

Dieser Artikel ist Teil des  
Open Source Jahrbuch 2005



erhältlich unter <http://www.opensourcejahrbuch.de>.

Das Open Source Jahrbuch 2005 enthält neben vielen weiteren interessanten Artikeln ein Glossar und ein Stichwortverzeichnis.

# Das Microsoft-Shared-Source-Programm aus der Business-Perspektive

WALTER SEEMAYER UND JASON MATUSOW



*(CC-Lizenz siehe Seite 463)*

Dieser Artikel beschreibt und begründet die strategischen Erwägungen hinter der Shared-Source-Initiative von Microsoft. Hierfür wird der Begriff des Software-Ökosystems eingeführt, der die wechselseitigen Abhängigkeiten von kommerziellen Firmen und nicht-kommerziellen Institutionen wie Universitäten beschreibt. Die Shared-Source-Initiative versucht, einige der unbestrittenen Vorteile von Open-Source-Entwicklung zu übernehmen und mit dem Geschäftsmodell von Microsoft zu vereinbaren, dass traditionell auf der Wertschöpfung durch Lizenzierung von proprietären Softwareprodukten basiert.

## 1. Hintergrund

In der Softwarebranche reißt die Diskussion über den Stellenwert von offener, kostenloser, kommerzieller und nichtkommerzieller Software nicht ab. Tatsächlich gibt es jedoch weitaus mehr Gemeinsamkeiten als Unterschiede. Wo aber doch Unterschiede bestehen, sollten deren Auswirkungen auf Unternehmen, Einzelpersonen, wissenschaftliche Institutionen und Regierungsbehörden näher beleuchtet werden.

In der herrschenden Diskussion um Open Source geht es um ein breites Spektrum von Themen wie Nutzungsflexibilität, Preis-Leistungs-Verhältnis, wirtschaftliche Aspekte, geistige Eigentumsrechte für Software, Industriestandards, Sicherheit, Datenschutz, Geschäfts- und Lizenzmodelle usw. Durch all diese Aspekte zieht sich einheitlich die Thematik des Zugangs zum Quellcode. Oberflächlich betrachtet ist der Quellcode ausschließlich für den Entwickler bestimmt. Dennoch stellt sich neben rein technischen Fragen der Generierung und Modifizierung des Quellcodes die grundlegende Frage nach der Zukunft von Software-Innovationen. Aus diesem Grund geht auch die Diskussion weiter.

Seltsamerweise spricht zwar fast Jeder davon, wie wichtig es ist, auf den Quellcode von Software zugreifen zu können, doch sind nur sehr wenige bereit oder in der Lage, mit diesem zu arbeiten. Um in dieser Hinsicht eindeutige und aussagekräftige Resultate zu erhalten, startete Microsoft 2002 eine Untersuchung (unveröffentlicht) bezüglich des Quellcode-Zugriffs für die in Unternehmen und Regierungsbehörden

verwendete Software.<sup>1</sup> Man hätte nun vielleicht erwartet, dass die IT-Experten der Unternehmen in ihrer täglichen Arbeit viel mit Quellcode umgehen. Stattdessen ergab die Untersuchung, dass ca. 95 % der Unternehmen und Behörden den Quellcode der Betriebssysteme, auf denen ihre technische Infrastruktur basiert, überhaupt nicht beachten. Darüber hinaus wurde festgestellt, dass zwar fünf Prozent der Befragten den Quellcode kennen, aber weniger als ein Prozent ihn verändert. Bei kleineren Unternehmen ist der Prozentsatz derer, die auf den Quellcode zugreifen und ihn verändern, sogar noch geringer.

Die Schwelle für das Verstehen komplexen Quellcodes ist extrem hoch. Zwar gibt es weltweit Millionen von Softwareentwicklern, doch machen diese im Vergleich zu allen Anwendern von Computern nur einen Bruchteil aus. Ferner sind die Programmierkenntnisse dieser Entwickler unterschiedlich gut ausgeprägt, so dass die Community, die hochgradig komplexe Codes handhaben kann, insgesamt noch kleiner ist. Für die meisten Unternehmen ist das Verhältnis von Kosten und Nutzen bei der Beschäftigung hochgradig qualifizierter Entwickler untragbar, vor allem weil es heute ohnehin eine Vielzahl qualitativ hochwertiger Softwarepakete gibt.<sup>2</sup>

Trotzdem gaben Unternehmen an, dass ihnen die Möglichkeit des Zugriffs auf den Quellcode von Betriebssystemen wichtig sei.<sup>3</sup> Die Mehrzahl der Unternehmen und Regierungsbehörden befürworteten die Option eines zugänglichen Quellcodes. Einfach ausgedrückt: Transparenz erhöht das Vertrauen.

Das legt den Schluss nahe, dass es für die meisten Anwender weit wichtiger ist, die Möglichkeit zu haben, etwas zu tun, als es dann tatsächlich auch zu tun. Ein gutes Beispiel hierfür ist die gesetzlich vorgeschriebene Offenlegung der Bilanzen von Aktiengesellschaften. Zwar sind diese Bilanzen öffentlich zugänglich, doch sind sie äußerst kompliziert. Um den Status einer Gesellschaft bewerten zu können, ist ein fundiertes Finanzwissen erforderlich. Der Großteil der privaten Investoren benötigt daher die Hilfe einer kleinen Gruppe von Fachleuten, die für sie die Zahlen deuten und Empfehlungen aussprechen. Die Möglichkeit, die Zahlen einzusehen, ist also für alle gegeben: Aufgrund der gegebenen Transparenz besteht daher Vertrauen. Die meisten Investoren werden jedoch von dieser Möglichkeit nie Gebrauch machen.

Der private Investor und der typische Anwender heutiger Betriebssysteme befinden sich in einer ähnlichen Situation. Die meisten Unternehmen oder Einzelpersonen haben gar nicht vor, in die Tiefen des Betriebssystems vorzudringen und den Quellcode zu verändern.<sup>4</sup> Unternehmen und der durchschnittliche Konsument hängen stark von

---

1 An diesen privaten Forschungsarbeiten waren mehr als 1110 Personen aus fünf Ländern beteiligt, darunter Entscheidungsträger von Unternehmen, IT-Fachleute und Entwickler. Die Studie wurde im April 2003 abgeschlossen. Befragt wurden Personen welche mit Open-Source-Software (OSS) und/oder Microsoft-Systemen arbeiten und die Möglichkeit haben, auf Quellcode zuzugreifen.

