

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.

Coases Pinguin beginnt zu fliegen – Der institutionelle Wandel in der Softwareindustrie

MATTHIAS BÄRWOLFF



(CC-Lizenz siehe Seite 463)

Dieser Artikel behandelt den Wandel der Prozesse und Strukturen in der kommerziellen Softwareindustrie bedingt durch den wachsenden Einfluss von Open Source als „Produktionsparadigma“. Der zu beobachtende institutionelle Wandel lässt sich als effizienzsteigernde Anpassung der Produktions- und Nutzungsstrukturen im Bereich Software begreifen.

1. Einleitung

Als Coase 1937 in seinem bahnbrechenden Artikel „The nature of the firm“ die Existenz der Firma als ökonomische Reaktion auf die „Kosten der Nutzung des Preissystems des Marktes“ – seither Transaktionskosten genannt – zurückführte, erschuf er die bis heute allgemein akzeptierten ökonomischen Grundmodi Markt und Firma. Dieser Artikel will der häufig aufgeworfenen Frage nach der Einordnung von Open Source relativ zu diesen beiden Modi nachgehen.

Einerseits kann man die Entwicklung von Open-Source-Software (OSS) im analytischen Rahmen der *Firma* auffassen, bei dem die Beziehungen der Mitwirkenden durch Open-Source-Lizenzverträge geprägt sind und eine Struktur durch „psychologische Verträge“ (Rousseau 1989) geschaffen wird. Andererseits wird jedoch oft auch der *Markt* als Analogie für die „offene Evolution“ bei Open-Source-Software verwendet.¹ Genau genommen passt Open Source eigentlich in keines der beiden Raster, tatsächlich bewegt es sich inner- und außerhalb des Kontinuums Markt-Firma, ohne sich dabei allzu leicht festlegen zu lassen.

Der Titel wie auch das Thema dieses Artikels sind weitgehend motiviert durch Benklers Artikel „Coase’s Penguin, or, Linux and *The Nature of the Firm*“ (2002). Dieser nutzt Coases Transaktionskostentheorie² als methodologischen Hintergrund für seine Analyse und befasst sich mit der Frage, warum Open Source als Produktionsparadigma so erfolgreich sein kann, wenn weder Preissignale noch Anweisungen für eine

1 Siehe der Beitrag von Bauer und Pizka in diesem Buch. Ökonomisch gesehen macht die von ihnen aufgegriffene Analogie nur bedingt Sinn, da bei der Entwicklung von OSS keine Preissignale zur Steuerung der Ressourcenverteilung zum Tragen kommen.

2 In der englischsprachigen Literatur findet man hierfür den Begriff *transaction costs theory* (TCE).

effiziente Ressourcenverteilung zum Tragen kommen. Benkler schließt daraus, dass es sich bei Open Source um einen „entstehenden neuen dritten Modus der Produktion“ handelt. Demil und Lecocq (2003) nennen diesen Modus, der sich durch die Bedingungen von Open-Source-Lizenzen konstituiert, „*bazaar governance*“, in Anlehnung an Raymonds berühmten Aufsatz „The Cathedral and the Bazaar“.

Es ist unzweifelhaft, dass dieser „dritte Modus“ 2005, also drei Jahre später, die kommerzielle Softwareindustrie erfasst hat.³ Selbst Firmen wie Microsoft, die traditionell fast keine strategischen Überschneidungen mit Open Source haben, erkennen dessen Bedeutung und befassen sich intensiv damit, ihr Geschäftsmodell zumindest teilweise den veränderten Rahmenbedingungen anzupassen.⁴ Der Wandel in der Softwareindustrie, bedingt durch Open-Source-Software, ist also unübersehbar. Dieser Artikel soll sich also der Frage widmen, wie man diesen Wandel ökonomisch begreifen und sinnvoll fassen kann. Hierzu wird ein Blick auf die relevanten ökonomischen Institutionen sowie deren Evolution und Effizienz geworfen. Speziell wird der Frage nachgegangen, wie Open Source die ökonomische Rationalität der kommerziellen Akteure beeinflusst und verändert.

