

Open-Source-Softwareproduktion: Ein neues Innovationsmodell?

MARGIT OSTERLOH, SANDRA ROTA UND BERNHARD KUSTER

1. Einleitung

Während langer Zeit war der Begriff Open Source nur in der Informatikbranche bekannt. Seit Kurzem interessiert sich jedoch auch die Innovationsforschung für das Phänomen. Das ist kein Zufall, handelt es sich hier doch um ein möglicherweise völlig neues Innovationsmodell, das folgende drei Fragen aufwirft (Kogut und Metiu 2001):

1. Warum sind private Eigentumsrechte an Innovationen nicht immer effizient?
2. Warum tragen Menschen freiwillig zu einem öffentlichen Gut bei?
3. Wie muss die Kooperation zwischen kommerziellen Firmen und nicht-kommerziellen Gemeinschaften ausgestaltet sein?

Die Beantwortung dieser Fragen wirft zugleich Licht auf die Bedingungen, unter denen die Produktionsweise von Open-Source-Software auch auf andere Bereiche übertragen werden kann. Schon heute gibt es Projekte, die nach ähnlichen Prinzipien funktionieren (Barbera 1999, Benkler 2002). Beispiele sind die NASA Clickworkers (ein Projekt, bei dem Freiwillige Krater auf dem Mars klassifizieren), Slashdot (ein Forum zum Kommentieren und Klassifizieren von Artikeln) und das Projekt Gutenberg (Scannen und Korrekturlesen von Büchern, deren Urheberrechte abgelaufen sind). Gemeinsames Charakteristikum dieser Projekte ist die freiwillige kollektive Innovation unter weitgehendem Verzicht auf private geistige Eigentumsrechte. Ähnlich wie in weiten Bereichen der wissenschaftlichen Produktion tauschen die Mitglieder dieser virtuellen Gemeinschaften untereinander Beiträge aus, ändern oder verbessern sie, ohne Lizenzverträge abschließen zu müssen.

Im Folgenden werden wir die eingangs gestellten Fragen am Beispiel von Open Source beleuchten. In *Abschnitt zwei* zeigen wir, inwiefern das Open-Source-Modell vom traditionellen ökonomischen Innovationsmodell abweicht. Der Hauptunterschied besteht darin, dass bei der Open-Source-Softwareproduktion unentgeltlich zu einem öffentlichen Gut beigetragen wird. In *Abschnitt drei* bilden wir eine Typologie von Akteuren in der Open-Source-Szene anhand ihrer Motivation, zum öffentlichen Gut beizutragen. In *Abschnitt vier* zeigen wir, dass die „Tragödie der Allmende“ bei der Produktion von Open-Source-Software auf zwei Ebenen überwunden werden muss. In *Abschnitt fünf* fragen wir, unter welchen Bedingungen das Open-Source-Modell generalisiert werden kann. Wir unterscheiden dazu motivationale, situative und institutionelle Faktoren. Ihr Zusammenspiel ermöglicht eine erfolgreiche Zusammenarbeit ohne die Bedingungen, die bisher als die Voraussetzung erfolgreicher Innovation galten: Privates geistiges Eigentum der einzelnen Akteure oder die zentrale Autorität eines Unternehmers. Wir kommen zum Ergebnis,

dass das neue Innovationsmodell nur dann verstanden werden kann, wenn man motivationale Faktoren, insbesondere die Dynamik von intrinsischer und extrinsischer Motivation, in die Betrachtung einbezieht.

2. Open-Source-Software: Die Überwindung der „Tragödie der Allmende“

Gemäss dem konventionellen ökonomischen Innovationsmodell entstehen Innovationen nur dann, wenn sie durch starke private Eigentumsrechte (z.B. Patente) geschützt werden (zum Beispiel North 1981). Ohne diese Eigentumsrechte haben Innovationen den Charakter eines öffentlichen Gutes. Öffentliche Güter zeichnen sich dadurch aus, dass niemand von ihrer Nutzung ausgeschlossen werden kann, auch diejenigen nicht, die zur Erstellung des Gutes nichts beigetragen haben (Snidal 1979). Deshalb sind die Anreize, in Innovationen zu investieren, mangelhaft. Es kommt zu einer „Tragödie der Allmende“ (Hardin 1968), d. h. zu einer Übernutzung oder Unterversorgung von Ressourcen der Innovation. Die Folge ist, dass das öffentliche Gut nicht erzeugt wird. Staatlich geschützte intellektuelle Eigentumsrechte sind in diesem Denkmodell unentbehrlich, da sie die Nutzungsrechte an Innovationen regeln und somit die Innovationsanreize stärken.

Es gibt aber auch kritische Stimmen, die die innovationsfördernde Wirkung von intellektuellen Eigentumsrechten anzweifeln (vgl. Bessen und Maskin 2000, Gallini und Scotchmer 2001, Heller und Eisenberg 1998, Mazzoleni und Nelson 1998). Heller und Eisenberg (1998) argumentieren, dass private intellektuelle Eigentumsrechte eine effiziente Weiterentwicklung von Innovationen geradezu verhindern können. Sie bezeichnen dieses Problem als „Tragödie der Anti-Allmende“. Eine „Tragödie der Anti-Allmende“ entsteht, wenn die Eigentumsrechte an einer Ressource auf so viele Parteien aufgesplittert sind, dass eine effiziente Bündelung der Eigentumsrechte praktisch unmöglich gemacht wird. Die Folge ist, dass diese Ressource unternutzt wird. Konkret bedeutet dies: Muss ein potentieller Innovator mit zu vielen Parteien Lizenzverhandlungen führen, können die entstehenden Transaktionskosten derart hoch werden, dass sich die Weiterentwicklung einer Technologie nicht mehr lohnt.

Empirische Evidenz für eine „Tragödie der Anti-Allmende“ in der Softwarebranche finden Bessen und Maskin (2000). Seit den 1980er Jahren wurde der Patentschutz für Software in den USA erheblich gestärkt. Im Gegensatz zu den Voraussagen des klassischen ökonomischen Innovationsmodells stiegen die Ausgaben für Forschung und Entwicklung jedoch nicht an, sondern nahmen relativ zum Umsatz sogar ab.

Die Ursprünge von Open Source liegen in den 1980er Jahren und sind in der kritischen Haltung vieler Softwareentwickler gegenüber Softwarepatenten zu suchen (Lerner und Tirole 2002a). Open Source ist ein Sammelbegriff für Softwarelizenzen, die den Softwarebenutzern nicht nur das Recht einräumen, den Quellcode zu lesen, sondern diesen auch zu verändern und diese Veränderungen zusammen mit dem originalen oder dem veränderten Quellcode wiederum Dritten zugänglich zu machen. Außerdem dürfen keinerlei Lizenzgebühren oder andere Beiträge für die Soft-

ware erhoben werden (O'Reilly & Associates 1999). Allen Open-Source-Lizenzen ist gemein, dass einmal unter eine solche Lizenz gestellter Quellcode nicht reappropriert werden kann. Das bedeutet, dass das öffentliche Gut nicht in ein privates Gut zurückverwandelt werden darf. Allerdings enthalten die verschiedenen Lizenzen unterschiedlich restriktive Bedingungen bezüglich Weiterentwicklungen des Quellcodes. So erlaubt es beispielsweise die „Birkley Software Distribution License“, Weiterentwicklungen unter eine proprietäre Lizenz zu stellen. Andere Lizenzen, wie beispielsweise die GNU General Public License, die auch unter dem Begriff „copyleft“ zusammengefasst werden, verbieten dies. Diese Programme müssen in ihrer Gesamtheit unter die gleiche Open-Source-Lizenz gestellt werden.