2 Diese Annahme basiert auf der stark vereinfachten Auslegung des Transaktionskosten-Konzepts von Ronald Coase und dessen Einfluss auf das Verhalten von Unternehmen. Coase erhielt 1991 den Nobelpreis für Wirtschaftswissenschaften für seine Arbeiten in Bezug auf die Bedeutung von Transaktionskosten und Eigentumsrechten für die institutionelle Struktur und Funktion der Wirtschaft (<http://coase.org/>, Stand 20. Mai 2003).

3 Die in Fußnote 1 genannte Studie ergab, dass ca. 60 % der Befragten die Möglichkeit zum Einblick in den Quellcode für die Nutzung von Software in einer Geschäftsumgebung für unerlässlich halten.

4 Dies gilt in gleichem Maße für Windows, Linux, Mac OS, Netware, OS/400 und andere bedeutende

kommerziellen Anbietern ab, die ihnen das erwartete Qualitäts- und Supportniveau garantieren. Und genau an dieser Stelle liefern kommerzielle Softwareanbieter einen Mehrwert für die von ihnen entwickelten und vertriebenen Produkte.<sup>5</sup>

Vor drei Jahren wurde die Shared-Source-Initiative bei Microsoft ins Leben gerufen. Im Rahmen dieser Initiative stellen wir verschiedene Arten von Microsoft-Quellcode für Kunden, Regierungsbehörden, Partner und Mitbewerber in der ganzen Welt bereit. Ein Teil des Quellcodes – wie der von Windows – wird nur zur Einsichtnahme verfügbar gemacht (d. h. es können keine Änderungen durchgeführt werden), während andere Programme, die Technologien wie Windows CE.NET betreffen, die Änderung und den Weitervertrieb des Quellcodes erlauben.<sup>6</sup>

Im Zuge unserer Tätigkeit in der Shared-Source-Initiative haben wir unzählige Gespräche mit Einzelanwendern und Unternehmen über die Bedeutung von Quellcode für ihre speziellen Anforderungen geführt. Obwohl wir mehr als 800 000 Entwicklern den Quellcode an bekannt gegeben haben, ist das nur ein sehr geringer Prozentsatz aller Entwickler, die mit Microsoft-Technologien arbeiten. Unsere praktischen Erfahrungen auf globaler Ebene bestätigen, dass die operationalen Bedürfnisse, wie oben erwähnt, mit dem Beruhigungsfaktor der Transparenz zusammenhängen. Bei dem Ansatz, unseren Quellcode zu lizenzieren, spielen ebenfalls Faktoren wie Transparenz, Wahlmöglichkeiten, Vertrauen und Bedürfnisse eine wichtige Rolle.

Unser Ansatz beruht auf drei einfachen Grundgedanken: Erstens verlangen die Kunden Zugriff auf den Quellcode aus technischen Gründen und aufgrund der Tatsache, dass Transparenz das Vertrauen steigert. Zweitens gibt es keinen einheitlichen Weg der Bereitstellung von Quellcode durch Microsoft, um alle geschäftlichen und lizenztechnischen Anforderungen in der gesamten Produktpalette zu erfüllen. Drittens ist der Kunde beim Umgang mit dem Quellcode erfolgreicher, wenn er zusätzlich zur Technologie solide Tools und Informationen erhält. Bei der Entwicklung des Shared-Source-Ansatzes hat Microsoft diese Grundgedanken berücksichtigt. Shared Source ist nicht gleichbedeutend mit Open Source. Vielmehr ist es ein Mittel für Unternehmen, die Software direkt vermarkten, um Quellcode bereitzustellen, ohne dabei ihre Alleinstellungsmerkmale und ihr Geschäftsmodell zu schwächen. Microsoft erkennt die Vorteile des Open-Source-Modells zwar an, doch gilt es zu berücksichtigen, dass dieses Modell nicht universell anwendbar ist.

Diese Abhandlung verfolgt zwei Zielsetzungen: Erstens sollen die Shared-Source-

---

kommerzielle Betriebssysteme. Schon das kleinste von ihnen umfasst Millionen von Zeilen an Quellcode (ein Betriebssystem ist mehr als nur ein Kernel), wobei die Komplexität dieser Systeme im Verlauf ihrer Entwicklung eher zunimmt als abnimmt.

- 5 Natürlich dürfen die Fähigkeiten eines Community-Supports nicht unterschätzt werden. Seit Jahren bestehen Newsgroups und Mailinglisten, in denen die Community Support für kommerzielle, offene, kostenlose und als Shareware verfügbare Software leistet. Die Unternehmen bevorzugen im Hinblick auf ihre unternehmenskritischen Systeme allerdings eher einen professionellen Support mit Service Level Agreements zur Eindämmung der Risiken.
- 6 Mit Stand vom Mai 2003 bietet Microsoft entsprechende Programme für Windows, Windows CE.NET, Visual Studio .NET, C#/CLI, ASP.NET und Passport an. Für die Windows-Programme werden nur akademisch tätigen Wissenschaftlern Rechte zur Veränderung des Quellcodes gewährt. Für alle anderen Quellcode-Programme werden Rechte zur Änderung und zum Weitervertrieb eingeräumt. Weitere Informationen finden sich unter <http://www.microsoft.com/sharedsource/>.

Initiative sowie kommerzielle Software in den breiteren Kontext der derzeitigen Debatte über die Quellcode-Lizenzierung gestellt werden. Zweitens soll erläutert werden, wie Microsoft die Lizenzierung seiner zentralen Ressourcen an geistigem Eigentum angeht.

## 2. Allmähliche Annäherung

2000 und 2001 schienen die Fronten der Beteiligten der Quellcode-Debatte klar gezogen: Microsoft galt als eindeutiger Pol in der ständig neue Positionen hervorbringenden Debatte mit den traditionellen Verfechtern geistigen Eigentums auf der einen Seite und den Gegnern kommerziellen Eigentums an einer Software auf der anderen Seite. Dem Open-Source-Ansatz verschriebene Einzelpersonen und Unternehmen stellten sich bewusst als Alternative zu Microsofts Geschäftspraktiken oder gar als aktive Gegner von Microsoft dar. Heute, 2003, hat sich infolge der Nachwirkungen des Dot.com-Zeitalters ein pragmatischer Ansatz bei Unternehmen wie auch bei Einzelpersonen durchgesetzt.

Open-Source-Software (OSS) ist als Softwarekategorie in kommerzielle und nicht-kommerzielle Sektoren unterteilt. Für viele finden die interessantesten OSS-Fortschritte heute in der kommerziellen Kategorie statt, da bedeutende Investitionen, Ressourcen und Technologien von Seiten all derer einfließen, die OSS als Grundlage für strategische Geschäftszwecke nutzen wollen.