2. Institutionen im Bereich Software

Die wichtigsten Institutionen⁵ im Bereich von Softwareentwicklung und -nutzung, und damit auch der Transaktionen, die gemeinhin mit Software in Verbindung stehen, sind rechtlicher und technischer Natur. Nicht ohne Grund konzentriert sich die aktuelle Diskussion im Bereich „Internet Governance“ primär auf rechtliche und technische Regulierungsaspekte bei Software. Die Bedeutung rechtlicher Rahmenbedingungen wird schon alleine durch die konstituierende und äußerst umstrittene Rolle von Urheberrecht und Patentrecht für Software deutlich. Lessig (1999) hat außerdem die generell zunehmende Bedeutung von *Code* als „regulierende Instanz“, also als Institution, identifiziert.

Beide institutionellen Bereiche, Recht und Code, werden, sowohl für sich selbst als auch in ihren Wechselwirkungen, in der aktuellen Literatur relativ umfassend behandelt. Dabei wird primär auf die tatsächlichen und potentiellen negativen Effekte für die Verbraucher und die allgemeine Wohlfahrt eingegangen.⁶ Zumeist wird bei diesen Betrachtungen das bestehende Regime der Institutionen des „geistigen Eigentums“

3 So bemerkt ein Analyst treffend: „The impact of open source in the enterprise is undeniable“ (Scot Petersen, „Open Source Is Showing Its Value“, eWeek, 14. Februar 2005, <http://www.eweek.com/article2/0,1759,1763034,00.asp>).

4 Siehe der Beitrag von Seemayer und Matusow in diesem Buch.

5 Zur Definition von Institutionen sagt North (1990, S. 4): „Institutions define and limit the set of choices of individuals. Institutional include any form of constraint that human beings devise to shape human interaction. They can be [...] both [...] formal [...] and [...] informal.“ Siehe zum Begriff Institution auch einen der maßgeblichen Begründer der Institutionenökonomie Commons (1931).

6 Lessig (1999) hat in diesem Zusammenhang insbesondere das Argument populär gemacht, dass Regulierung durch Code ohne Einflussmöglichkeiten für die Verbraucher, wenn man so will, „hinter ihrem Rücken“ stattfindet und dadurch letztendlich einen negativen Einfluss auf die allgemeine Wohlfahrt hat. Als Beispiel führt er unter anderem die derzeit zu beobachtenden Abweichungen vom ursprünglichen *End-to-End-Design* des Internets an.

Coases Pinguin beginnt zu fliegen

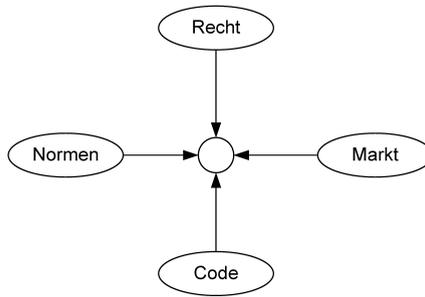


Abbildung 1: „Punkt-Modell“ von Lessig (1999)

als inadäquat und hinderlich für den technischen Fortschritt, die günstige Entwicklung im Bereich *Code* also, identifiziert, der speziell in diesem Bereich auf sequentiellen und verteilten Innovationen beruht.⁷ Der institutionelle Einfluss von Code als Einschränkung von Handlungsspielräumen von Akteuren wird also erkannt, die Lösung des daraus folgenden Problemkomplexes jedoch praktisch ausschließlich im Bereich des Rechts gesehen.

Angesichts der aktuell zu beobachtenden Entwicklungen stellt sich jedoch die Frage, ob die Institution Recht tatsächlich den entscheidenden strukturellen Einfluss auf Softwareentwicklung und -nutzung hat, wie dies bislang unterstellt wurde, oder ob sich dieser im Zuge eines institutionellen Wandels verringert. Um uns einer möglichen Antwort zu nähern, ist es sinnvoll, in Anlehnung an Commons (1931) zwei weitere institutionelle Bereiche zu betrachten: *Ökonomie*, oder besser *wirtschaftliche Rationalität*, und *Ethik*, oder auch Normen. Lessig (1999) hat in „Code and Other Laws of Cyberspace“ diesen Gedanken mit einem einfachen „Punkt-Modell“ aufgegriffen (Abbildung 1). In diesem finden sich Code, Recht, Markt und Normen als „regulierende Modalitäten“.⁸ Lessig betont, dass die einzelnen „Regularien“, oder, wenn man so will, Institutionen, sich gegenseitig beeinflussen und in diesem Prozess einer Evolution unterliegen. Dieser Gedanke greift unter anderem auf North (1990) zurück, der deutlich gemacht hat, dass Organisationen und Institutionen sich wechselseitig beeinflussen.