Open-Source-Programmierer verzichten somit freiwillig auf private Eigentumsrechte an ihren Entwicklungen. Sie bilden virtuelle „Communities of Practice“⁴, deren Mitglieder gleichzeitig Nutzer und Entwickler sind, und die selbst entscheiden, wieviel und was sie zu den Projekten beitragen wollen. Die Beiträge werden den anderen Mitgliedern zur Verwendung und Weiterentwicklung zur Verfügung gestellt. Die dadurch erreichte Gleichzeitigkeit von Design und Test der Produkte in raschen Feedback-Zyklen ist charakteristisch für Open-Source-Projekte und ermöglicht eine effiziente Entwicklung zuverlässiger Produkte (Raymond 2001). Die entstehenden Programme stellen öffentliche Güter im klassischen Sinne dar. Weshalb aber beobachten wir hier keine „Tragödie der Allmende“?

Bei der Produktion von Open-Source-Software besteht kein Problem der *Übernutzung*, da keine Rivalität im Konsum vorhanden ist. Zusätzliche Benutzer von Software können sogar positive externe Netzwerkeffekte generieren, etwa indem sie deren Bekanntheitsgrad erhöhen. Schwieriger ist zu erklären, warum das Problem der *Unterversorgung* nicht auftritt. Zu diesem Zweck fragen wir im folgenden Abschnitt, aus welchen Gründen sich Open-Source-Programmierer auch ohne private intellektuelle Eigentumsrechte in diesen Projekten engagieren.

3. Typologie von Open-Source-Programmierern

Es gibt zwei alternative Erklärungen, warum Programmierer freiwillig zu einem öffentlichen Gut beitragen:

- Sie sind eigennützige, kalkulierende Individuen, die rational in ihre Reputation investieren (Lerner und Tirole 2002a) oder die aus der Anpassung der Software an ihre eigenen Bedürfnisse einen Nutzen ziehen, der ihre Kosten übersteigt (von Hippel 2001).
- Sie haben Spaß am Programmieren oder sind überzeugt, zu einer „guten Sache“ beizutragen (z.B. Kollock 1999, Raymond 2001, Stallman 1999, Torvalds 1998).

Die beiden alternativen Erklärungen stellen auf zwei unterschiedliche Arten von Motivation ab (Deci und Ryan 2000, Frey 1997, Osterloh und Frey 2000). Die erste bezieht sich auf eine extrinsische, die zweite auf eine intrinsische Motivation der

¹ „Communities of Practice“ sind informale Gruppen, die durch geteilte Interessen und Expertise zusammengehalten werden (Brown und Duguid 1991, 1998). Virtuelle „Communities of Practice“ wurden beschrieben von Faraj und Wasko (2001) sowie Tuomi (2000).

Beitragenden. Bei der extrinsischen Motivation geht es um eine indirekte Bedürfnisbefriedigung. Die Tätigkeit wird nicht um ihrer selbst willen ausgeführt, sondern zum Beispiel, um Geld zu verdienen. Erst mit dem Geld wird dann das unmittelbare Bedürfnis erfüllt, etwa eine Urlaubsreise. Intrinsische Motivation hingegen bezieht sich auf eine direkte Bedürfnisbefriedigung. Eine Aktivität wird um ihrer selbst willen geschätzt, sie wird auch ohne Belohnung oder Bestrafung ausgeführt.²

Intrinsische Motivation hat zwei Dimensionen (Lindenberg 2001), die von verschiedenen Forschungsgruppen untersucht wurden.

- Intrinsische Motivation basierend auf der *Freude an einer Tätigkeit* wurde vor allem durch die Forschungsgruppe um Deci (Deci u.a. 1999) untersucht. Diese Art Tätigkeit verursacht ein „Flow-Erlebnis“ (Csikszentmihalyi 1975), wie es sich etwa beim Lesen eines guten Romans, beim Musizieren oder beim Bergsteigen einstellen kann. Dabei bereitet die Tätigkeit selbst einen Genuss.
- Intrinsische Motivation kann aber auch auf die *Erfüllung von Normen um ihrer selbst willen* gerichtet sein. Dies wurde vor allem durch die Forschungsgruppe um Frey (1997) untersucht. Die Relevanz dieser Form der Motivation zeigt sich zum Beispiel bei umweltgerechtem Verhalten, bei Steuerehrlichkeit oder bei kollegialem Verhalten in Unternehmen („organizational citizenship behavior“, vgl. Organ 1988).

Wir werden zeigen, dass in der Open-Source-Softwareproduktion nicht nur alle erwähnten Formen der Motivation auftreten, sondern dass nur das gleichzeitige Vorhandensein dieser verschiedenen Motivationen den durchschlagenden Erfolg einiger Projekte erklären kann. Im Folgenden unterscheiden wir fünf Idealtypen auf Grund ihrer unterschiedlichen Beitragsmotivation, die sich in der Realität allerdings durchaus überlappen können (Hars und Ou 2002, Lakhani u.a. 2002).³

3.1. Kommerzielle Unternehmen

Es existieren verschiedene Business-Modelle dafür, wie kommerzielle Unternehmen im Umfeld von Open Source einen Profit erwirtschaften können (einen Überblick geben Berlecon Research 2002 und Markus u.a. 2000). So verkauft etwa der prominenteste kommerzielle Dienstleister Red Hat nicht die Software an sich. Diese kann als öffentliches Gut gratis im Internet heruntergeladen werden. Vielmehr verkauft Red Hat Support-Dienstleistungen rund um die Software. Darüber hinaus integriert das Unternehmen verschiedene Open-Source-Komponenten zu einem vollständigen und verlässlichen Betriebssystem, welches auch von unerfahrenen Benutzern installiert werden kann. Mit diesem Business Modell konnte Red Hat etwa 50% des Marktes für Linux gewinnen (vgl. Deutsche Bank Research 2002). Andere kom-

² In der Ökonomie wird intrinsische Motivation kaum betrachtet. Zu den Ausnahmen gehören Akerlof (1982), Benabou und Tirole (2002), Frey (1997) oder Kreps (1997). Milgrom und Roberts (1992, S. 42) stellen fest, dass die Annahme ausschliesslich extrinsisch motivierter Individuen eine „extreme Karikatur“ sei. Ihrer Meinung nach müssen Institutionen dennoch so gestaltet werden, als seien alle Individuen nur eigennützig. Dies kann aber zur Verdrängung der intrinsischen Motivation führen, wie später noch ausgeführt wird.

³ Ein Idealtyp ist nach Weber (1973) ein Konstrukt, bei dem ein Bündel von Merkmalen herausgelöst wird und jene betont werden, die für einen bestimmten Bedeutungszusammenhang als relevant angesehen werden.

merzielle Firmen tragen zu Open-Source-Projekten bei, um damit den Absatz komplementärer Produkte zu fördern. So stellt zum Beispiel Hewlett Packard Druckertreiber zur Verfügung, um ihre Drucker mit Open-Source-Software kompatibel zu machen.

Diese kommerziellen Unternehmen sind unerlässlich für die Verbreitung von Open-Source-Software. Sie machen diese Software, die ursprünglich von Experten für Experten entwickelt wurde, auch für unerfahrene Nutzer handhabbar (Kogut und Metiu 2000).