Ein genauer Blick auf die kommerzielle Software-Community zeigt, dass sich die Quellcode-Lizenzierung bei den wichtigsten Anbietern durchgesetzt hat. Auf dem heutigen Markt bilden Softwareentwicklung, Lizenzierung und Geschäftsstrategien eine Mischung aus Community-basierten und kommerziellen Modellen. Nur wenige Softwareunternehmen können sich heute noch ausschließlich als OSS-orientiert (im Sinne von OSS als durch die Community entwickelte und nicht gewinnorientierte Software) oder rein kommerziell bezeichnen.

Für diese Diskussion soll zunächst einmal eine klare Linie zwischen nichtkommerzieller und kommerzieller Software gezogen werden. Die wichtigen Rollen beider Kategorien im Rahmen des in den letzten 30 Jahren entstandenen Software-Ökosystems werden später in dieser Abhandlung erörtert.

Nichtkommerzielle Software lässt sich grob in drei Kategorien aufteilen:

*Forschung* Regierungsbehörden und akademische Wissenschaftler entwickeln Technologien, die den allgemeinen Entwicklungsstand vorantreiben sollen.

*Lehre* Professoren, Studenten und Autodidakten arbeiten und lernen mit kostenloser Software, wobei ihre Forschungsergebnisse nicht der kommerziellen Vermarktung dienen.

*Entwicklung und Problembehandlung innerhalb der Community* Freizeitprogrammierer oder professionelle Entwickler erstellen Software ohne kommerzielle Absichten, so z. B. Software, die bestehende kommerzielle Programme ersetzt oder Probleme behebt, die vom Hersteller nicht berücksichtigt wurden.

Kommerzielle Software lässt sich grob in zwei Kategorien aufteilen:

*Direkte Vermarktung* Von der Community und/oder von Unternehmen entwickelte Produkte werden zur Generierung direkter Einnahmen verwendet.

*Indirekte Vermarktung* Von der Community und/oder von Unternehmen entwickelte Produkte werden für die gewinnbringende Vermarktung anderer Produkte oder Dienstleistungen eingesetzt.

Hierbei ist anzumerken, dass die Konzepte der kommerziellen und nichtkommerziellen Software nichts mit der Zugänglichkeit des Quellcode zu tun haben. Wenn der langjährige Anbieter einer kommerziellen Software den Quellcode eines bestimmten Produkts bekannt gibt, ändert das nichts an dem kommerziellen Ursprung der Software. Gleichzeitig kann ein kommerziell auftretendes Unternehmen eine gemeinschaftliche, nichtkommerzielle Software nutzen, ohne dass diese Software dadurch ihren Status als nichtkommerzielle Software einbüßt.

In der folgenden Liste werden die oben angeführten kommerziellen Kategorien anhand von Beispielen aus der Softwarebranche dargestellt. Obwohl sich viele der unten aufgeführten Unternehmen der OSS-Bewegung zuordnen, sind sie doch eindeutig kommerzielle Unternehmen. Einige der genannten Unternehmen verfolgen wiederum das OSS-Konzept nicht in direkter Weise und können doch aufgrund ihres Ansatzes bei der Softwarevermarktung als Beispiel dafür herangezogen werden.

*Direkte Vermarktung von Community-Entwicklungen* Die Linux-Versionen von Red Hat, Inc. und SuSE sind Kombinationen aus von der OSS-Community entwickelter Software und vom Unternehmen finanzierten, professionellen Entwicklungsbeiträgen. Die Preisgestaltung der *Premium Editions*, die Zertifizierungsverfahren für Hardware und Anwendungen sowie die Support-Modelle von Red Hat dienen alle als Mechanismen zur direkten Vermarktung des Betriebssystems.

Apple Computer, Inc. hat Community-Software mit kommerzieller Software kombiniert, um das Betriebssystem OS X zu erstellen. Das Unternehmen vermarktet die Software direkt und verwendet von der Community entwickelten Code.

*Direkte Vermarktung von Unternehmens-Entwicklungen* Microsoft hat das Windows-Produkt unter Nutzung unternehmenseigener Entwicklungsressourcen erstellt. Das Produkt wird durch die Lizenzierung seiner Binärversion direkt vermarktet. Der Quellcode ist nun im Rahmen der Shared-Source-Initiative für eine begrenzte Community verfügbar.

CollabNet, Inc. hat ein proprietäres Tool entwickelt, das durch eine Lizenzierung seiner Binärversion sowie durch zugehörige Dienste direkt vermarktet wird. Das Produkt erleichtert die Nutzung des OSS-Entwicklungsmodells, das zur Entwicklung nichtkommerzieller Software eingesetzt werden kann.

*Indirekte Vermarktung von Community-Entwicklungen* IBM Corp. spielte eine wichtige Rolle in der Community-basierten Entwicklung des Apache-Webservers. Obwohl IBM den Apache-Server nicht direkt vermarktet, generiert das Unternehmen durch den Verkauf des WebSphere-Produktes Einnahmen aus dieser Investition.

RealNetworks, Inc. veröffentlichte entscheidende Segmente des Quellcodes seines Produktes Helix, das ursprünglich kommerziell entwickelt wurde. Ziel der Community-Entwicklung im Zusammenhang mit der Helix-Produktpalette ist die Erschließung eines breiteren Marktes für andere wertschöpfende Produkte.

*Indirekte Vermarktung von Unternehmens-Entwicklungen* Der Acrobat Reader von Adobe Systems, Inc. ist eine Verbindung aus unternehmenseigener Entwicklung und streng gewährtem geistigen Eigentum.<sup>7</sup> Der Reader steht kostenlos zum Download bereit, um so den Verkauf der Vollversion des Acrobat-Produktes zu fördern.

*Driver Development Kits (DDKs) und Software Development Kits (SDKs)* werden von allen Anbietern kommerzieller Betriebssysteme bereitgestellt (so z. B. das NDK von Novell, Inc. und das DDK von Microsoft). Diese Entwicklertools enthalten häufig Beispiele für Quellcode, die verändert und weiterverbreitet werden können, obgleich die Kits den Entwicklern kostenlos zur Verfügung gestellt werden. Die DDKs oder SDKs selbst haben keinen direkten kommerziellen Wert, doch schaffen sie für andere die Gelegenheit, Software und Hardware für die jeweilige Plattform zu erstellen.