In Abbildung 2 versuchen wir, das Zusammenspiel der Institutionen im Bereich Softwareentwicklung und -nutzung so darzustellen, wie wir es insbesondere im Hinblick auf die zentrale Rolle von *Code* im Folgenden betrachten wollen. Letztlich konzentrieren wir uns in diesem Modell auf die Wechselwirkungen von Code mit anderen Institutionen, speziell mit *ökonomischer Rationalität*. Dabei begreifen wir Code sowohl als Institution als auch als ökonomisches Gut, weil es als solches einen signifikanten Ein-

⁷ Siehe hierzu die Beiträge von Bessen und Maskin sowie von Hippel in diesem Buch.

⁸ Lessigs Modell ist in sofern sehr allgemein, als dass der konkrete Gegenstand der Regulierung weitgehend im Unklaren bleibt. Auch die Komponente „Markt“ ist sehr unspezifisch. Die Beispiele, die Lessig zu deren Illustration heranzieht, deuten vielmehr auf den Aspekt der Preisregulierung durch Steuern als „regulierende Modalität“ hin.

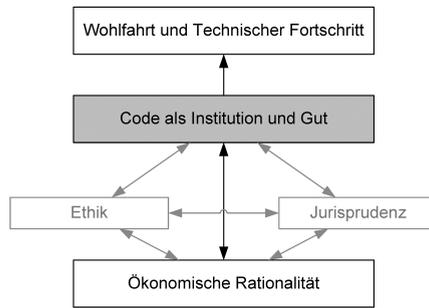


Abbildung 2: Die Beziehung von Code und ökonomischer Rationalität

fluss auf ökonomische Wohlfahrt und Fortschritt hat. Der Charakter von Code ist also ein dualer, zum einen hat er Einfluss auf Organisationen und andere Institutionen, zum anderen beeinflusst er die allgemeine Wohlfahrt und den technischen Fortschritt.

3. Die Effizienz der Institutionen im Hinblick auf Code

Praktisch seit der Einführung von Urheber- und Patentrecht wurden deren institutionelle Auswirkungen auf die Entwicklung und Nutzung von Software kritisiert.⁹ Dabei bezieht sich die Kritik an den bestehenden Institutionen nicht allein auf statische Wohlfahrtserwägungen, sondern primär auf das Argument, dass technologischer Fortschritt basierend auf sequentiellen Innovationen durch die bestehenden Institutionen erschwert und behindert wird.¹⁰ Insbesondere wurde von vielen Beobachtern angemerkt, dass die ökonomische Rationalität, bedingt durch Urheber- und Patentrecht, einen negativen Einfluss auf die sequentielle Entwicklung im Bereich Software hat: Die Maximierung privater Interessen mittels Durchsetzung von Ausschließlichkeitsrechten, gewährt durch das Urheber- und Patentrecht, steht in unauflösbarem Konflikt mit den Interessen der Allgemeinheit.

Während die empirischen Befunde für die Gültigkeit dieser Argumentationen lange Zeit ausstanden (Liebowitz und Margolis 1999), stellt spätestens der Erfolg von Open-Source-Software die bestehenden rechtlichen Institutionen hinsichtlich ihrer Effizienz ernsthaft in Frage. Der Bestand einer solchen Ineffizienz wird jedoch nach North (1990) zumindest teilweise bedingt durch das Wechselspiel zwischen bestehenden Institutionen und Organisationen, die einem institutionellen Wandel entgegenwirken. Es ist also mithin schwierig, bestehende Ineffizienzen aufzulösen, wenn Organisationen wie große Firmen davon profitieren.

Die meisten Betrachtungen beklagen zwar eben diesen Umstand, übersehen jedoch den zunehmenden institutionellen Charakter von Code, speziell in der Ausprägung

⁹ Siehe z. B. Samuelson (1984).

¹⁰ Siehe der Beitrag von Bessen und Maskin in diesem Buch.

Open Source, für die ökonomische Rationalität der Beteiligten.¹¹ Man kann Open Source, im Sinne von Romer (1993), als „Meta-Idee“ begreifen, die die ökonomische Rationalität bei der Produktion, Nutzung und Weitergabe von Code grundlegend verändert und damit qualitativ neues, zuvor nicht da gewesenes Wachstum erzeugt.¹² Diese veränderte Rationalität wäre Ausdruck der „adaptiven Effizienz“ (North 1990), die Reaktion auf veränderte Gegebenheiten durch die Existenz von Open Source.