3.2. Entwickler für den Eigenbedarf

Diese Benutzergruppe passt die Open-Source-Software an ihre eigenen Bedürfnisse an (von Hippel 1988). Sie geben ihre Weiterentwicklungen weiter, wenn ihr persönlicher Nutzen aus der Veröffentlichung ihre Kosten übersteigt.

Wann tritt dieser Fall ein? *Erstens* besteht durch die Veröffentlichung die Chance, dass andere Programmierer den Programmcode unterhalten, weiterentwickeln oder nach Fehlern im Quellcode suchen (z.B. von Hippel 2001, Lerner und Tirole 2002a). *Zweitens* sind die Kosten der Publikation von Beiträgen im Internet sehr niedrig. Deshalb kann sich eine Weitergabe lohnen, auch wenn die Wahrscheinlichkeit von nützlichen Rückmeldungen oder Weiterentwicklungen gering ist.

3.3. Investoren in Reputation

Programmierer können indirekt mit Beiträgen zu Open-Source-Software Geld verdienen, indem sie damit möglichen Arbeitgebern ihre Fähigkeiten signalisieren. Die daraus entstehende Reputation kann dazu führen, dass der Programmierer entweder eine besser bezahlte Anstellung erhält oder einfacher Zugang zu einem Venture-Kapitalisten bekommt. In Open-Source-Software Projekten kann man seine Fähigkeiten einfacher als bei proprietären Produkten signalisieren. Die Mailing-Listen sind öffentlich zugänglich, und oft werden Listen der wichtigsten Beitragenden publiziert (Lerner und Tirole 2002a, Moon und Sproull 2000). Zudem steht es jedem Programmierer frei, seine Beiträge zu signieren. Es ist ein ungeschriebenes Gesetz, dass diese Signaturen nicht entfernt werden dürfen (Raymond 2001). Insofern ist Open-Source-Softwareproduktion mit der wissenschaftlichen Forschung vergleichbar. Auch hier entsteht Reputation durch häufiges Zitieren, und es gehört zu den Spielregeln wissenschaftlichen Arbeitens, Quellen angemessen anzugeben.

3.4. Homo ludens

Kommerzielle Unternehmen, Entwickler für den Eigenbedarf und Investoren in Reputation gehören zur Klasse der extrinsisch Motivierten. Es gibt jedoch empirische Evidenz dafür, dass viele Programmierer intrinsisch motiviert sind. So stellt Raymond (2001) fest: „We’re proving not only that we can do better software, but that joy is an asset“⁴. Diese Aussage steht im Einklang mit Huizingas Bild vom Menschen als *Homo ludens*, dem verspielten Wesen, das Aktivitäten einfach aus Spaß an der Tätigkeit ausführt (Huizinga 1986). Viele Akteure in der Open-Source-Gemeinschaft betonen, dass ihr wichtigstes Motiv Freude am Programmieren ist, wel-

⁴ Siehe auch Brooks 1995 und Torvalds 1998.

ches ein „Flow-Erlebnis“ (Csikszentmihalyi 1975) verursacht. Über 70% der Open-Source-Entwickler sagen aus, dass sie während des Programmierens die Zeit vergessen (Lakhani u.a. 2002). In diesem Falle stellen Beiträge zum Open-Source-Code keine Kosten, sondern Nutzen dar. Sie sind nicht Investition, sondern Konsum.

3.5. Reziprokateure

Im Gegensatz zu der Austauschkultur, wie man sie üblicherweise auf Märkten findet, wird die Kultur der Open-Source-Gemeinschaft oft als Geschenkkultur charakterisiert (z.B. Raymond 2001). Ein Geschenk zeichnet sich dadurch aus, dass dem Schenkenden außer der psychologischen Befriedigung keine persönlichen Vorteile erwachsen (Rose-Ackerman 1998). Der Schenkende ist intrinsisch motiviert, die Norm der generalisierten Reziprozität um ihrer selbst willen zu verfolgen. Generalisierte Reziprozität unterscheidet sich von Austauschbeziehungen dadurch, dass es nicht um eine kalkulierende Haltung des „Wie du mir, so ich dir“ geht. Diese würde voraussetzen, dass Geber und Empfänger wechselseitig identifizierbar wären. Dies ist in der Open-Source-Gemeinschaft jedoch typischerweise nicht der Fall. Normen generalisierter Reziprozität führen dazu, dass Menschen in einer Gruppe einander Hilfe zukommen lassen, ohne dass einzelne Gruppenmitglieder identifiziert werden (Constant u.a. 1996). Eine solche Motivation drückt sich durch die folgende Aussage aus: „The person I help may never be in the position to help me, but someone else might be“ (Rheingold 1993). Teilnehmer einer Newsgroup über technische Computerprobleme berichten, dass sie der Gemeinschaft, von der sie Hilfe erhalten haben, etwas zurückgeben wollen (Faraj und Wasko 2001). Wenn sie Fragen beantworten, richtet sich ihre Hilfe nicht an den Fragesteller allein, sondern an die Gruppe als Ganzes (Wellman und Gulia 1999). Das Wohl der Gemeinschaft fließt in die Präferenzen der Einzelnen ein. Lakhani und von Hippel (2003) fanden in einer empirischen Studie über Apache Newsgroups, dass generalisierte Reziprozität („I have been helped before, so I reciprocate“, „I help now so I will be helped in the future“) sogar das am meisten genannte Motiv ist.

Wichtig ist dabei, dass Reziprokateure auf zwei Ebenen zu öffentlichen Gütern beitragen.⁵ Sie fördern ein öffentliches Gut erster Ordnung, indem sie die Funktionalität und Qualität der Software direkt verbessern. Zum öffentlichen Gut zweiter Ordnung tragen sie bei, indem sie dafür sorgen, dass der Quellcode ein öffentliches Gut bleibt. Dabei gibt es durchaus verschiedene Meinungen darüber, welche Lizenzart am besten geeignet ist, die Entwicklung und Verbreitung von Open-Source-Software voranzutreiben. Während einige die Ansicht vertreten, dass nur die GNU General Public License die Offenheit des Quellcodes garantiere, sind andere der Ansicht, der virale Effekt dieser Lizenz schränke die Freiheit der Programmierer zu sehr ein.⁶

⁵ Eine begriffliche Unterscheidung zwischen öffentlichen Gütern erster und zweiter Ordnung wird im nächsten Abschnitt vorgenommen.

⁶ „Viral“ bezieht sich auf die Eigenschaft der GPL, dass ein Programm, das unter der GPL lizenzierten Code enthält, in seiner Gesamtheit unter die GPL gestellt werden muss. Der GPL Code ist in diesem Sinne ansteckend.

4. Zwei Ebenen der „Tragödie der Allmende“: Soziale Dilemmata erster und zweiter Ordnung

Im Folgenden legen wir dar, dass die „Tragödie der Allmende“ bei der Produktion von Open-Source-Software auf zwei Ebenen überwunden werden muss.