Oftmals wird fälschlicherweise angenommen, dass nach dem Open-Source-Modell entwickelte Software zufällig von einer Gruppe verstreut arbeitender Entwickler geschrieben und dann von Unternehmen übernommen wird. Das mag zwar für kleinere Projekte zutreffen, doch werden die zentralen OSS-Technologien von professionellen Teams in Unternehmen oder hochgradig strukturierten gemeinnützigen Organisationen erstellt, getestet, vertrieben und gewartet. Das hinter dem Zugriff auf Quellcode stehende Konzept hat mit der Unterscheidung, ob eine Software kommerziell ist oder nicht, nichts zu tun. Der Zugriff auf den Quellcode ist sowohl im kommerziellen als auch nichtkommerziellen Umfeld von Bedeutung.

Die Problematik des Quellcodes hat zweifellos Einfluss auf die Innovationen in der Branche. Die oben erwähnte Annäherung ist das Ergebnis dieses Einflusses auf die heutige Branche.

### 3. Das Software-Ökosystem

Grundlage jeglicher Softwareentwicklung ist die Interaktion zwischen von Regierungsbehörden, akademischen Institutionen und privaten Stellen betriebenen Forschungsbemühungen. Diese Beziehungen stellen ein verknüpftes, natürliches Ökosystem dar.

<sup>7</sup> Adobe Systems, Inc. stellt zwar die Spezifikation des Dateiformats PDF zur Verfügung, legt jedoch nicht den Quellcode für die Implementierung offen. Weitere Informationen unter <http://www.adobe.com>.

Obwohl diese Organe unabhängig voneinander bestehen und unabhängige Entwicklungen vorantreiben, bestehen ganz klar Abhängigkeiten, durch die erheblich weitreichendere Ergebnisse für die Allgemeinheit erbracht werden.

Dieses Ökosystem ist der Motor für fortwährende Innovationen, die den Bereich der Informationstechnik zu einer der dynamischsten Wirtschaftsbranchen gemacht haben.<sup>8</sup> Die Zusammenführung verschiedener Entwicklungs-, Lizenz- und Geschäftsmodelle war dabei der entscheidende Faktor für den Erfolg.

Grundlagenforschung wird von Regierungsstellen und Universitäten erbracht und deren Ergebnisse werden der Allgemeinheit zur Verfügung gestellt.<sup>9</sup> Einige der so entstehenden Technologien werden wiederum im privaten Sektor aufgegriffen und mit stetig steigenden Forschungs- und Entwicklungsetats<sup>10</sup> weiterverfolgt, um neue kommerzielle Produkte zu entwickeln und gleichzeitig die Arbeit von Standardisierungsgremien zu unterstützen. Die Erfolge im privaten Sektor führen zu mehr Beschäftigung, mehr Steueraufkommen und einer zusätzlichen Finanzierung akademischer Forschungsprojekte.<sup>11</sup>

Die mit dem Software-Ökosystem verbundenen Konzepte gelten natürlich nicht nur für die Diskussionen über den Zugriff auf den Quellcode. Ein Beispiel hierfür ist die Luftfahrt. Obgleich die meisten Personen in ihrem Alltag eher wenig mit einem F-15 Eagle-Kampfflugzeug anfangen können, hat die zivile Luftfahrt dennoch von den staatlichen und akademischen Forschungen in diesem Bereich profitiert (angefangen von der Metallurgie bis hin zu den Anzeigedisplays im Flugzeug).

Übertragen auf die IT-Branche könnte man das Beispiel TCP/IP heranziehen. Diese

---

8 Allein in der zweiten Hälfte der 1990er Jahre machten datenbezogene Sektoren 8,3% der gesamten US-Wirtschaft aus, wobei sie für ca. 30% des gesamten Wirtschaftswachstums sowie mindestens für die Hälfte des Anstiegs der Produktivitätsraten verantwortlich waren (U.S. Department of Commerce, U.S. Government Working Group on Electronic Commerce, „Leadership for the New Millennium: Delivering on Digital Progress and Prosperity“, 16. Januar 2001).

9 In der Tat sind US-amerikanische Bundesbehörden gesetzlich verpflichtet, die Empfänger von Zuschüssen sowie Vertragsnehmer der öffentlichen Hand zu einer Patentierung der Ergebnisse von staatlich geförderten Forschungsarbeiten anzuhalten, und die Universitäten bemühen sich aktiv um die Sicherung von Urheberrechten im Zusammenhang mit ihren Forschungsarbeiten. Mindestens seit 1980 verfolgt die Regierung der USA eine strikte Politik der Bereitstellung von Ergebnissen aus von ihr geförderten technologischen Forschungen an die Industrie, um so deren kommerzielle Nutzung sowie Innovationen zu fördern. Siehe hierzu den Bayh-Dole Act von 1980, den Stevenson-Wydler Technology Innovation Act von 1980, den Federal Technology Transfer Act von 1986, Executive Order 12591 von 1987, den National Technology Transfer and Advancement Act von 1995 und den Technology Transfer Commercialization Act von 2000.

10 Von 1969 bis 1994 beliefen sich die US-amerikanischen Investitionen in Forschung und Entwicklung im Hochtechnologiebereich auf 77,6 Milliarden US-Dollar seitens der Regierung und 262 Milliarden US-Dollar seitens der privaten Industrie (Industrial Research and Development Information System (IRIS) der National Science Foundation, Stand vom 11. Oktober 2002 unter <http://www.nsf.gov/sbe/srs/iris/start.htm>).

11 Ein gutes Beispiel hierfür ist Google, Inc. Google war als eines von 15 Phase-1-Projekten der Universität Stanford im Rahmen der *Digital Libraries Initiative* staatlich gefördert worden. 1996 wurde die Technologie gegenüber dem Office of Technology Licensing (OTL) der Universität Stanford offen gelegt. 1998 erteilte das OTL Sergey Brin und Larry Page die Erlaubnis zur Gründung einer kommerziellen Struktur auf Basis der Technologie. Heute ist Google, Inc. ein erfolgreiches Unternehmen, das Einkünfte sowohl für sich als auch für die Universität Stanford generiert.



Technologie ging aus einem staatlichen Forschungsprojekt hervor, das in den Universitäten nach dem OSS-Modell weiterentwickelt und schließlich zu einem offenen Industriestandard wurde. Seitdem wurde sie laufend verbessert und hielt durch proprietäre Implementierungen kommerzieller Softwareunternehmen wie Novell, Apple, IBM und Microsoft in den meisten Rechnern Einzug.

