: „Lehrbuch der Software-Technik: Software-Entwicklung“. Spektrum, Akademischer Verlag, Heidelberg 1996

Balzert, Helmut: „Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung“. Spektrum, Akademischer Verlag, Heidelberg 1998

Baumgartner, Jens D.: „Deutsche Übersetzung der ‚Open Source Definition‘“. www.wdrcc.de/sendungen/report/informationen/20000429/Thema01.htm

Bessen, James: „OSS: Free Provision of Complex Public Goods, Research on Innovation, Wallingford 2001
www.researchoninnovation.org/opensrc.pdf

Bezroukov, Nikolai: „A second look at the Cathedral and the Bazaar“. First Monday, Vol. 4, No. 12, December 1999
www.firstmonday.dk/issues/issue4_12/bezroukov/index.html

Bezroukov, Nikolai: „OSS Development as a special type of academic research (a critique of vulgar Raymondism)“. First Monday, Vol. 4, No. 10, October 1999
www.firstmonday.dk/issues/issue4_10/bezroukov/index.html

Blase, Paul: „Open Source White Paper - A Software Company's Dilemma“. DiamondCluster, 2000
www.diamondcluster.com/work/Wpapers/WPSoftware.asp

BMWi: „Open-Source-Software-Ein Leitfaden für kleinere und mittlere Unternehmen“. 2001
www.bmwi.de/Homepage/download/infogesellschaft/Open-Source-Software.pdf

Busa, Patricia: „‘Free’ Pricing Model - Can businesses really make money by giving goods away for free?“. University of Texas at Austin, 1999
cci.bus.utexas.edu/research/white/free-price.htm

Claybrook, Bill: „AberdeenGroup InSight: Linux is on the Move Up!“. Aberdeen Research, July 24, 2001
www.ibm.com/linux/LinuxInSight.pdf

Cimetiere, Jean-Christophe: „A New Era of Service Providers Driven by the Open Source Software (OSS) and Free Software wave“. Intranetjournal/TechMetrix Research
www.intranetjournal.com/articles/200005/asp_05_30_00b.html

Cook, Rick: „Embedded Linux Basics”. LinuxWorld.com, May 2000
www.linuxworld.com/linuxworld/lw-2000-05/lw-05-embedded.html

Dachary, Loic: „SourceForge drifting”, Free Software Foundation, 2001.
www.fsfeurope.org/news/article2001-10-20-01.en.html

Daffara, Carlo: „Free Software / Open Source: Information Society Opportunities for Europe?”. Working group on Libre Software, April 2001
eu.conecta.it/paper/paper.html

Dafermos, George N.: „Management and Virtual Decentralised Networks: The Linux Project”. First Monday, Vol. 6, No. 11, November 2001
www.firstmonday.dk/issues/issue6_11/dafermos/index.html 2001

Dibona, Chris; Ockman, Sam; Stone, Mark: „Open Sources. Voices from the Open Source Revolution”. O’Reilly, Sebastopol, CA, 1999.
www.sindominio.net/biblioweb/telematica/open-sources-html/main.html

Diedrich, Oliver. „Schnellster Webserver unter Linux ”. heise online News, 4. Juli 2000
www.heise.de/newsticker/data/odi-03.07.00-004/

Diedrich, Oliver. „Sprung auf den Desktop für Linux zu hoch?”. heise online News, 15. Mai, 2001a
www.heise.de/newsticker/data/odi-25.05.01-002/

Diedrich, Oliver. „Linux-Kernel kann Dateisystem beschädigen ”. heise online News, 25. November 2001b
www.heise.de/newsticker/data/db-25.11.01-000/

Diedrich, Oliver. „Sicherheitslücke im Linux-Kernel”. heise online News, 19. Oktober 2001c
www.heise.de/newsticker/data/odi-19.10.01-001/

de Icaza, Miguel: „Is Linux ready for the corporate desktop?”. Computerworld, August 9, 2001
www.computerworld.com/storyba/0,4125,NAV47_STO62924,00.html