Die von Hardin (1968) beschriebene „Tragödie der Allmende“ gehört zu den sogenannten sozialen Dilemmata. Diese entstehen, wenn Handlungen von eigennützligen Akteuren nicht zu kollektiv erwünschten Resultaten führen (Dawes 1980, Ostrom 1998). Im Gegensatz zu Hardins pessimistischer Voraussage sind allerdings längst nicht alle Menschen rein eigennützig. Empirische Untersuchungen zeigen vielmehr, dass viele Menschen freiwillig zu öffentlichen Gütern beitragen (z.B. Frey und Meier 2002), etwa weil sie Freude an der Tätigkeit haben oder weil sie einen Nutzen daraus ziehen, wenn das Gemeinschaftswohl erhöht wird. In diesem Fall wird das soziale Dilemma in ein Koordinationsspiel transformiert, bei dem es mehr als ein Gleichgewicht gibt (Sen 1974). Soziale Dilemmata können demnach gelöst werden, wenn es eine genügende Anzahl von intrinsisch motivierten Akteuren gibt.

Zu beachten ist dabei, dass soziale Dilemmata auf zwei Ebenen entstehen können. Auf der *ersten* Ebene geht es um das Trittbrettfahren bei der Produktion des Quellcodes: Weil dieser im Fall der Open-Source-Software ein öffentliches Gut ist und niemand von der Nutzung ausgeschlossen werden kann, besteht das Problem der Unterversorgung.

Auf der *zweiten* Ebene geht es um die Sanktion von Regelverletzern bei der Erstellung des öffentlichen Gutes auf der ersten Ebene. Die schlimmsten Regelverletzungen bei der Open-Source-Softwareproduktion sind die Missachtung von Lizenzbestimmungen, der Gebrauch von Open-Source-Komponenten in kommerziellen Produkten, ohne der Gemeinschaft etwas zurückzugeben, und die Entfernung der Signaturen (der „credits“) von Programmteilen.⁷ Die Sanktionierung solcher Regelverletzungen ist selbst ein öffentliches Gut: „Punishment almost invariably is costly to the punisher, while the benefits from punishment are diffusely distributed over all members. It is, in fact, a public good“ (Elster 1989, S. 41). Deshalb entsteht ein soziales Dilemma höherer Ordnung. Als klassische Lösung dieses Dilemmas gilt das Einsetzen einer zentralen Instanz, welche die Einhaltung der Regeln überwacht und dafür ein Entgelt erhält (Alchian und Demsetz 1972). Wenn es jedoch eine genügende Anzahl Freiwilliger gibt, denen es ein Anliegen ist, dass Regeln eingehalten werden, und die sogar unter Inkaufnahme persönlicher Nachteile bereit sind, Regelbrecher zu sanktionieren, kann das soziale Dilemma zweiter Ordnung ohne eine zentrale Autorität gelöst werden. Es gibt zahlreiche Laborexperimente, die zeigen, dass es solche Personen gibt (Camerer und Fehr 2003, Ledyard 1995). In der Open-Source-Gemeinschaft funktioniert Sanktionierung zum Beispiel durch „flaming“, das öffentliche Beschimpfen von Personen im Internet. Flaming ist nicht nur ein Sanktionsmittel, sondern hat zugleich eine expressive Funktion. Es demonstriert, dass eine zentrale Norm der Gemeinschaft verletzt wurde (Kollock und Smith 1996). Experimente zeigen, dass eine solche Demonstration die Bereitschaft, zu öffentli-

⁷ Vgl. zum Beispiel Raymond (2001), der feststellt: „Surreptitiously filing someone’s name off a project is, in cultural context, one of the ultimate crimes.“

chen Gütern beizutragen, signifikant erhöht (Sally 1995). Andere Sanktionsmöglichkeiten sind das „kill-filing“, d. h. die öffentliche Bekanntgabe, dass man von einer Person keine E-Mails mehr erhalten möchte, und das „shunning“, d. h. das Ignorieren von E-Mails.

Die Regelüberwachung ist in der Open-Source-Softwaregemeinschaft im Allgemeinen einfach, weil das Internet eine sehr hohe Transparenz gewährleistet.⁸ Hingegen ist die Sanktionierung von Missetätern weitaus problematischer. *Erstens* sind die genannten Sanktionen weitgehend informeller Natur. Den Regelbrechern können keine direkten monetären Kosten auferlegt werden. *Zweitens* sind die Mitglieder der Gemeinschaft teilweise anonym, und es bestehen keine klar identifizierbaren Gruppen- und Ressourcengrenzen. Damit sind die von Ostrom (1990) ermittelten Bedingungen einer Selbstregulation in Gruppen nicht gegeben. Dennoch scheint die Selbstregulation in der Open-Source-Gemeinschaft zu funktionieren. Es zeigt sich, dass Sanktionen einen signifikanten Effekt auf das Verhalten haben (Kollock und Smith 1996), obwohl diese Sanktionen oft nur Scham bei den Missetätern erzeugen können. Scham entfaltet aber nur dann eine Wirkung, wenn auch bei den Sanktionierten eine intrinsische Motivation vorhanden ist. Rein extrinsisch motivierte Egoisten fühlen keine Scham (Elster 1999, Orr 2001).

Damit sollte deutlich geworden sein, dass intrinsische Motivation eine Voraussetzung dafür ist, dass öffentliche Güter erster und zweiter Ordnung in einer Gemeinschaft von Freiwilligen entstehen können, und somit eine Bedingung für das Funktionieren des Open-Source-Innovationsmodells darstellt. Im Folgenden werden wir der Frage nachgehen, wie diese intrinsische Motivation erhalten werden kann. Dazu unterscheiden wir einen motivationalen, einen situativen und einen institutionellen Faktor.

5. Bedingungen für das Funktionieren des Open-Source-Innovationsmodells: Drei Faktoren

Open-Source-Entwickler bilden nicht die einzigen freiwilligen Gemeinschaften, bei denen das geistige Eigentum der Gemeinschaft gehört und die ohne zentrale Instanz auskommen (Benkler 2002). Dennoch zeichnet sich die große Mehrheit aller Innovationsprojekte immer noch durch das Bestehen privater geistiger Eigentumsrechte aus.

In diesem Abschnitt gehen wir näher darauf ein, unter welchen Bedingungen ein Innovationsmodell ohne geistige Eigentumsrechte funktionieren kann, ohne dass es zu einer „Tragödie der Allmende“ kommt.

Zu diesem Zweck gehen wir zunächst nochmals kurz auf die motivationalen Bedingungen für den Erfolg des Open-Source-Innovationsmodells ein.

Wir zeigen, dass intrinsische und extrinsische Motivation in Open-Source-Projekten nicht nur koexistieren, sondern unter bestimmten situativen und institutionellen Bedingungen komplementär sind. Anschließend stellen wir diese situativen und institutionellen Bedingungen dar.

⁸ Dies mit einer Ausnahme: Die illegale Integration von Open-Source-Elementen in proprietäre Software ist nur schwer nachweisbar, da bei proprietärer Software der Quellcode nicht zugänglich ist.

5.1. Der motivationale Faktor: Portfolio intrinsischer und extrinsischer Motivation

Nur durch das Engagement intrinsisch motivierter Akteure entwickeln Open-Source-Projekte genügend Zugkraft, um extrinsisch motivierte Beitragende anzuziehen (Bessen 2002, Franck und Jungwirth 2002):

- Kommerzielle Unternehmen können erst Geld mit ihren Komplementärprodukten verdienen, wenn die Software bereits eine gewisse Reife und Verbreitung erreicht hat.
- Entwickler für den Eigenbedarf haben erst dann einen die Kosten übersteigenden Nutzen, wenn die Entwicklung der Software bereits weit fortgeschritten ist und funktionsfähige Produkte vorliegen.
- Für Investoren in Reputation lohnt sich der Einstieg erst bei bereits bekannten und erfolgreichen Projekten, die genügend öffentliche Aufmerksamkeit erhalten.