Das Betriebssystem Windows wurde hingegen von Microsoft selbstständig und gewinnorientiert entwickelt. Das Produkt verwendet dennoch viele Komponenten, die aus akademisch oder staatlich geförderten Projekten entstanden sind, sowie Dutzende offener Industriestandards. Ferner bot die Veröffentlichung unzähliger Schnittstellen für die Anwendungsentwicklung Geschäftsmöglichkeiten für eine Vielzahl von Softwareunternehmen, aus denen wiederum auf ganz bestimmte Anforderungen ausgelegte, spezifische Anwendungen entstanden.

Was bedeutet dies nun also für die Zukunft? Wenn man von der Vergangenheit ausgeht, wird die Zukunft der Software nicht das Ergebnis eines einzelnen Entwicklungs-, Lizenz- oder Geschäftsmodells sein. Zukünftige Innovationen werden nicht allein durch den Staat, die Privatindustrie oder lockere Zusammenschlüsse von Einzelpersonen erreicht werden, die zum Wohl der Gesellschaft arbeiten. Ob der Prozess der ständigen Innovationen, wie er bereits über 30 Jahre anhält, auch weiter erfolgreich sein kann, hängt einzig und allein davon ab, ob auch weiterhin Ansätze und Technologien miteinander verschmelzen können. Letztendlich werden sowohl Konsumenten von kundenspezifischen Programmen wie auch Kunden von Paketlösungen die Nutznießer dieser Entwicklung sein, da es letztlich die Kräfte des Marktes sind, die alle Aktivitäten und Ergebnisse von Unternehmen und Einzelpersonen bestimmen.

#### 4. Das richtige Verhältnis

Angesichts der zunehmenden gegenseitigen Annäherung der Softwareanbieter und der Auswirkungen des Software-Ökosystems stellt sich für Microsoft nun die Herausforderung, ein ausgewogenes Verhältnis von mehr Transparenz, nachhaltigen Geschäftsmöglichkeiten und Investitionen in Innovationen zu finden.

OSS hat sicherlich Auswirkungen auf die Einstellung der Softwareunternehmen zur Frage des geistigen Eigentums und dessen Handhabung.<sup>12</sup> Neben den Vorteilen, die ein gemeinsam genutzter Quellcode beinhalten kann, stellt er auch eine Herausforderung für die gängigen Konzepte der Vermarktung von Software und ihrer Abgrenzung im Wettbewerb dar. Die Vorgehensweise der meisten Softwareunternehmen bestand bislang in einem strikten Schutz der geistigen Eigentumsrechte an ihren Softwareprodukten, um deren Alleinstellungsmerkmale und Wettbewerbsfähigkeit am Markt zu schützen. Die Gesetze in Bezug auf Geschäftsgeheimnisse, die viele Aspekte der Software abdecken, die durch die Kompilierung verborgen sind, spielen eine zentrale Rolle bei der Strategie zum Schutz geistigen Eigentums der meisten kommerziellen Softwareunternehmen. Früher wurde dieser Schutz entweder durch den Vertrieb

---

12 Im Rahmen der Diskussion um die Lizenzierung von Quellcode wird häufig auch über die Verwendung des Ausdrucks „geistiges Eigentum“ gestritten. Hier umfasst dieser Ausdruck das gesamte Konzept von Urheberrechten, Patenten, Geschäftsgeheimnissen und Warenzeichen.

reiner Binärversionen oder eine Quellcode-Weitergabe im Rahmen von Geheimhaltungsvereinbarungen sichergestellt. Heute müssen Unternehmen jedoch aufgrund der OSS und anderer Source-Sharing-Modelle ein ausgewogenes Verhältnis zwischen dem Schutz der geistigen Eigentumsrechte (insbesondere in Bezug auf den Schutz von Geschäftsgeheimnissen) und dem Nutzen des Kunden/Partners finden.

Durch den Wunsch nach mehr Transparenz sind Urheberrechte in den Vordergrund gerückt. Die kreative Kombination und Anwendung aller vier Optionen zum Schutz geistigen Eigentums ebnet den Weg für einen verstärkten Austausch von Quellcode, wobei die Unternehmen ihre Geschäftsgeheimnisse nun selektiv schützen können.<sup>13</sup>

Das heißt nicht, dass es in jedem Fall für ein Unternehmen angebracht ist, den Quellcode zu seinen Produkten preiszugeben und damit seine Geschäftsgeheimnisse ganz oder teilweise zu offenbaren. Kommerzielle Softwareunternehmen wägen Kundenanforderungen und -wünsche gegen eine Reihe anderer Geschäftsfaktoren ab. So fordern beispielsweise viele Investoren, dass Unternehmen ihr Vermögen mit allen Mitteln schützen sollen, um zukünftige Anlagenrenditen und stabile Einnahmen zu sichern. Das wird jeder bestätigen können, der einmal versucht hat, Kapital für ein Softwareunternehmen aufzubringen. Mitunter kann es aber durchaus sein, dass das Geschäftsgeheimnis eines Softwareproduktes tatsächlich seinen Marktvorteil ausmacht. In diesem Fall wäre das betreffende Unternehmen natürlich nicht bereit, den Quellcode hierfür preiszugeben.

Die erfolgreichsten Softwareunternehmen investieren erhebliche Summen ihrer Bruttoerträge in Forschung und Entwicklung. Microsoft investiert derzeit ca. 5 Milliarden US-Dollar jährlich, bzw. ca. 15 % des Bruttoertrags, in die Zukunft des Unternehmens.<sup>14</sup> Aber wohin führt dieser Weg? Wie ist ein ausgewogenes Verhältnis zwischen den offenkundigen Vorteilen der Quellcode-Transparenz und Flexibilität für Entwickler und den Realitäten der Softwarebranche zu erreichen, die ihre Ressourcen schützen und sichere Einnahmequellen gewährleisten muss?

Jedes Unternehmen muss hier seinen eigenen Weg finden. Für Microsoft stand fest, dass Kunden, Partner und Regierungsstellen mehr Transparenz und Flexibilität wünschten. Andererseits mussten wir auch die andere Seite der Gleichung im Auge behalten. Daraus entstand die Shared-Source-Initiative.