Tatsächlich verdichtet sich dieser Eindruck, wenn man die aktuellen Markttendenzen beobachtet. So bemerkt beispielsweise John Terpstra, einer der Hauptentwickler von Samba und Redner auf der diesjährigen *LinuxWorld* in Boston: „Whatever the outcome of current IP [intellectual property] litigation involving Linux and open source, the market will move forward“.¹³ Damit deutet er auf eben diesen Trend hin: Open Source spielt für immer mehr kommerzielle Firmen eine bedeutende Rolle, die unweigerlich den Kern ihres Geschäftsmodells berührt.

4. Der Einfluss von Open Source auf ökonomische Rationalität

Angesichts der steigenden Relevanz von Open-Source-Software für kommerzielle Unternehmen, nicht nur bezüglich deren Anwendung, sondern insbesondere auch der Mitwirkung an deren Produktion, sollte es möglich sein, ökonomische Erklärungsmuster für diesen Wandel zu identifizieren.

Open Source wird als „Produktionsparadigma“ vollständig definiert durch Open-Source-Lizenzen.¹⁴ Mit solchen Lizenzen wird eine Institution erzeugt, die hauptsächlich vertragsrechtliche, zu einem gewissen Grade auch ethische, Rahmenbedingungen für die Schaffung, Nutzung und Weitergabe von Open-Source-Software festlegt.

Es wird häufig unterstellt, dass Open Source für kommerzielle Firmen, oder überwiegend „extrinsisch motivierte Akteure“ nur von marginalem Interesse sein kann, da Open-Source-Software selbst eine direkte Aneignung von Mehrwert durch Transaktionen basierend auf herkömmlichen exklusiven Eigentumsrechten scheinbar ausschließt. Schließlich ist Exklusiveigentum an Open-Source-Software durch die Lizenzbedingungen, der sie unterliegt, praktisch unmöglich. So stellen beispielsweise Osterloh et al. (2004) in Anlehnung an Fehr und Falk (2001) die These auf, dass das Vorhandensein von „intrinsischer Motivation“ eine unabdingbare Voraussetzung für erfolgreiche Open-Source-Projekte darstellt.¹⁵ Und auch Weber (2004) widmet sich

11 Die institutionelle Bedeutung von Open Source wird sehr gut illustriert durch die teilweise haarsträubend scharfen Attacken von Microsoft gegen Open Source und speziell die GPL am Anfang des 21. Jahrhunderts.

12 So bemerkt Paul Romer: „Open Source erinnert uns daran, dass der Markt nicht die einzige Lösung ist“ (zitiert in Ludwig Siegele, „Lieber Ruhm im Netz als Rubel im Sack“, *Die Zeit*, Dezember 2000, http://zeus.zeit.de/text/archiv/2000/12/200012.open_content.xml).

13 Jan Stafford, „LinuxWorld preview: Open source rules, SCO fades and apps abound“, *SearchEnterpriseLinux.com*, 9. Februar 2005, http://searchenterprise-linux.techtarget.com/originalContent/0,289142,sid39_gci1052513,00.html

14 Siehe die Webseite der *Open Source Initiative* (OSI, <http://www.opensource.org>) zu den verschiedenen existierenden Open-Source-Lizenzen.

15 Es soll hier angemerkt werden, dass die Dichotomie extrinsische und intrinsische Motivation keineswegs

in seiner exzellenten Abhandlung über die Wirkungsweise von „klassischen“ Open-Source-Projekten, wie Linux und Apache, nur am Rande der Bedeutung kommerzieller Akteure für Open Source.

Es mehrt sich jedoch die empirische Evidenz dafür, dass Reputationseffekte und damit einhergehende Erhöhungen des individuell erzielbaren Lohnniveaus für einzelne Entwickler sehr wohl eine wichtige extrinsische Motivation für die Mitarbeit an Open-Source-Software sein können (Lerner und Tirole 2000, Hann et al. 2004). Außerdem lässt sich die unübersehbare Mitwirkung von kommerziellen Firmen zweifellos *nur* „extrinsisch“ erklären. Es scheint mithin praktikabel zu sein, wenigstens ökonomische Rationalität im Sinne von Jevons (1923) für unsere Ausführungen anzunehmen.¹⁶