Delio, Michelle: „It'll Be an Open-Source World”. wired.com, August 15, 2000
www.wired.com/news/print/0,1294,38240,00.html

Dyck, Timothy (eWEEK): „MySQL”. CNET.com, March 5, 2001
enterprise.cnet.com/enterprise/0-9513-707-6618957.html?tag=st.it.9500-704-0.sr.9513-717-6716336-7053536

Edlbauer, Florian: „Studie: 75 Prozent lehnen Linux ab“. Zdnet.de-News, 24. März 2001.
news.zdnet.de/story/0,,s2056048,00.html

EITO: „European Information Technology Observatory”. 2001

EDC (Evans Data Corporation): „Embedded Systems Developer Survey 2001“. August 2001

www.evansdata.com

Engdegard, Staffan: „Jupiter MMXI: Preise für Online-Werbung sinken weiter“. Juli 2001, zitiert nach

www.golem.de/0105/14072.html

Emnid: „Bevölkerungsbefragung über Nutzung und Akzeptanz des Betriebssystems Linux“. Eine repräsentative Befragung von TNS EMNID Telecommunication&IT im Auftrag der SuSE Linux AG, 24. März 2001

www.suse.de/de/press/press_releases/archive01/misc/emnid_studie/index.html

Everitt, Paul: „How we reached the Open Source Business Decision“. Digital Creations, 2001.

www.zope.org/Members/paul/BusinessDecision

Farbey, Barbara; Finkelstein, Anthony: „Evaluation in Software Engineering: ROI, but more than ROI“. Paper submitted for „Third International Workshop on Economics-Driven Software Engineering Research“ (EDSER-3 2001), 2001

www.cs.virginia.edu/~sullivan/edser3/finkelstein.pdf

Forschungsstelle für Verwaltungsinformatik, Universität Koblenz-Landau: „Protokoll der Veranstaltung ‘OSS (OSS) in der Bundesverwaltung’ der Koordinierungs- und Beratungsstelle für Informationstechnik in der Bundesverwaltung, 26.-28. September“. 2000

linux.kbst.bund.de/auftakt/ergebnis/oss-protokoll.pdf

Gallist, Rudi: „Offener Brief an den Linux-Verband“ im Rahmen der Diskussion über die IT-Ausstattung des Deutschen Bundestags. Verband der Softwareindustrie Deutschlands, 2002.

www.vsi.de/inhalte/spezial/spezial.asp?cid=Spezial&id=20

Galli, Peter: „Linux takes a leap and a stumble“. Zdnet News, 20 Mai 2001.

zdnet.com.com/2100-11-503728.html?legacy=zdnm

FSF/GNU: „Categories of Free and Non-Free Software“. gnu.org

www.gnu.org/philosophy/categories.html

Gartner Viewpoint: „Linux support services: Like any other operating system?“, 2001

techupdate.zdnet.com/techupdate/stories/main/0,14179,2808791-1,00.html

Gehring, Robert: „Software Patents - IT-Security at Stake?“. Informatik und Gesellschaft, TU Berlin, 2001.

www.ig.cs.tu-berlin.de/ap/rg/2001-10/Gehring2001Full-SWPatITSec.pdf

Geipel, Philipp: „Intermediäre im Internet am Beispiel des B2C E-Commerce“. Seminar Marketing und Electronic Commerce, Institut für Entscheidungstheorie und Unternehmensforschung.

www.corps-saxonia.de/seminararbeiten/bwl/Intermediaere_B2C_Vortrag.pdf

Gertz, Winfried: „Das Consulting-Geschäft um Open Source“. Computerwoche, April 27, 2001

www2.computerwoche.de/index.cfm?pageid=267&type=ArtikelDetail&id=149632&cfid=2388386&cftoken=35343586&nr=8#