Der Erfolg von Open-Source-Software ist aber auch abhängig von eigennützigem, extrinsisch motivierten Akteuren. So haben kommerzielle Unternehmen, Entwickler für den Eigenbedarf und Investoren in Reputation eine wichtige Funktion für die Verbreitung von Open-Source-Software: Sie passen die Produkte, die ursprünglich von Experten für Experten entwickelt wurden, an die Bedürfnisse einer breiten Benutzerschaft an.

Wir wissen noch wenig über das zahlenmäßige Verhältnis von intrinsisch und extrinsisch motivierten Entwicklern. Auf Grund verschiedener empirischer Untersuchungen (Lakhani und von Hippel 2003, Ghosh u.a. 2002, Hars und Ou 2002, Lakhani u.a. 2002) kann man vorläufig die Schlussfolgerung ziehen, dass extrinsisch und intrinsisch motivierte Akteure jeweils etwa die Hälfte stellen.

5.2. Der situative Faktor: Niedrige Kosten

Wenn das Open-Source-Innovationsmodell für die Entwicklung von Software so gut funktioniert, weshalb wird es dann nicht überall, beispielsweise in der pharmazeutischen Industrie, angewandt? Die einfache Antwort lautet, dass dieses Innovationsmodell nur in Situationen brauchbar ist, in denen der Nutzen der Entwickler deren Kosten übersteigt. Dies schließt auch den intrinsisch motivierten Nutzen ein. Auch unter intrinsisch motivierten Individuen sind Märtyrer und Heilige nur selten anzutreffen. Wohltäter tragen umso mehr zu einem öffentlichen Gut bei, je niedriger die Kosten sind (Rose-Ackerman 2001, S. 553). Es gilt: Je mehr Kosten moralisches Verhalten verursacht, desto seltener ist es (North 1990, S. 43). Sind die Kosten hingegen niedrig, sind viele bereit, ihren Beitrag zum öffentlichen Gut zu leisten (Kirchgässner 1992, Kliemt 1986), der in seiner Gesamtheit – wie im Fall Open Source – hoch sein kann.

Kosten wie auch Nutzen können in zwei Aspekte aufgeteilt werden, auf die wir im Folgenden detaillierter eingehen wollen: Kosten und Nutzen der Produktion von Quellcode sowie Kosten und Nutzen der Publikation dieses Quellcodes.

- Softwareentwicklung zeichnet sich durch ihren sequentiellen und komplexen Charakter aus (Bessen und Maskin 2000). Dadurch werden die

Kosten bei der Produktion des Quellcodes für den einzelnen Beitragenden gering gehalten.

- *Sequentiell* bedeutet, dass Innovation auf vorhergehenden Entwicklungen aufbaut, sodass Innovation nicht in radikalen, großen Schritten, sondern inkrementell erfolgt.⁹
- Innovation ist *komplementär*, wenn durch die Teilnahme vieler Entwickler die Chance steigt, eine Lösung für das gemeinsame Problem zu finden, weil Synergien zwischen den verschiedenen Lösungsansätzen auftreten.
- Die monetären Kosten der Publikation des Quellcodes sind im Verhältnis zum potentiellen Nutzen niedrig. Zwei Arten von Kosten müssen betrachtet werden. Zum Ersten sind die Kosten der Verbreitung gering. Beiträge zu Open-Source-Projekten können einfach auf der entsprechenden Internetseite publiziert werden. Zweitens sind die Kosten aus dem Verlust von privatem intellektuellem Eigentum oft niedrig, da viele Entwickler gar nicht die Möglichkeit hätten, ihr Produkt zu vermarkten, oder der Beitrag für eine Vermarktung zu geringfügig ist (z.B. im Falle einer reinen Fehlerbeseitigung). Der erwartete Nutzen, bestehend aus dem Feedback oder möglichen Weiterentwicklungen durch Dritte, ist somit oft größer als die direkten Kosten.

Auch wenn die Kosten für jeden einzelnen Entwickler niedrig sind, bedeutet dies nicht, dass auf intrinsische Motivation verzichtet werden kann. In Abschnitt vier haben wir gezeigt, dass eine nachhaltige Lösung der sozialen Dilemmata erster und zweiter Ordnung ohne intrinsisch motivierte Akteure nicht möglich ist. Daher ist die Frage von großer Bedeutung, wie intrinsische Motivation erhalten oder gefördert werden kann.

5.3. Der institutionelle Faktor: Intrinsische Motivation darf nicht verdrängt werden

Wir wissen noch wenig darüber, wie intrinsische Motivation entsteht. Mehr ist über die Bedingungen bekannt, unter denen eine vorhandene intrinsische Motivation verdrängt oder verstärkt wird (Frey und Osterloh 2002):

- Intrinsische Motivation ist von der empfundenen Selbstbestimmung abhängig. Deshalb wird sie durch externe Interventionen gemindert, die als kontrollierend empfunden werden. Ein Verstärkungseffekt tritt ein, wenn das Selbstbestimmungsgefühl durch Autonomie erhöht wird (Deci und Ryan 2000, für einen umfassenden Überblick über die empirische Evidenz siehe Frey und Jegen 2001).

In Open-Source-Projekten wird das Gefühl der Selbstbestimmung auf zwei Arten gestärkt. Erstens entscheiden die Programmierer selbst, bei welchem Projekt sie mitmachen und welche Beiträge sie leisten (Benkler 2002, für eine empirische Untersuchung siehe von Krogh u.a. 2003). Zweitens existiert eine Vielzahl von Partizipationsmöglichkeiten. Zahlreiche empirische Labor- und Felduntersuchungen zeigen, dass durch Partizipation die Einsatzbereitschaft

⁹ Von Krogh u.a. (2003) untersuchten anhand des Beispiels Freenet den Prozess, den ein Teilnehmer durchlaufen muss, bis er ein vollwertiges Mitglied des Projektes wird. Dieser Prozess beinhaltet langwierige Lernprozesse und zieht sich typischerweise über längere Zeit hin.

für selbstgewählte Projekte gestärkt wird (Frey u.a. 2002, Frey und Stutzer 2002, Osterloh u.a. 2002, Organ und Ryan 1995). Obwohl die Entscheidungsverfahren verschiedener Open-Source-Projekte sehr unterschiedlich sind (für eine Übersicht siehe Markus u.a. 2000), ist ihnen doch gemein, dass sie ein hohes Ausmaß an Partizipation und Autonomie gewährleisten.

- Empirische Evidenz zeigt, dass viele Individuen freiwillig zu einem öffentlichen Gut beitragen, falls auch andere einen Anteil leisten, d.h. sie sind konditional kooperationsbereit (Fischbacher u.a. 2001, Levi 1988, Ostrom 2000).