## **5. Die Shared-Source-Initiative**

Microsoft stellt Kunden, Partnern und Regierungsbehörden weltweit Quellcode zur Verfügung. Wir haben Quellcode-Programme auf den Markt gebracht, die weit über 100 Millionen Zeilen Quellcode liefern. Die Shared-Source-Initiative entwickelte sich aus dem Versuch, sowohl auf die Forderungen unserer Kunden und Partner nach mehr Quellcode-Zugriff einzugehen als auch die Vor- und Nachteile des OSS- und Free-Software-Ansatzes sorgfältig zu evaluieren. Schließlich haben wir gezielt unsere

---

<sup>13</sup> Die in jüngster Zeit aufgekommene Diskussion um das Thema Patente und Linux hat gezeigt, wie wichtig Klarheit in diesem Bereich ist.

<sup>14</sup> Viele Softwareunternehmen investieren 15 bis 30 Prozent ihres Bruttoertrags in Forschung und Entwicklung.

Erfahrungen mit diesen beiden Ansätzen sowie unser bestehendes Geschäftsmodell angewandt, um besser auf die Kundenbedürfnisse eingehen zu können.

Shared Source ist ein Framework und keine Lizenz.<sup>15</sup> Jedes kommerzielle Softwareunternehmen muss das Zusammenspiel aus Entwicklungsmodellen, Lizenzierung und Geschäftsmodellen auswerten, um eine erfolgreiche Strategie aufstellen zu können, bei der Quellcode gemeinsam genutzt oder veröffentlicht werden kann, wobei sowohl die Vorteile für die Kunden als auch die Wahrung der Wettbewerbsfähigkeit des Unternehmens berücksichtigt werden. Das Lizenzierungs-Konzept von Microsoft reicht daher von Rechten zur bloßen Einsichtnahme (wobei der Lizenznehmer den Quellcode von Microsoft zu Referenzzwecken und zum Debugging einsehen kann, Änderungen oder ein Weitervertrieb jedoch ausgeschlossen sind) bis hin zu weitreichenden Rechten, die dem Lizenznehmer die Prüfung und Änderung sowie den Weitervertrieb und Verkauf abgeleiteter Programme ohne Lizenzgebühren an Microsoft erlauben.

Inzwischen besitzen Hunderttausende von Entwicklern Quellcode von Microsoft. Wir haben damit das wohl kommerziell wertvollste geistige Eigentum der Softwarebranche für Tausende von Unternehmen in mehr als 60 Ländern zugänglich gemacht.<sup>16</sup> Shared-Source-Programme bieten nun Quellcode für Windows, Windows CE .NET, Visual Studio .NET, C#/CLI, ASP.NET und Passport-Technologien. Wir werden auch weiterhin den Wert des Quellcodes als wertvolle Eigenschaft unserer Produkte sowie die Möglichkeit zur Nutzung durch unsere Kunden und Partner abwägen.

Eines der größten Missverständnisse bezüglich des Shared-Source-Modells ist die Annahme, dass es auf eine einzelne Lizenz und die Ansicht des Quellcodes ohne Möglichkeit zur Änderung beschränkt sei. Tatsächlich umfasst Shared Source vier zentrale Konzepte:

*Support für bestehende Kunden* Bestehenden Kunden wird Quellcode-Zugriff gewährt, um den Produkt-Support, die Implementierung, Sicherheitstests und kundenspezifische Anwendungsentwicklungen zu erleichtern.

*Entwicklung neuer Programme* Anhand von Beispielen und Kernkomponenten wird Quellcode zu Übungszwecken bereitgestellt, um neue Entwicklungsprojekte voranzutreiben.

---

15 Die *Open Source Initiative* hat 43 Lizenzen als ihren Kriterien für Open-Source-Lizenzen entsprechend eingestuft (<http://www.opensource.org/>, Stand 20. Mai 2003). Im Zuge der fortschreitenden Vermarktung der OSS und angesichts des zunehmenden Source-Sharing durch kommerzielle Softwareunternehmen ist mit einer weiteren Verbreitung von Open-Source-Lizenzen zu rechnen, wenn immer mehr Unternehmen und Einzelpersonen festlegen müssen, wie sie ihre Urheberrechte in geeigneter Weise bereitstellen können.

16 Derzeit gibt es in der Softwarebranche keine vergleichbare Bereitstellung von Quellcode für führende Produkte. Zwar liefern viele Anbieter auf Anfrage ihren Kunden Quellcode, doch bieten nur wenige breit angelegte, entsprechende Programme. Dies dürfte sich mit der Zeit ändern, wenn die positiven Auswirkungen von OSS, Shared Source und anderen Programmen zur Bereitstellung von Quellcode allgemeine Anerkennung finden.

*Förderung von Lehre und Forschung* Quellcode und Dokumentation werden für Schulungen und Lehrmaterial sowie als Grundlage für weiterführende Forschung bereitgestellt.

*Förderung neuer Geschäftsmöglichkeiten* Lizenzstrukturen und Quellcode werden an Partner bereitgestellt, um für beide Parteien vorteilhafte neue Geschäftsmöglichkeiten zu fördern.

Zum Zeitpunkt des Verfassens dieser Abhandlung wurde für fünf Produktgruppen von Microsoft Quellcode bereitgestellt, der bestimmte Optionen zum Ändern des Codes und Erstellen abgeleiteter Arbeiten umfasst.<sup>17</sup> Drei der Produktgruppen bieten allgemeine Rechte, um abgeleitete Entwicklungen zu erstellen und kommerziell zu vertreiben. Das bedeutet, dass der Programmierer den Code erhält, ändert und unter einer herkömmlichen kommerziellen Binärcode-Lizenz gewinnbringend weiterverbreitet, ohne dafür Zahlungen an Microsoft entrichten zu müssen. Alle aktuellen Quellcode-Programme werden von Microsoft kostenlos zur Verfügung gestellt.<sup>18</sup>

## **6. Aufbau eines Shared-Source-Programms**

Für die Einführung der verschiedenen Shared-Source-Programme hat Microsoft erhebliche Ressourcen aufgewandt. Die Bekanntgabe jedes einzelnen Quellcodes basiert auf Entscheidungen, die ein ausgewogenes Verhältnis zwischen Kunden- und Geschäftsvorteilen zum Ziel haben. Zusätzlich haben wir in technische Ressourcen investiert, um erweiterte Tools und Dokumentation zu liefern, die den Nutzen des Quellcode-Zugriffs steigern.

In den Tabellen 1 bis 4 findet sich eine kleine Auswahl der Fragen und Aspekte, die bei Microsoft vor dem Release eines Codes berücksichtigt werden. Diese Übersicht stellt natürlich kein vollständiges Analysetool dar, sondern lediglich ein Beispiel für den Prozess der Entscheidungsfindung, wie er bei einem kommerziellen Softwareanbieter für ein Quellcode-Programm ablaufen kann.