Warum also wirken kommerziell motivierte Firmen, denen wir ökonomisch rationales Verhalten unterstellen, an Open-Source-Entwicklung mit? Eine tragfähige und weitreichende Erklärung, die auch Boyle (2000) aufgegriffen hat, ist, dass Informationsgüter nur selten in vollständiger Isolation von dinghaften Gütern oder nachgelagerten Dienstleistungen auftreten:¹⁷

„Information goods do not exist in isolation. The good [...] comes “bundled” with a large number of other customer needs, social implications, market effects, and business opportunities. These linked or bundled phenomena may well be excludable to a greater degree than the information good itself.“ (S. 2015)

Koenig (2004) laboriert entsprechende aktuelle Open-Source-Strategien von Softwarefirmen und führt als Beispiele unter anderem Firmen wie Zope, JBoss und Red Hat an, die ihre Softwareprodukte als Open-Source-Software handhaben, Dienstleistungen wie Anpassung, Beratung, Schulung oder Zertifizierung jedoch als kommerzielle Dienstleistungen anbieten.

Open Source kann also, wenn wir den Bogen von der Diskussion von Geschäftsmodellen etwas weiter spannen, als Infrastruktur aufgefasst werden, auf der weitere Produkte oder Dienstleistungen aufsetzen.¹⁸ Die Erkenntnis, dass viele Softwarepro-

so trivial ist, wie es auf den ersten Blick scheinen mag. Viele scheinbar intrinsische Handlungen von Personen schließen extrinsische Motivation nicht aus.

16 Es kann zweifellos als gänzlich unproblematische Annahme gelten, dass Akteure in ökonomischen Systemen im Allgemeinen bessere Situationen den schlechteren vorziehen. Diese Annahme ist unabhängig von der empirischen Beobachtung, dass es keine vollkommenen Märkte gibt, auf denen sämtliche Akteure zu jedem Zeitpunkt vollständig informiert sind und in unendlich kurzer Zeit ihren Präferenzen entsprechend optimale Transaktionen tätigen.

17 Siehe auch Hirshleifer (1971) für eine frühere Elaboration.

18 Michael Tiemann („How Linux will Revolutionize the Embedded Market“, Linuxdevices.com, 22. Mai 2002, <http://linuxdevices.com/articles/AT7248149889.html>) trifft in diesem Zusammenhang eine wichtige Unterscheidung zwischen Open-Source-Software als Projekt und Open-Source-Software als Plattform:

„Projects are organized around specific, though possibly far-reaching, technical goals (such as running on a laptop, in a cluster, in a router, on a mainframe, on a particular chip architecture, etc.). A platform, on the other hand, provides a common environment that can be widely deployed and serviced. Linux itself is a project (it moves forward largely

dukte letztlich Plattformen bzw. Infrastrukturen begründen, ist fundamental für das Verständnis von Wertschöpfung im Bereich Software. Nicht nur sind Anwendungen von Infrastrukturen im Vorhinein weitgehend unspezifiziert und offen, es ist sogar möglich, dass Software als Infrastruktur für weitere Infrastrukturen dient, etwa eine Datenbank oder ein *Content Management System* auf einem Betriebssystem.

Frischmann (2004) führt in seiner Elaboration zu den ökonomischen Eigenschaften von Infrastrukturen aus, dass eben durch die Unspezifiziertheit und damit Unvorhersehbarkeit der Nutzungen ein auf privaten Exklusivrechten und Markttransaktionen basierendes Eigentumsregime ökonomisch ineffizient ist. Effizienter ist es, Frischmann zufolge, solche Infrastrukturen als Allmenden, engl. *Commons*, zu verwalten. Er greift damit einen Gedanken auf, den erstmals Rose (1986) in dieser Form postuliert hat.¹⁹ Zope, JBoss und RedHat zeigen die kommerzielle Sinnhaftigkeit eines solchen Vorgehens und damit dessen ökonomische Rationalität.