Daraus folgen zwei Konsequenzen: *Erstens* wird intrinsische Motivation verdrängt, wenn es zu viele Trittbrettfahrer gibt. Institutionelle Arrangements müssen sicherstellen, dass dieses Verhalten verhindert wird. Open-Source-Lizenzen, insbesondere „copyleft“-Lizenzen wie die GNU General Public License, sind Beispiele für solche institutionellen Arrangements (Franck und Jungwirth 2002).¹⁰ Die virale¹¹ Eigenschaft der „copyleft“-Lizenzen unterstellt Open-Source-Projekte einer besonderen Selbstbeschränkung, genannt „nondistribution constraint“ (Hansmann 1980), die typisch für Non-Profit-Organisationen ist (Franck und Jungwirth 2002). Sie gewährleistet, dass niemand die freiwilligen Beiträge oder Spenden in seinen Privatbesitz überführen kann. Auch Organisationen wie das Rote Kreuz oder viele Universitäten unterwerfen sich dieser Beschränkung. Kommerzielle Unternehmen, die im Open-Source-Umfeld tätig sind und von Beiträgen freiwilliger Mitarbeiter abhängen, müssen glaubwürdig kommunizieren, dass sie sich nicht einseitig an diesen freiwilligen Beiträgen bereichern, wollen sie die konditionale Kooperationsbereitschaft erhalten. Anderenfalls ist ihr Business-Modell gefährdet (Benkler 2002).

Dies kann am Beispiel von Red Hat gezeigt werden. Red Hat verkauft nicht den eigentlichen Linux Code. Vielmehr verkauft das Unternehmen Support und Dienstleistungen rund um Linux. Dazu werden unter anderem verschiedene Open-Source-Programme auf ihre Kompatibilität getestet und auf einer CD-ROM zusammengestellt. Kurze Zeit nach der Veröffentlichung einer solchen CD-Rom verkauften Konkurrenten Kopien dieser CD-ROM zu einem bedeutend tieferen Preis. Obwohl Red Hat nicht im Besitz des Quellcodes der Open-Source-Programme auf der CD-ROM ist, hat das Unternehmen doch Eigentumsrechte an Teilen der CD-ROM. Wie reagierte Red Hat? Die erstaunliche Antwort lautet: überhaupt nicht. Die Manager von Red Hat argumentierten, dass die Normen der Open-Source-Gemeinschaft die Durchsetzung von Eigentumsrechten nicht zulassen. Weil Red Hat vom Wohlwollen der Open-Source-Gemeinschaft abhängig ist, hat sich das Unternehmen an die Normen der Gemeinschaft gehalten, um die konditionale Kooperationsbereitschaft der freiwilligen Entwickler nicht zu zerstören.

Zweitens wird intrinsische Motivation verdrängt, wenn Kooperationsregeln nicht eingehalten werden. Wie bereits erwähnt, sind auf Grund der Transparenz des Internets die Kosten für Beobachtung und Sanktionierung von Regelbrechern in der

¹⁰ Für Ausführungen zu den verschiedenen Lizenzarten vgl. Abschnitt 2.

¹¹ Der Begriff „viral“ umschreibt plastisch die ansteckende Eigenschaft dieser Lizenzen. Jeder Quellcode, der mit unter einer solchen Lizenz stehendem Code in Berührung kommt, wird mit dem Open-Source-„Virus“ infiziert.

Open-Source-Gemeinschaft gering. Diese Sanktionen sind jedoch häufig nur dann wirksam, wenn die Regelbrecher wenigstens ein Minimum an intrinsisch motivierter Scham empfinden. Die in Open-Source-Projekten praktizierten informellen Sanktionen sind besser geeignet, die intrinsisch motivierte Kooperationsbereitschaft zu erhalten, weil sie das Gefühl der Selbstbestimmung weniger mindern als formelle Strafen (Kollock und Smith 1996, Orr 2001, Ostrom 1990).

6. Schlussbemerkungen

Open-Source-Software ist das bekannteste Beispiel eines neuen und erfolgreichen Innovationsmodells, das aber keineswegs einzigartig ist. Welche sind die Bedingungen, unter denen dieses Innovationsmodell generell funktionieren kann? Dazu haben wir drei Fragen aufgeworfen:

Die *erste* Frage lautete, warum private Eigentumsrechte an Innovation manchmal ineffizient sind. Diese Frage nach den situativen Bedingungen haben wir folgendermaßen beantwortet: Private Eigentumsrechte an Wissen sind dann ineffizient, wenn *erstens* die Innovationen sequentiell und komplementär sind, und *zweitens* die Kosten der Offenlegung im Vergleich zum erwarteten Nutzen gering sind. In diesem Fall bestehen auch ohne private intellektuelle Eigentumsrechte Innovationsanreize (vgl. z.B. Bessen und Maskin 2000). Zusätzlich droht eine „Tragödie der Anti-Allmende“: Die Inhaber privater intellektueller Eigentumsrechte können Dritte von der Nutzung einer Innovation ausschließen und so deren Weiterentwicklungen behindern. Diese Situation tritt nicht nur bei der Softwareproduktion auf, sondern auch bei anderen wissensintensiven Gütern, die auf Gruppenarbeit angewiesen sind (Benkler 2002). Ungeeignet ist dieses Innovationsmodell dann, wenn hohe Einzelinvestitionen nötig sind, wie zum Beispiel in der pharmazeutischen Industrie. In diesem Fall wären die Kosten der Offenlegung im Vergleich zum erwarteten Nutzen einer kollektiven Weiterentwicklung sehr hoch.

Die *zweite* Frage lautete, warum Individuen freiwillig zu einem öffentlichen Gut beitragen. Diese Frage bezieht sich auf den motivationalen Faktor. Wir haben gezeigt, dass intrinsisch und extrinsisch motivierte Individuen hierbei effizient interagieren. *Intrinsisch* motivierte Personen tragen zum öffentlichen Gut erster und zweiter Ordnung bei. Das öffentliche Gut erster Ordnung bezieht sich auf die Produktion von Open-Source-Software. Das öffentliche Gut zweiter Ordnung, die Sanktionierung von Regelbrechern, ist notwendig, um die konditionale Kooperationsbereitschaft zu erhalten. Sofern dies nicht nur mit Nutzen – d.h. mit Spaß an der Tätigkeit – verbunden ist, kann man davon ausgehen, dass auch intrinsisch motivierte um so eher zum öffentlichen Gut beitragen, je geringer die individuellen Kosten sind und je leichter sich die einzelnen, kleinen Beiträge zu einem effektvollen Ganzen aufsummieren lassen. Genau dies ermöglicht die Sequentialität und Komplementarität der Softwareproduktion (im Unterschied etwa zur pharmazeutischen Produktion). Deshalb sind der motivationale und der situative Faktor miteinander verbunden. *Extrinsisch* motivierte Individuen tragen aus eigennützigen Gründen zum öffentlichen Gut bei. Mit ihrem Beitrag wollen sie ihre eigenen Produktbedürfnisse befriedigen, in ihre Reputation investieren oder die Benutzerbasis

für ihre Komplementärprodukte erweitern. Wir haben gezeigt, dass der Erfolg des neuen Innovationsmodells das Zusammenspiel von intrinsischer und extrinsischer Motivation voraussetzt.

Die *dritte* Frage bezog sich auf die Kooperation zwischen kommerziellen Unternehmen und nicht-kommerziellen Gemeinschaften. Die Beantwortung dieser Frage ist verbunden mit dem institutionellen Faktor. Lizenzen wie „copyleft“ sind ein gutes institutionelles Arrangement, um die konditionale Kooperationsbereitschaft von intrinsisch motivierten Programmierern zu erhalten (Franck und Jungwirth 2002). Kommerzielle Unternehmen – wie etwa Red Hat – müssen die Normen der generalisierten Reziprozität respektieren, auch wenn sie über die rechtlichen Lizenzverpflichtungen hinausgehen, um die Kooperationsbereitschaft der Open-Source-Programmierer nicht zu untergraben. Bisher ist dies der Fall, obwohl immer mehr große kommerzielle Unternehmen (wie z.B. IBM) in der Open-Source-Gemeinde eine gewichtige Rolle spielen. Es bleibt abzuwarten, ob die für das Funktionieren des Open-Source-Innovationsmodells notwendige Balance zwischen intrinsischer und extrinsischer Motivation auch bei einem Überwiegen kommerzieller Akteure in Zukunft aufrechterhalten werden kann.