Ghosh, Rishab: „Cooking pot markets: an economic model for the trade in free goods and services on the Internet“. First Monday, Vol. 3, No. 3, March 1998

www.firstmonday.org/issues/issue3_3/ghosh/index.html

Ghosh, Rishab; Prakash, Vipul Ved: „The Orbiten Free Software Survey“. First Monday, Vol. 5, No. 7, July 2000

www.firstmonday.dk/issues/issue5_7/ghosh/index.html

Gillen, Al, Dan Kusnetzky, Scott McLarnon: „The Role of Linux in Reducing the Cost of Enterprise Computing. An IDC White Paper Sponsored by Red Hat Inc.“ IDC/Red Hat, 2001.

www.redhat.com

Goder, Andrea: „Entdeckung der Linux-Welt“. Computerwoche Special, 02/2001, 2001

Goder, Andrea: „Pinguine und Pleitegeier: Open-Source-Firmen in der Konsolidierungsphase“. Computerwoche Special, 02/2001, 2001

Grassmuck, Volker: „Geschichte und Mechanismen freier Software“, Wizards of OS-Kongreß, 2000.

www.mikro.org/Events/OS/text/gesch-freie-sw.html

Hall, Mark: „Start-up's Apps Make Desktop Linux Easier“. Computerworld, September 9, 2000

www.computerworld.com/cwi/story/0%2c1199%2cNAV47_STO63615%2c00.html

Hecker, Frank: „Setting up Shop“. 1999

www.hecker.org/writings/setting-up-shop.html

Henkel, Thomas: „Sooner or Later Even Linux Has to Turn a Profit“. Gartner Group, 2001

www.gartner.com/resources/96600/96692/96692.pdf

Sebastian Hetze, Dirk Hohndel, Olaf Kirch, Martin Müller: „Das Linux Anwenderhandbuch“. LunetIX GbR/ Softfair Verlag, 1993.

Hillesley, Richard: „Money to burn“. Linux User UK, 2001

www.linuxuser.co.uk/articles/issue10/lu10-Cover_feature-Money_to_burn.pdf

Hoch, Detlev J.: „Secrets of Software Success: Management Insights from 100 Software Firms around the World“. Harvard Business School Press; Boston, Massachusetts, 1999

Hoffmann, Naomi: „OSS“. 1999.

public.kitware.com/VTK/pdf/oss.pdf

Howe, Carl D.: „Open Source Cracks The Code“. Forrester Research, 2000.

www.forrester.com/ER/Research/Report/Summary/0,1338,9851,FF.html

Ilan, Yaron: „The Economics of Software Distribution over the Internet Revisited”. Firstmonday, 2001.

www.firstmonday.dk/issues/issue6_12/ilan/index.html

IDC: „Linux: What’s the use? Western Europe 1999-2004”. IDC; 2000

Jaeger, Till: „Klage wegen GPL-Verletzung“. August 13, 2001

www.ifross.de/

Jaeger, Till: „Copyright oder Copyleft”. Computerwoche Spezial 4/2000, p. 36, 2000

www.ifross.de/ifross_html/art6.html

Jeong, Byung Seon: „Analysis of the Linux Operating System, a New Entrant in the Operating System Markt: Technological Innovations and Business Model”. Dissertation, 1999

Johnson, Justin Pappas: „Some Economics of OSS”. December 2000

www.idei.asso.fr/Commun/Conferences/Internet/Janvier2001/Papiers/Johnson.pdf

Kaven, Oliver: „Performance Tests: File Server Throughput and Response Times”. PC Magazine, November 13, 2001

www.pcmag.com/print_article/0,3048,a%253D16554,00.asp

KBSt: „KBSt-Brief Nr. 2/2000: OSS in der Bundesverwaltung”. KBSt, 2000

linux.kbst.bund.de/02-2000/

McKelvey, Maureen: „Internet Entrepreneurship: Linux and the dynamics of OSS”. CRIC, 2001

les1.man.ac.uk/cric/dp44.htm

Khalak, Asif: „Economic model for impact of OSS”. MIT, 2001

opensource.mit.edu/papers/osseconomics.pdf

Klever, Ralf: „die tageszeitung, Redaktionssystem“. Wizards of OS 1, 1999.