Zusammenfassend lässt sich sagen, dass es für eine durchdacht konzipierte Bereitstellung von Quellcode eine Reihe primärer Zielsetzungen zu erfüllen gilt:

- Es muss ein anschaulicher Einblick in das offengelegte Produkt oder Projekt gegeben werden. Ein Wert entsteht häufig schon allein durch die Zugänglichkeit und Analyse des Codes, ohne diesen zu verändern.

---

17 Die Windows Academic Shared-Source- und OEM-Lizenzen ermöglichen Forschern das Erstellen zeitweiliger Änderungen des Windows-Quellcodes für Forschungs- und Erprobungszwecke. Alle anderen Gruppen mit Windows-Source-Zugriff (Unternehmenskunden, Systemintegratoren und Regierungsbehörden) besitzen lediglich Rechte zur Einsichtnahme, d. h. sie können den Quellcode zwar anzeigen und debuggen, aber keine abgeleiteten Entwicklungen aus ihm erstellen.

18 Aufgrund der zeitlichen Verzögerung zwischen dem Verfassen und der Veröffentlichung dieses Dokuments wurden keine spezifischen Details zu den Programmen aufgeführt. Weitere Informationen zur Verfügbarkeit von Quellcode von Microsoft unter <http://www.microsoft.com/sharedsource/>.

Fragen	Aspekte
Welche Community soll diesen Quellcode erhalten?	Nicht alle Quellcode-Releases müssen vollständig und allgemein verfügbar sein. In bestimmten Fällen kann eine Zusammenarbeit mit eingeschränkten Communities, großen Kunden, wichtigen Partnern oder bestimmten Regierungsbehörden besser geeignet sein.
Worin besteht der Vorteil dieses Quellcodes für diese Unternehmen und Einzelpersonen?	Ein Quellcode deckt nicht alle IT-Belange ab. Daher ist die korrekte Einschätzung seines Wirkungsbereichs ein kritischer Faktor bei der Festlegung der Vorgehensweise für Rechtevergabe und Bereitstellung.
Wie viele Personen werden über den Quellcode verfügen und wie gestaltet sich die Interaktion mit diesen?	Weitreichende Programme können umfangreiche Ressourcen für die Bereitstellung des Quellcodes und/oder die Einbeziehung der Community erfordern. Dabei geht es nicht nur um logistische Belange wie beispielsweise die Download-Kapazität. Es muss u. a. das Ausmaß an technischem Aufwand, Feedback-Verarbeitung und laufenden Investitionen berücksichtigt werden.
Welche geografischen Gebiete kommen für den Quellcode-Zugriff infrage?	Abgesehen von offensichtlichen Aspekten wie der Lokalisierung der Dokumentation, bestehen von Land zu Land auch erhebliche rechtliche Unterschiede, die es zu berücksichtigen gilt. Die Handhabung von Fragen des geistigen Eigentums variiert sehr stark, weshalb sich hier eine spezielle Rechtsberatung empfiehlt. <sup>19</sup>

*Tabelle 1: Definition der Zielsetzungen*

- Es müssen entsprechende Tools, Dokumentation und Support-Leistungen bereitgestellt werden, um den Wert für die mit dem Code arbeitenden Einzelpersonen zu erhöhen, insbesondere bei Programmen mit Rechten für abgeleitete Entwicklungen.
- Es müssen Methoden für die Verarbeitung von Feedback festgelegt werden, um so Verbesserungen der Code-Basis zu erleichtern.
- Es muss diejenige Community bestimmt werden, die von dem Zugriff auf den Quellcode den größten Vorteil hat.
- Es müssen Rechte festgelegt werden, die den Urheber des Quellcodes sowie alle an ihm arbeitenden Beteiligten schützen.

<sup>19</sup> Dies gilt für alle Open-Source-Modelle – OSS, Shared Source u. a. Beispielsweise basieren viele der populärsten OSS-Lizenzen auf im US-amerikanischen Urheberrechtssystem verankerten Konzepten. Da der Code weltweit genutzt wird, variieren die angewandten rechtlichen Grundlagen erheblich.

Fragen	Aspekte
Welcher Quellcode soll verfügbar gemacht werden?	Genau wie es sich bei der Community, für die der Quellcode bereitgestellt werden soll, nicht unbedingt um 100 Prozent der Bevölkerung handelt, muss der bereitgestellte Quellcode auch nicht 100 Prozent eines Produkts darstellen. Bestimmte Bestandteile eines Produkts sind extrem wertvoll, unter Ausschluss einer Offenlegung des Quellcodes an Dritte lizenziert oder von Exportbeschränkungen der jeweiligen Regierung betroffen.
Sind die Rechte zur Offenlegung der jeweiligen Source vorhanden?	Kommerzielle Software enthält oft Bestandteile, die ursprünglich in einem anderen Zusammenhang entwickelt wurden und innerhalb eines umfangreicheren Produkts wiederverwendet oder lizenziert werden. Je mehr Inhaber von Urheberrechten für ein bestimmtes Software-Segment vorhanden sind, desto komplexer wird die Bereitstellung des Quellcodes kommerzieller Produkte.
Wurde der Code für die allgemeine Verwendung bereinigt und wurde die Qualität der Kommentare überprüft?	Innerhalb des Quellcodes finden sich Kommentare der Entwickler, die einen Einblick in deren Gedankengänge liefern sollen. Es kann äußerst hilfreich sein, wenn besonders komplexe Code-Segmente mit Kommentaren versehen sind. Leider enthält der Quellcode häufig sehr lockere Formulierungen, deren öffentliche Bereitstellung erst überprüft werden sollte.
Ist eine Strategie für die Verwaltung des Quellcodes vorhanden, die Bug-Fixes und zukünftige Versionen im Zusammenhang mit der offengelegten Source vorsieht?	Für die meisten erfolgreichen Softwareentwicklungsprojekte bestehen ausgereifte Verwaltungsprozesse in Hinblick auf den Quellcode. Soll ein bislang geschützter Code an eine breitere Basis von Entwicklern bereitgestellt werden, ist ein Prozess erforderlich, um laufende intern durchgeführte Entwicklungsarbeiten für die Community verfügbar zu machen, die nun den Quellcode besitzt.
Wie werden eingereichte Vorschläge oder Code-Fixes gehandhabt?	Genau wie bei der Bereitstellung von neuem intern entwickelten Code muss ein Prozess erstellt werden, der die Entgegennahme von Vorschlägen, Bug-Fixes und neuen Funktionen von Seiten der Community regelt, die Zugriff auf den Code hat. Dabei sollten auch die rechtlichen Konsequenzen und Risiken im Zusammenhang mit der Einbeziehung eingereicherter Vorschläge oder Code-Fixes in die Code-Basis erwogen werden.