Ein in diesem Zusammenhang weiterer wichtiger Faktor, der die Entstehung von Allmendegütern gerade bei Softwareprodukten mit Infrastruktureigenschaften begünstigt, ist deren zunehmender *Commodity*-Charakter. Eine Software, deren Schnittstellen und Funktionalitäten weitgehend spezifiziert und standardisiert sind, ist relativ leicht nachahmbar.²⁰ Da die Grenzkosten für Software bekanntermaßen bei Null liegen, gehen bei hinreichender Wettbewerbsintensität die Margen und der Gleichgewichtspreis für solche Güter ebenfalls gegen Null. Mithin schwinden dadurch die Opportunitätskosten²¹ für eine Firma, ihr Softwareprodukt als Open-Source-Software zu handhaben.²²

Die zwei genannten Faktoren – der höhere Nutzen auf der Nachfrageseite durch die Handhabung von Infrastrukturen als Allmendegüter und die vernachlässigbaren Grenzkosten, die den Gleichgewichtspreis und damit die Opportunitätskosten von Open Source gegen Null treiben – bedingen zu einem gewissen Grade einen dritten

for technical reasons), whereas commercial Linux distributions are platforms (which move forward more for commercial reasons).“

- 19 Die akademische Diskussion um die Effizienz und Nachhaltigkeit von Allmende-basierten Eigentumsregimen ist, nicht zuletzt inspiriert durch das Internet und Open Source, sehr lebhaft. Benkler (2004), der schon 2002 einen als klassisch geltenden Beitrag zur Diskussion geleistet hat, spricht in seiner empirisch geprägten Diskussion der Übertragbarkeit von *peer production* auf dinghafte Güter von einem entstehenden „dritten Modus der organisierten ökonomischen Produktion“ (S. 357). Der Beitrag von Lutterbeck in diesem Buch befasst sich mit der Schwierigkeit, die relevanten Gesichtspunkte der Thematik schlüssig zusammenzuführen. Seiner Meinung nach darf sich die Diskussion nicht nur mit Ökonomie befassen, sondern muss sich zudem die Ingenieurwissenschaften berücksichtigen.
- 20 Diese Argumentation ist natürlich sehr vereinfacht. Es gibt durchaus komplexe Schnittstellen deren Nachahmung nicht trivial ist, insbesondere wenn deren Spezifikation nicht verfügbar ist, etwa bei einem proprietären Betriebssystem wie *Microsoft Windows*. Jedoch bejaht die Rechtssprechung prinzipiell die Nachahmung von Schnittstellen zum Zwecke der Interoperabilität und Konkurrenz. So kann ein Konkurrent ein System mit ähnlicher Benutzerschnittstelle und Funktionalität auch dann anbieten, wenn die internen Schnittstellen des nachgeahmten Systems unbekannt sind. Siehe hierzu auch Samuelson und Scotchmer (2002).
- 21 Opportunitätskosten bezeichnet die Kosten, die durch den Verzicht auf eine nahe liegende Alternative entstehen – in diesem Falle: die Weiterführung der Software als proprietäre Software.
- 22 Als Beispiel sei hier die Freigabe des Quelltextes von Netscapes *Netscape Navigator* 1998 unter einer Open-Source-Lizenz genannt. Siehe hierzu auch der Artikel von Bauer und Pizka in diesem Buch.

Faktor (Atul Chitnis 2004, pers. comm.): Die Nachfrage nach Open Source steigt, weil sowohl der Gebrauchswert einer Software als auch die Konsumentenrente für den Anwender höher ist, wenn es sich um eine Open-Source-Software handelt. So erhöht sich bei Spezialsoftware das Spektrum an Optionen und damit der potenzielle Wert der Software durch die Modifizierbarkeit des Quelltextes. Bei Standardsoftware steigt ob der geringen Transaktionskosten durch Open Source die Verfügbarkeit einer Software, was wiederum eine höhere Wertschöpfung auf der Nachfrageseite bedingt.

Open Source gewinnt nicht nur an Relevanz durch nachfrageseitige Effekte, sondern auch durch produktionsseitige Effekte. Benkler (2002) widmet sich der Effizienz der Produktion von Open Source und zeigt in Elaborierung von Coases (1937) Transaktionskostenansatz, dass es ökonomisch rational sein kann, ein Produkt zusammen mit anderen unter Aufgabe von exklusiven Eigentumsrechten zu erzeugen. Er nennt diesen Ansatz *commons-based peer production* und argumentiert instruktiv, dass diese Art von Produktion insbesondere Informationskosten effektiver senkt, als dies bei den ökonomischen Produktionsmodi Markt oder Firma der Fall ist. Obgleich hier weder Preissignale, wie im Markt, noch transaktionskostensenkende Anweisungen, wie in einer Firma, zum Tragen kommen und man damit Ineffizienz unterstellen könnte (Kooths et al. 2003), machen die Kostenvorteile durch die Senkung von Informationskosten *commons-based peer production* in einigen Belangen zu einem effizienteren Modus der Produktion als Markt oder Firma.²³