mikro.org/Events/OS/ref-texte/klever.html

Köhntopp, Kristian et al: „Sicherheit durch Open Source? Chancen und Grenzen“ in „Datenschutz und Datensicherheit (DuD) 24/9 (2000)“. Vieweg, Wiesbaden 2000; S. 508-513.

www.koehntopp.de/marit/pub/opensource/KoeKP_00SicherheitOpenSource.pdf

Köppen, A.; Nüttgens, M.: „Open Source: Strategien für die Beratung”. In: Scheer, A.-W.; Consulting – Wissen für die Strategie-, Prozess- und IT-Beratung, Saarbrücken, pp. 231-242, 2000

www.iwi.uni-sb.de/nuettgens/Veroef/Artikel/Consulting/OpenSourceStrategien.pdf

Krempl, Stefan: „Studie empfiehlt Schily Server-Umrüstung auf Linux“. heise online News, 7. Dezember 2001.

www.heise.de/newsticker/data/jk-07.12.01-008/

Krempf, Stefan: „Bundesrechnungshof fordert Einsatz von Open Source“. heise online News, 25. Februar 2002.

www.heise.de/newsticker/data/anw-25.02.02-004/

Krueger, Patricia: „Tour de Source - A Guide to the Start Ups“. Wired, 1999.

www.wired.com/wired/archive/7.05/tour.html

Krüger, Wilfried, Bach, Norbert: „Geschäftsmodelle und Wettbewerb im e-Business“, in Supply Chain Solutions - Best Practices im E-Business, hrsg. von Buchholz W./ Werner, H., Schaeffer-Poeschel 2001, S. 29-51.

www.eic-partner.de/material/ebus.pdf

Lancashire, David: „Code, Culture and Cash: The Fading Altruism of Open Source Development“. Firstmonday, 2001.

www.firstmonday.dk/issues/issue6_12/lancashire/index.html

Lerner, Josh; Tirole, Jean: „The simple Economics of Open Source“. December 2000

www.univ-tlse1.fr/idei/Commun/Articles/Tirole/Simple-Economics.pdf

Lewis, Michael, Walch, Kameron, Whitehouse, Janelle: „Managing Technological Innovation in the Wired Society:

The Case of Linux Operating System“, Reality Bites, LLP, 2001

eplu.netgistics.com/PDF/1999-2000/Fall/BUSA541/Bytes.pdf

Lutterbeck, Bernd: „Gutachten für das Expertengespräch ‚Softwarepatente und Open Source““. Informatik und Gesellschaft, TU Berlin, 2001.

www.ig.cs.tu-berlin.de/bl/061a/Lutterbeck-BT-6-2001.pdf

Mantarov, Bojidar: „OSS as a new business model“. Dissertation, Reading, 1999

bmantarov.free.fr/academic/msc_essays/dissert.htm

Materna Unternehmensberatung: „Ihre erfolgreiche E-Business-Strategie“. 2001.

www.materna.de/Dateien/eBusiness/07220e-Business_RZ1.pdf

Miller, Robin: „Learning from Mozilla's mistakes“. NewsForge, October 23, 2001

www.newsforge.com/article.pl?sid=01/10/20/1841215&mode=thread

Monaco, Carol, Volpi, Ana: „Linux Support Services Forecast and Analysis, 1999-2004“. IDC Doc #23598, Dezember 2000

www.idc.com/getdoc.jhtml?containerId=23598

Mercer Management Consulting: „Managing the professional service firm“. Lehrstuhl Prof. Wolfgang Becker, Universität Bamberg, 2001

www.uni-bamberg.de/sowi/ufc/inhalt/download/studienunterlagen/fs_mmc.pdf

Mundie, Craig: „The Commercial Software Model“. The New York University Stern School of Business, 3. Mai 2001

www.microsoft.com/presspass/exec/craig/05-03sharedsource.asp

Mustonen, Mikko: „Copyleft - the Economics of Linux and other OSS“. Univ. of Helsinki, 2001

data.vatt.fi/antitrust/papers/m_mustonen.pdf

Nadeau, Tom: „Learning from Linux“. OS/2 and the Halloween Memos. OS2Headquarters, 1999.