Tabelle 2: Verwaltung des Quellcodes

Fragen	Aspekte
Welche Rechte sollen für Anzeige, Debugging, Änderung und Vertrieb vergeben werden?	Obgleich der rechtliche Aspekt bei der Festlegung von Lizenzierungsrechten für Quellcode eine wichtige Rolle spielt, kommt es doch vor allem auf die Wünsche der Kunden und Partner an. Entscheidend ist, wie diese von dem Quellcode profitieren können. Auch die treuhänderische Verantwortung gegenüber den Aktionären im Hinblick auf den Schutz des geistigen Eigentums ist ein kritischer Faktor, der allerdings erst sekundär zum Tragen kommen sollte. Ein erfolgreiches Quellcode-Lizenzierungsprogramm schafft einen Ausgleich zwischen den genannten Faktoren zum Wohle aller Beteiligten.
Kann ein kommerzieller Weitervertrieb auf Basis eingeräumter abgeleiteter Rechte erfolgen?	Microsoft hat sich dafür entschieden, zwei verschiedene Ansätze hinsichtlich der Lizenzen für abgeleitete Entwicklungen zu implementieren, die unterschiedliche Geschäftsziele innerhalb unseres Programms zur Lizenzierung von Quellcode widerspiegeln. Die kommerzielle Nutzung abgeleiteter Entwicklungen ist ein zentraler Punkt in der Bereitstellung von Quellcode durch Anbieter kommerzieller Software.

Tabella 3: Lizenzierung

## 7. Erfahrungen und Prognosen

Die wichtigste Erkenntnis, die wir aus dem Shared-Source-Prozess gewonnen haben, ist die Tatsache, dass der Quellcode eine Produkteigenschaft darstellt. Diese Eigenschaft wird zwar von vielen Anwendern nie genutzt werden, doch bedeutet allein schon die Möglichkeit ihrer Nutzung einen Mehrwert. Unsere Kunden und Partner, die über den Quellcode verfügen und diesen aktiv nutzen, berichten, dass er einen unschätzbaren Mehrwert bei der Verwendung unserer Produkte darstellt. Bisher stellt diese Gruppe allerdings im Vergleich zu allen Einzel- und Unternehmensanwendern von Microsoft-Produkten einen verschwindend kleinen Anteil dar.

Die Shared-Source-Initiative besteht nun seit drei Jahren, wobei Microsoft jedoch bereits seit mehr als 13 Jahren Quellcode an akademische Institutionen und OEMs liefert. Vor 2001 lief der Austausch von Quellcode nur mit einem begrenzten Teilnehmerkreis und weniger formell als heute ab. Wir haben jedoch aus dem Dialog mit unseren Kunden sowie aus OSS-Modellen gelernt, dass es bei dieser Initiative vor allem um das richtige Verhältnis geht. In vieler Hinsicht befindet sich Shared Source immer noch seiner Version-2.0-Phase. Der Erfolg der bisherigen Programme hat uns gezeigt, wie wichtig eine Ausweitung der Initiative auch für anderen Code ist.

In absehbarer Zukunft wird die Lizenzierung von Quellcode auch weiterhin ein heikles Thema in der Branche darstellen. Grundlegende Fragen zur Rolle des geistigen

Eigentums für zukünftige Innovationen sind zu stellen. Im Zentrum der Shared-Source-Initiative steht die Annahme, dass das geistige Eigentum und dessen Schutz die Grundlage für den Erfolg eines fortwährenden Innovationskreislaufs sind. Die Bekanntgabe von Quellcode hat zwar sehr viele Vorteile, sie darf aber nicht zum Abschluss einer erfolgreichen Softwarebranche führen. Darüber hinaus kann sie nicht als Wundermittel für alle Belange der Informationstechnik gesehen werden. Es werden daher auch weiterhin verschiedene Modelle nebeneinander existieren. Shared Source ist Teil eines großen Mosaiks, wobei wir uns glücklich schätzen, ein paar von dessen Steinchen zusammenfügen zu dürfen.



Fragen	Aspekte
Wie erfolgt die Bereitstellung des Quellcodes? Wird einfach ein Quellcode-Paket veröffentlicht oder entsteht ein Mehrwert durch ein Bereitstellungstool?	Es gibt zahlreiche Möglichkeiten für die Bereitstellung von Quellcode. Die OSS-Community verfügt über eine Reihe von Websites wie z. B. SourceForge von VA Software Corp., um Quellcode bereitzustellen und Projekte verwalten zu können. Ein Unternehmen kann seinen Quellcode aber auch selber bereitstellen, wie Microsoft dies mit seinem Projekt GotDotNet WorkSpaces getan hat. Microsoft hat zudem eine sichere Web-Infrastruktur für die Bereitstellung von Windows-Quellcode erstellt: MSDN Code Center Premium. Andere Gruppen innerhalb von Microsoft haben sich für einfache Web-Downloads von Source-Dateien entschieden, wobei die Wahl des Toolsets und der Entwicklungsumgebung dem einzelnen Entwickler überlassen bleibt. Die Bestimmung des Umfangs der beteiligten Community sowie des Volumens des in dem jeweiligen Release enthaltenen Quellcodes ist ein wichtiger Faktor für die Wahl der Bereitstellungsmethode.
Wie soll die Community aus Einzelpersonen und Unternehmen mit Zugriff auf den Quellcode einbezogen werden?	Auch wenn eine Community aus Entwicklern sich für eine bestimmte Source-Basis spontan zusammenfinden kann, umfassen erfolgreiche Programme dennoch eine Beteiligung der für die Erstellung des Codes verantwortlichen Entwickler. Darüber hinaus sorgt ein durchdachtes Projektmanagement dafür, dass die Community auf produktive Weise einbezogen wird.
Wurde eine zusätzliche Support-Dokumentation für die Source-Basis erstellt?	Das Erstellen einer hochwertigen Softwaredokumentation hat sich als eines der teuersten und schwierigsten Probleme in dieser Branche erwiesen. Je mehr Informationen den mit einer bestimmten Source-Basis arbeitenden Entwicklern bereitgestellt werden, desto besser. Obwohl dies nicht zwingend erforderlich ist, wird dadurch sicherlich die Qualität des Source-Sharing-Programms als Ganzes verbessert.

*Tabelle 4: Umsetzung*