Dem lässt sich hinzufügen, dass es sich schlüssig und mit einiger empirischer Fundierung argumentieren lässt, dass es in vielen Open-Source-Projekten durchaus *managerial direction* gibt, aber eben nicht durch Vorstände oder Geschäftsführer, sondern durch Komitees oder dergleichen.²⁴ Außerdem steht die empirische Evidenz für die Effizienz von preisgesteuerten Marktmechanismen im Bereich Software angesichts der unbestreitbar enormen Transaktionskosten weitgehend aus.²⁵

Während also Frischmann argumentiert, dass ein Allmendegut zu einem größeren nachfrageseitigen Nutzen führen kann und damit indirekt Nutzen und somit kommerzielle Motivation für Produzenten nach sich ziehen kann (Beratung, Schulung, Anpassung etc.), identifiziert Benkler Effizienzgewinne bei der Produktion auf der Angebotsseite als Faktoren, die eine ökonomische Sinnhaftigkeit von Allmende-basierter Produktion bedingen können.²⁶ Open Source macht also nicht nur ökonomischen Sinn für Nachfrager, sondern auch für Anbieter, indem die Wertschöpfung von der Software selbst auf angrenzende Bereiche hin verlagert wird.

23 Siehe zu einer Elaboration der Vor- und Nachteile auch Demil und Lecocq (2003).

24 Siehe wiederum der Artikel von Bauer und Pizka im Kapitel Technik und die dortigen Referenzen.

25 Insbesondere durch die relativ leichte digitale Kopierbarkeit von Software sind die Transaktionskosten aus der Durchsetzung von Eigentumsrechten immens. Es ist zumindest denkbar, dass die Kosten der Durchsetzung solcher Rechte die der Aufgabe bzw. Abgabe solcher Rechte, beispielsweise in Form von Open Source, übersteigen. Siehe zur Diskussion um die Effizienz von Marktmechanismen im Bereich Software auch die Replik von Pasche und von Engelhardt (2004) auf Kooths et al. (2003).

26 Von Hippel und andere Innovationsforscher gehen in ihrer Analyse von Open Source noch einen Schritt weiter und stellen die Dichotomie Produzent-Nutzer gleich gänzlich in Frage. Dabei verweisen sie unter anderem auf anekdotische Evidenz aus der Geschichte der Produktion von Surfbrettern. Die Reichweite dieses Ansatzes lässt sich jedoch noch nicht vollständig absehen.

5. Fazit

Open Source hat unzweifelhaft die Strukturen in der Softwareindustrie verändert. So bemerken Demil und Lecocq: „[I]ndustries are not constrained to adopt only pre-existing forms of organization“ (2003, S. 21). In ihrer Elaboration der *governance structure*, die durch Open-Source-Lizenzen geschaffen wird, kommen Demil und Lecocq zu dem Schluss, dass Firmen Open Source als strukturierendes Element ihrer Beziehungen zu anderen Firmen wählen, wenn sie sich strategischen Vorteile davon versprechen, Netzwerkeffekte durch die Schaffung von Standards zu erzeugen und für sich auszunutzen.

Während dies sicher ein Teil der Erklärung für die ökonomische Tragfähigkeit von Open Source ist, glauben wir, dass Open-Source-Software einen so fundamentalen Wandel in der Softwareindustrie bedingt, weil sie den effizienteren Modus der Produktion begründet und einen höheren absoluten Mehrwert erzeugt. Dies ist insbesondere der Fall für Infrastruktur-Güter und in ihrer Funktionalität weitgehend spezifizierte Güter.

Wenn es also einen ökonomischen wie auch technischen Imperativ für Interoperabilität von Komponenten in Softwaresystemen gibt, so hat dieser zur Folge, dass Open Source als strukturierendes Prinzip die Softwareindustrie nachhaltig beeinflusst. Dabei setzen sich nicht nur die Komponenten, die auf essenzielle Infrastrukturen zugreifen, einem wohlfahrtssteigernden Wettbewerb aus, sondern eben auch die Infrastrukturen selbst. Die nächsten Jahre sollten zeigen, ob und wie sich diese Prognose bewahrheitet.