pcsupport.about.com/gi/dynamic/offsite.htm?site=http%3A%2F%2Fwww.os2hq.com%2Farchives%2Flinmemo1.htm

N.N.: „The Open Source Case for Business, Opensource.org“. 2001

www.opensource.org/advocacy/case_for_business.html

N.N.: „SuSE will im zweiten Halbjahr profitabel sein“. Linux-Enterprise 07.02.2002.

www.entwickler.com/news/2002/02/5505/news.shtml

Netcraft: „Netcraft Web Server Survey, October 2001“. 2001

www.netcraft.com/survey

Nüttgens, M.; Tesei, E.: „Open Source - Konzept, Communities und Institutionen“. In: Scheer, A.-W. (Editor): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 156, Saarbrücken 2000a

www.iwi.uni-sb.de/nuettgens/Veroef/Artikel/heft156/heft156.pdf

Nüttgens, M.; Tesei, E.: „Open Source - Marktmodelle und Netzwerke“. In: Scheer, A.-W. (Editor): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 158, Saarbrücken 2000b

www.iwi.uni-sb.de/nuettgens/Veroef/Artikel/heft158/heft158.pdf

Nüttgens, M.; Tesei, E.: „Open Source - Produktion, Organisation und Lizenzen“. In: Scheer, A.-W. (Editor): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 157, Saarbrücken 2000c

www.iwi.uni-sb.de/nuettgens/Veroef/Artikel/heft157/heft157.pdf

von Oberkulm , Peter Michael Fäs: „Business Models des Electronic Commerce“. Lizentiatsarbeit, 2001.

www.peterfaes.ch/lic/files/lpfxay.pdf

Osterberg, Jürgen: „Flotter Auftritt mit Zope“. Computerwoche Nr. 44/ 2001, p.26, October 19, 2001

O'Reilly (Autor N.N.): „Open Source - kurz & gut“. O'Reilly & Associates, 1999.

Porter, Michael: „Competitive Strategy“. New York, 1995

Pomerantz, Gregory M.: „Business Models for Open Source Hardware Design“. New York, 2000.

pages.nyu.edu/~gmp216/papers/bmfosh-1.0.html

Prasad, Ganesh: „Open Source-onomics: Examining some pseudo-economic arguments about Open Source“. Linuxtoday, 2001.

linuxtoday.com/news_story.php3?ltsn=2001-04-12-006-20-OP-BZ-CY

Quade, Jürgen: „Qualität hat keinen Preis“. Computerwoche Spezial 2/2001, Computerwoche Verlag, 2001.

Rasch, Chris: „The Wall street Performer Protocol: Using Software completion Bonds to Fund OSS development“. First Monday, Vol. 6, No. 6, June 2001

www.firstmonday.dk/issues/issue6_6/rasch/

Raymond, Eric: „Homesteading the Noosphere“. 1998a

www.tuxedo.org/~esr/writings/cathedral-bazaar/

Raymond, Eric: „The Cathedral and the Bazaar“. 1998b

www.tuxedo.org/~esr/writings/cathedral-bazaar/

Raymond, Eric: „The Magic Cauldron“. 1999

www.tuxedo.org/~esr/writings/cathedral-bazaar/

Roehrl, Armin, Schmiedl, Stefan: „Vogelfrei. Die wichtigsten Open Source-Lizenzen“. ct 1/2002, Heise Verlag 2002.

Robles , Gregorio: „WIDI - Who Is Doing It? Knowing more about Developers“. Informatik und Gesellschaft Institute of the Technical University of Berlin, 2001.

widi.berlios.de

Rosenberg, Donald K.: „Open Source: The Unauthorized White Papers“. M&T Books, Foster City, 2000

Sandred, Jan: „Managing Open source Projects: A Wiley Tech Brief“. John Wiley & Sons, Inc., 2001

Schenk, Thomas: „Linux: Its history and current distributions“. IBM

www.developer.ibm.com/library/articles/schenk1.html

Schmitz, Ludger: „Der steinige Weg zu Linux-Standards“. Computerwoche, May 4, 2001