Literatur

- Akerlof, G. A. (1982): *Labor Contracts as Partial Gift Exchange*, Quart. J. of Econom. 97(4), S. 543–569
- Alchian, A. / Demsetz, H. (1972): *Production, Information Costs, and Economic Organization*, Am. Econom. Rev. 62, S. 777–795
- Barbera, F.R. (1999): *Berkman Center Rebuffed*, Harvard Law Record, 109(9),1
- Benabou, R., J. Tirole (2002): *Intrinsic and Extrinsic Motivation*, Arbeitspapier, Princeton University, Princeton, NJ
- Benkler, Y. (2002): *Coase's Penguin, or, Linux and the Nature of the Firm*, The Yale Law J. 112(3)
- Berlecon Research (2002): *Floss Final Report – Part 3: Basics of Open Source Software Markets and Business Models*,
online http://www.berlecon.de/studien/floss/FLOSS_Grundlagen.pdf
- Bessen, J. (2002): *What Good is Free Software?* In: Hahn, R. W. (Hrsg.): *Government Policy toward Open Source Software*, Washington, DC, S. 12–33
- Bessen, J., E. Maskin (2000): *Sequential Innovation, Patents and Imitation*, Arbeitspapier MIT, Cambridge, MA
- Brooks, F. P. (1995): *The Mythical Man-Month: Essays on Software Engineering*, Reading, MA
- Brown, J. S. / Duguid, P. (1991): *Organizational Learning and Communities-of-Practice: Toward a Unified View of Working, Learning, and Innovation*, Organ. Sci. 2(1), S. 40–57
- Brown, J. S. / Duguid, P. (1998): *Organizing Knowledge*, California Management Rev. 40(3), S. 90–111
- Camerer, C. F. / Fehr E. (2003): *Measuring Social Norms and Preferences using Experimental Games: A Guide for Social Scientists*, In: Henrich, J. / R. Boyd / S.

- Bowles / C. Camerer / E. Fehr / H. Gintis (Hrsg.): *Cooperation, Self Interest and Punishment: Experimental and Ethnographic Evidence from Small-scale Societies*, Oxford, NY, im Druck
- Constant, D. / Sproull, L. / Kiesler, S. (1996): *The Kindness of Strangers: The Usefulness of Electronic Weak Ties for Technical Advice*, *Organ. Sci.* 7(2), S. 119–135
- Csikszentmihalyi, M. (1975): *Beyond Boredom and Anxiety*, San Francisco, CA
- Dawes, R. M. (1980): *Social Dilemmas*, *Annual Rev. of Psych.* 31, S. 169–193
- Deci, E. L. / Ryan, R. M. (2000): *The „What“ and „Why“ of Goal Pursuits: Human Needs and the Self-Determination of Behavior*, *Psych. Inq.* 11(4), S. 227–268
- Deci, E. L. / Koestner, R. / Ryan, R. M. (1999): *Meta-Analytic Review of Experiments: Examining the Effects of Extrinsic Rewards on Intrinsic Motivation*, *Psych. Bull.* 125 (3), S. 627–668
- Deutsche Bank Research. (2002): *Free software, Big Business? Open Source Programme erobern Wirtschaft und öffentlichen Sektor*, *E-conomics*. 32
- Elster, J. (1989): *The Cement of Society: A Study of Social Order*, New York, NY
- Elster, J. (1999): *Alchemies of the Mind: Rationality and the Emotions*, New York, NY
- Faraj, S. / Wasko, M. M. (2001): *The Web of Knowledge: An Investigation of Knowledge Exchange in Networks of Practice*, Arbeitspapier Florida State University, Tallahassee, FL
- Fischbacher, U. / Gächter, S. / Fehr, E. (2001): *Are People Conditionally Cooperative? Evidence from Public Good Experiments*, *Econom. Lett.* 71(3), S. 397–404
- Franck, E. / Jungwirth C. (2002): *Reconciling Investors and Donators – The Governance Structure of Open Source*, Arbeitspapier Universität Zürich, Zürich, Schweiz
- Frey, B. S. (1997): *Not Just for the Money: An Economic Theory of Personal Motivation*, Cheltenham
- Frey, B. S., R. Jegen (2001): *Motivation Crowding Theory: A Survey of Empirical Evidence*, *J. of Econom. Surv.* 15(5), S. 589–611
- Frey, B. S., / Meier, S. (2002): *Pro-social Behavior, Reciprocity or Both?* Arbeitspapier Universität Zürich, Zürich
- Frey, B. S. / Osterloh, M. (2002): *Successful Management by Motivation. Balancing Intrinsic and Extrinsic Incentives*, Berlin
- Frey, B. S. / Stutzer, A. (2002): *Beyond Outcomes: Measuring Procedural Utility*, Arbeitspapier Universität Zürich, Zürich, Schweiz
- Frey, B.S. / Benz, M. / Stutzer, A. (2002): *Introducing Procedural Utility: Not only What, but also How Matters*, Arbeitspapier Universität Zürich, Zürich, Schweiz
- Gallini, N. / Scotchmer, S. (2001): *Intellectual Property: When is it the Best Incentive System?* In: Jaffe, A., J. Lerner S. Stern (Hrsg.): *Innovation Policy and the Economy* Bd. 2. Cambridge, MA, S. 51–78
- Ghosh, R. A. / Glott, R. / Krieger, B. / Robles, B. (2002): *FLOSS Final Report – Part 4: Survey of Developers*,
online http://www.infonomics.nl/FLOSS/report/FLOSS_Final4.pdf
- Hansmann, H. B. (1980): *The Role of Nonprofit Enterprise*, *Yale Law J.* 89, S. 835–901
- Hardin, G. (1968): *The tragedy of the commons*, *Science* 162(3859), S. 1243–1248
- Hars, A. / Ou, S. (2002): *Working for Free? Motivations for Participating in Open Source Projects*, *Internat. J. of Electron. Comm.* 6(3), S. 25–39