Literaturverzeichnis

- Benkler, Y. (2002), ‘Coase’s penguin, or, Linux and The nature of the firm’, *Yale Law Journal* **112**(3), S. 369–446.
- Benkler, Y. (2004), ‘Sharing nicely: On shareable goods and the emergence of sharing as a modality of economic production’, *Yale Law Journal* **114**(2), S. 273–358.
http://www.yalelawjournal.org/pdf/114-2/Benkler_FINAL_YLJ114-2.pdf.
- Boyle, J. (2000), ‘Cruel, mean, or lavish? Economic analysis, price discrimination and digital intellectual property’, *Vanderbilt Law Review* **53**(6), S. 2007–2039.
- Coase, R. H. (1937), ‘The nature of the firm’, *Economica* **4**, S. 386–405.
- Commons, J. R. (1931), ‘Institutional economics’, *American Economic Review* **21**, S. 648–657.
- Demil, B. und Lecocq, X. (2003), ‘Neither market nor hierarchy or network: The emerging bazaar governance’, <http://opensource.mit.edu>.
- Fehr, E. und Falk, A. (2001), ‘Psychological Foundations of Incentives’. Working Paper No. 95. 2001.
- Frischmann, B. M. (2004), ‘An economic theory of infrastructure and sustainable infrastructure commons’, SSRN Electronic Library,
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=588424.

- Hann, I.-H., Roberts, J., Slaughter, S. und Fielding, R. (2004), 'An Empirical Analysis of Economic Returns to Open Source Participation', Faculty Development Grant und Carnegie Bosch Institute, Carnegie Mellon University.
- Hirshleifer, J. (1971), 'The private and social value of information and the reward to inventive activity', *American Economic Review* **61**, S. 561.
- Jevons, S. W. (1923), *Die Theorie der politischen Ökonomie*, Heinrich Wäntig, Leipzig.
- Koenig, J. (2004), 'Open Source business strategies', (überarbeitete Version)
<http://www.riseforth.com/images/Seven%20Strategies%20-%20Koenig.pdf>.
- Kooths, S., Langenfurth, M. und Kalwey, N. (2003), Open-Source-Software. Eine volkswirtschaftliche Bewertung, Economic Research Studies 4, Muenster Institute for Computational Economics, Münster. <http://mice.uni-muenster.de>.
- Lerner, J. und Tirole, J. (2000), 'The Simple Economics of Open Source', *Journal of Industrial Economics* **52**, S. 197–234. NBER Working Paper, Nr. 7600.
- Lessig, L. (1999), *Code and other laws of cyberspace*, Basic Books, New York.
- Liebowitz, S. J. und Margolis, S. E. (1999), *Winners, losers & Microsoft*, The Independent Institute, Oakland, CA.
- North, D. C. (1990), *Institutions, institutional change and economic performance*, Cambridge University Press, Cambridge.
- Osterloh, M., Rota, S. und Kuster, B. (2004), Open-Source-Softwareproduktion: Ein neues Innovationsmodell, in B. Lutterbeck und R. A. Gehring (Hrsg.), 'Open Source Jahrbuch 2004. Zwischen Softwareentwicklung und Gesellschaftsmodell', Lehmanns Media, Berlin, S. 121–138.
- Pasche, M. und von Engelhardt, S. (2004), 'Volkswirtschaftliche Aspekte der Open-Source-Softwareentwicklung', Jenaer Schriften zur Wirtschaftswissenschaft 18/2004. Wirtschaftswissenschaftliche Fakultät, Friedrich-Schiller-Universität Jena.
<http://www.wiwi.uni-jena.de/Papers/wp-sw1804.pdf>.
- Raymond, E. S. (1998), 'The cathedral and the bazaar', *First Monday* **3**(3).
<http://firstmonday.org>.
- Romer, P. M. (1993), Economic Growth, in D. R. Henderson (Hrsg.), 'The Fortune Encyclopedia of Economics', Warner Books, New York.
- Rose, C. (1986), 'The comedy of the commons: Custom, commerce, and inherently public property', *University Chicago Law Review* **53**.
- Rousseau, D. M. (1989), 'Psychological and implied contracts in organisations', *Employee Responsibilities and Rights Journal* **2**(2), S. 121–139.
- Samuelson, P. (1984), 'CONTU revisited: The case against copyright protection for computer programs in machine-readable form', *Duke Law Journal* **5**, S. 663–769.
- Samuelson, P. und Scotchmer, S. (2002), 'The law and economics of reverse engineering', *Yale Law Journal* **111**(7), S. 1575–1663.
- Weber, S. (2004), *The success of open source*, Harvard University Press, Cambridge, MA.