Shapiro, Carl, Varian, Hal R: „Information Rules: A Strategic Guide to the Network Economy“. Harvard Business School Press, 1998.

www.inforules.com

Shankland, Stephen: „VA Linux to sell proprietary software“. News.com, 2001a.

news.cnet.com/news/0-1003-200-6954900.html

Shankland, Stephen: „Sun acknowledges acquisition misfires“. News.com, 2001b.

news.cnet.com/news/0-1003-200-7383790.html

Spiller, Dorit: „Free/Libre and Open Source Software: Survey and Study. OS/FS business models and best practices (working paper). Berlecon Research Berlin, 2001.

IDA/Unisys: „Study into the use of OSS in the public sector. Part 1-3” Europäische Kommission, Unisys, 2001
ag.idaprogram.org/Indis35prod/doc/333

Security Space: „Security Space Web Server Survey, October 2001”. 2001
www.securityspace.com/s_survey/data/200110/index.html

Sergio, G.: „Got Linux? Many companies say no”. CNET, November 2001
news.cnet.com/news/0-1003-200-7803522.html?tag=mn_hd

Slywotzki, Adrian J., Morrison, David J.: „Die Gewinnzone. Wie Ihr Unternehmen dauerhaft Erträge erzielt“. Mercer Management Consulting, Verlag Moderne Industrie, 1997

Succi, Giancarlo; Paulson, James; Eberlein, Armin: „Preliminary Results from an Empirical Study on the Growth of Open Source and Commercial Software Products”. Paper submitted for „Third International Workshop on Economics-Driven Software Engineering Research”(EDSER-3 2001), 2001
www.cs.virginia.edu/~sullivan/edser3/paulson.pdf

TechConsult: „Fast jedes 5. Unternehmen setzt 2003 auf Linux”. Press Release TechConsult, 2001
www.techconsult.de/de/Services/studiendetail.cfm?id=82

Tuomi, Ilkka: „Internet, Innovation and Open Source: Actors in the Network”. First Monday, Vol. 6, No. 1, January 2001
www.firstmonday.dk/issues/issue6_1/tuomi/index.html#author

Valloppillil, Vinod, Cohen, Josh: „The Halloween Documents”. Open Source Initiative, 1998.
www.opensource.org/halloween

Varian, H.R.: “Versioning Information Goods”. Research paper prepared for Digital Information and Intellectual Property, Harvard University, 1997
www.sims.berkeley.edu/~hal/Papers/version.pdf

Varian, H.R.: „Differential Pricing and Efficiency”. First Monday, volume 1, number 2, 1996
firstmonday.org/issues/issue2/different/

Varian, H.R.: „Pricing Information Goods” presented at the Research Libraries Group Symposium on “Scholarship in the New Information Environment”. Harvard Law School, Cambridge, Mass, 1995
www.sims.berkeley.edu/~hal/Papers/price-info-goods.pdf

Välimäki, Mikko: „Strategic Use of Intellectual Property Rights in Digital Economy — Case of Software Markets”. Helsinki Institute for Information Technology, 2001.
www.hiit.fi/de/hamilton.pdf

VDC (Venture Development Corporation): „Linux's Future in the Embedded Systems Market (Summary for the Embedded Linux Community)”. June 2001

www.linuxdevices.com/articles

Vepstas, Linas: „Is Free Software Inevitable?”. 2001

linas.org/theory/freetrade.html

Wayner, Peter: „Open source databases bloom”. Computerworld, September 9, 2001

www.computerworld.com/storyba/0,4125,NAV47_STO63629,00.html

Weber, Steven: „The political Economy of OSS”. BRIE, 2000

brie.berkeley.edu/~briewww/pubs/wp/wp140.pdf

Weiss, George: „The Future of Linux and Open Source”. Gartner, Strategy & Tactics/Trends & Direction, Note Number AV-13-9850, June 20, 2001

www.gartner.com/resources/98800/98894/98894.pdf

Weiss, George: „OS Evaluation: Linux vs. Unix and Windows 2000”. Gartner Research Note, 2000