- Heller, M. A. (1998): *The Tragedy of the Anticommons: Property in the Transition from Marx to Markets*, Harvard Law Rev. 111(3), S. 622–688
- Heller, M. A. / Eisenberg, R. S. (1998): *Can Patents Deter Innovation? The Anticommons in Biomedical Research*, Science 280(5364), S. 698–701
- Huizinga, J. (1986): *Homo Ludens*, Boston, MA
- Kirchgässner, G. (1992): *Toward a Theory of Low-cost-decision*, Eur. J. of Polit. Econom. 8(2), S. 305–320
- Kliemt, H. (1986): *The Veil of Insignificance*, Eur. J. of Polit. Econom. 2(3), S. 333–344
- Kogut, B., A. Metiu (2000): *The Emergence of E-Innovation: Insights from Open Source Software Development*, Arbeitspapier Reginald H. Jones Center, Philadelphia, PA
- Kogut, B., A. Metiu (2001): *Open Source Software Development and Distributed Innovation*, Oxford Rev. of Econom. Policy. 17(2), S. 248–264
- Kollock, P. (1999): *The Economics of Online Cooperation: Gifts and Public Goods in Cyberspace*, in: Smith, M. A. / Kollock, P. (Hrsg.): *Communities in Cyberspace*. London, S. 220–242
- Kollock, P. / Smith, M. (1996): *Managing the Virtual Commons: Cooperation and Conflict in Computer Communities*, in: Herring, S. (Hrsg.): *Computer-mediated Communication: Linguistic, Social, and Cross-cultural Perspectives*. Amsterdam, S. 109–128
- Kreps, D. M. (1997): *Intrinsic Motivation and Extrinsic Incentives*, Amer. Econom. Rev. 87(2), S. 359–364
- Lakhani, K. / Hippel, E. von (2003): *How Open Source Software Works: „Free“ User-to-User Assistance*, Res. Policy. 32(7), S. 923–943
- Lakhani, K. / Wolf, B. / Bates, J. / DiBona, C. (2002): *The Boston Consulting Group Hacker Survey*,
online <http://www.osdn.com/bcg/BCGHACKERSURVEY.pdf> and
<http://www.osdn.com/bcg/BCGHACKERSURVEY-0.73.pdf>
- Ledyard, J. (1995): *Public Goods: A Survey of Experimental Research*, in: Roth, A. / Kagel, J. (Hrsg.): *Handbook of Experimental Economics*, Princeton, NJ, 111–194
- Lerner, J. / Tirole, J. (2002a): *Some Simple Economics of Open Source*, J. of Indust. Econom. 50(2), S. 197–234
- Lerner, J., J. Tirole. (2002b): *The Scope of Open Source Licensing*, Arbeitspapier Harvard Business School, Boston, MA
- Levi, M. (1988): *Of Rule and Revenue*, Berkeley, CA
- Lindenberg, S. (2001): *Intrinsic Motivation in a New Light*, Kyklos. 54(2/3), S. 317–343
- March, J. G. (1991): *Exploration and Exploitation in Organizational Learning*, Organ. Sci. 2(1), S. 71–87
- Markus, M. L. / Manville, B. / Agres, C. E. (2000): *What Makes a Virtual Organization Work?* Sloan Management Rev. 42(1), S. 13–26
- Mazzoleni, R. / Nelson, R. R. (1998): *The Benefits and Costs of Strong Patent Protection: A Contribution to the Current Debate*, Res. Policy. 27(3), S. 275–286
- Merton, R. K. (1983): *Auf den Schultern von Riesen : Ein Leitfaden durch das Labyrinth der Gelehrsamkeit*, Frankfurt am Main

- Milgrom, P. R. / Roberts, J. (1992): *Economics, Organization and Management*, Englewood-Cliffs, NJ
- Moon, J. Y. / Sproull, L. (2000): *Essence of Distributed Work: The Case of the Linux Kernel*. Firstmonday, 5(11)
- North, D. C. (1981): *Structure and Change in Economic History*, New York, NY
- North, D. C. (1990): *Institutions, Institutional Change, and Economic Performance*, New York, NY
- O'Reilly & Associates, Inc. (1999): *Open Source: kurz & gut*, Köln
- Organ, D. W. (1988): *Organizational Citizenship Behavior: The Good Soldier Syndrome*, Lexington, MA
- Organ, D. W. / Ryan, K. (1995): *A Meta-analytic Review of Attitudinal and Dispositional Predictors of Organizational Citizenship Behavior*, Pers. Psych. 48(4), S. 776–801
- Orr, S. W. (2001): *The Economics of Shame in Work Groups: How Mutual Monitoring Can Decrease Cooperation in Teams*, Kyklos. 54(1), S. 49–66
- Osterloh, M., B. / Frey, B. S. (2000): *Motivation, Knowledge Transfer, and Organizational Forms*, Organ. Sci. 11(5), S. 538–550
- Osterloh, M. / Frost, J. / Frey, B. S. (2002): *The Dynamics of Motivation in New Organizational Forms*, International Journal of the Economics of Business, Special Issue on New Organizational Forms. 9(1), S. 61–77
- Ostrom, E. (1990): *Governing the Commons: The Evolution of Institutions for Collective Action*, New York, NY
- Ostrom, E. (1998): *A Behavioral Approach to the Rational-Choice Theory of Collective Action*, Amer. Polit. Sci. Rev. 92(1), S. 1–22
- Ostrom, E. (2000): *Crowding out Citizenship*, Scand. Polit. Stud. 23(1), S. 3–16
- Raymond, E. S. (2001): *The Cathedral and the Bazaar*, Sebastopol, CA
- Rheingold, H. (1993): *The Virtual Community: Homesteading on the Electronic Frontier*, New York, NY
- Rose-Ackerman, S. (1996): *Altruism, Nonprofits, and Economic Theory*, J. of Econom. Lit. 34(2), S. 701–728
- Rose-Ackerman, S. (1998): *Bribes and Gifts*, in: Ben-Ner, A. / Putterman, L. (Hrsg.): *Economics, Values, and Organization*, New York, NY, S. 296–328
- Rose-Ackerman, S. (2001): *Trust, Honesty and Corruption: Reflection on the State-Building Process*, Eur. J. of Sociology. 42(3), S. 27–71
- Sally, D. (1995): *Conversation and Cooperation in Social Dilemmas: A Meta-Analysis of Experiments from 1958 to 1992*, Rationality and Society. 58(7)
- Sen, A. K. (1974): *Choice, Orderings and Morality*, in Körner, S. (Hrsg.): *Practical Reason: Papers and Discussions*, Oxford, S. 54–67
- Snidal, D. (1979): *Public Goods, Property Rights, and Political Organizations*, International Studies Quarterly. 23, S. 532–566
- Somaya, D. (2003): *Strategic Determinants of Decisions not to Settle Patent Litigation*, Strategic Management J. 24(1), S. 17–38
- Stallman, R. (1999): *The GNU Operating System and the Free Software Movement*, in: Dibona, C. / Ockman, S. / Stone, M. (Hrsg.): *Open Sources: Voices from the Open Source Revolution*, Sebastopol, CA, S. 53–70

- Torvalds, L. (1998): *FM interview with Linus Torvalds: What Motivates Free Software Developers*, Firstmonday. 3(3)
- Tuomi, I. (2000): *Internet, Innovation, and Open Source: Actors in the Network*, Firstmonday. 6(1)
- Ullman, E. (1997): *Close to the Machine: Technophilia and its Discontents*, New York, NY
- Von Hippel, E. (1988): *The Sources of Innovation*, New York, NY
- Von Hippel, E. (2001): *Innovation by User Communities: Learning from Open Source Software*, Sloan Management Rev. 42(4), S. 82–86
- Von Krogh, G. / Spaeth, S. / Lakhani, K. (2003): *Community, Joining and Specialization in Open Source Software Innovation: A Case Study*, Res. Policy. 32(7), S. 1217–1241
- Weber, M. (1973): *Die Objektivität sozialwissenschaftlicher und sozialpolitischer Erkenntnis*, in: Winckelmann, J. (Hrsg.): *Gesammelte Aufsätze zur Wissenschaftslehre*. Tübingen, S. 146–214
- Wellman, B. / Gulia, M. (1999): *Virtual Communities as Communities*, in: Smith, M. A. / Kollock, P (Hrsg.): *Communities in Cyberspace*. New York, NY, S. 167–194.