www.gartnerweb.com/public/static/hotc/hc00091281.html

Wheeler, David: „More Than a Gigabuck: Estimating GNU/Linux's Size, Version 1.06, 30.6.2001, updated 8.11.2001”. 2001a

www.dwheeler.com/sloc/redhat71-v1/redhat71sloc.html

Wheeler, David: „Why OSS / Free Software (OSS/FS)? Look at the Numbers!”. 2001b

www.dwheeler.com/oss_fs_why.html

Whitlock, Nathalie Walker: „The security implications of OSS”. IBM, March 2001

www-106.ibm.com/developerworks/library/l-oss.html

Wiehr, Hartmut; Wild, Martin: „Linux - Die Alternative”. Informationweek, July 7, 2001 and July 26, 2001

www.informationweek.de/

Wieland, Thomas: „Linux als Geschäftsfaktor”. Linux Enterprise, Linuxtag 2001.

www.cpp-entwicklung.de/download/linuxtag2000_wieland.pdf

Wysong, Thom: „Introduction to Open Source and Free Software”. 2000

www.techdemocracy.org/papers/ossfs.html

Yager, Tom: „BSD's strength lies in devilish details”. InfoWorld, November 2, 2001

www.infoworld.com/articles/tc/xml/01/11/05/011105tcbsd.xml

Zerdick, Axel, Picot, Arnold, Schrape, Klaus, Artopé, Alexander, Goldhammer, Klaus, Lange, Ulrich T., Vierkant, Eckart, López-Escobar, Esteban, Silverstone, Roger: „Die Internet-Ökonomie. Strategien für die digitale Wirtschaft“. Springer, 1999

Abbildungsverzeichnis

Abbildung 1: Die klassische Wertschöpfung der Softwareindustrie	8
Abbildung 2: Veränderung der Wertschöpfung durch OSS.....	8
Abbildung 3: Rollen bei der OSS-Entwicklung.....	12
Abbildung 4: Struktur des IT-Marktes	17
Abbildung 5: OSS-Produktmarktsegmente und deren Relevanz im kommerziellen Bereich .	18
Abbildung 6: Linux ist das Betriebssystem mit der größten Anwendungsbreite.....	20
Abbildung 7: Anzahl der Programme für die jeweilige Plattform	22
Abbildung 8: Linux-Programme nach Kategorien.....	22
Abbildung 9: Zusammenspiel von Marktplatz-Teilnehmern	27
Abbildung 10: Datendurchsatz von Linux/Samba im Vergleich mit Windows 2000.....	31
Abbildung 11: Allgemeine Software Value Chain	40
Abbildung 12: Business Design Modell.....	42
Abbildung 13: Zuordnung von Unternehmenstypen zu Marktsegmenten	44
Abbildung 14: Ressourcenfokus der Distributoren.....	48
Abbildung 15: Wertschöpfungskette der OSS-Applikationsanbieter	54
Abbildung 16: Wertschöpfungskette der Appliance-Anbieter.....	57
Abbildung 17: Der Ressourcenfokus der OSS-Dienstleister	61
Abbildung 18: OSS-Geschäftsmodelle und deren Transformationen.....	68
Abbildung 19: Entwicklung des Linux Stock Index 2000 im Vergleich zum Nasdaq Composite Index	70
Abbildung 20: Gartner-Prognose über die Entwicklung des Linux-Marktes	72

Tabellenverzeichnis

Tabelle 1: Vergleich der Eigenschaften wichtiger OSS-Softwarelizenzen.....	11
Tabelle 2: Marktanteile der Server-Betriebssysteme in Prozent.....	19
Tabelle 3: Der Markt für Linux-Dienstleistungen.....	25
Tabelle 4: Linux-Einsatzgebiete.....	28
Tabelle 5: Preise für Linux-Distributionen	46
Tabelle 6: Preise für Zusatzprodukte der Linux-Distributoren	46
Tabelle 7: Embedded-Unternehmen und deren Embedded-Applikationen	66
Tabelle 8: Gartner-Prognose über die kommerzielle Linux-Entwicklung.....	72