

Reziprozität in Open-Source-Software

Hagen Fritsch

<fritsch+psychologie@in.tum.de>

Seminar: Sozialpsychologie der Konsumenten

Lehrstuhl für Psychologie

Technische Universität München

24. Februar 2007

Inhaltsverzeichnis

1	Einleitung	1
1.1	Überblick	1
2	Theorie der Reziprozität	2
2.1	Direkte Reziprozität	2
2.2	Generalisierte Reziprozität	3
2.2.1	Reziprozität und Altruismus	4
3	Reziprozität und Open-Source-Software	5
3.1	Motivatoren	5
3.1.1	Lead Users	5
3.1.2	Reputation	6
3.1.3	Wissensvermehrung	7
3.1.4	Spaßfaktor	7
3.2	Reziprozität in Lizenzen	8
3.3	Reziprozität als Motivator	8
3.4	Bewertung der Rolle der Reziprozität	10
4	Schlussbemerkungen / Zusammenfassung	12

1 Einleitung

Die Frage, warum Open-Source-Software funktioniert und was Menschen motiviert an dieser mitzuwirken, wurde bereits vielfach diskutiert. In dieser Arbeit soll diese Frage unter dem Fokus, inwiefern Prinzipien der Reziprozität dabei eine Rolle spielen, aufgegriffen werden und zeigen, warum Open-Source ohne diese Prinzipien nicht funktionieren würde. Auch wenn der direkte Bezug zur Sozialpsychologie der Konsumenten nicht in übermäßig großer Ausprägung vorhanden sein mag, so schätze ich mich doch glücklich, mit diesem Thema eine weitere Brücke zwischen Psychologie und Informatik schlagen zu können. Denn genau das ist bekanntermaßen auch eine Intention dieses Nebenfachs.

1.1 Überblick

Im theoretischen Teil dieser Arbeit soll die Reziprozität vorgestellt und dabei ein besonderes Augenmerk auf die generalisierte Reziprozität gelegt werden, die die wesentlichste Form der Reziprozität im Kontext von Open-Source-Software¹ darstellt. Im folgenden Teil der Arbeit werden allgemeine Motivationsgrundlagen der Entwickler von Open-Source-Software vorgestellt, bewertet und die Reziprozität im Hinblick darauf eingeordnet.

¹Open-Source-Software ist Software, deren Quelltext offen liegt. Diese Software kann von jedem kostenlos benutzt und weiterentwickelt werden. Bekannte Beispiele sind das Betriebssystem *Linux* oder der Webbrowser *Firefox*.

2 Theorie der Reziprozität

In der Welt der Maori wird jeder Gegenstand von einem Geist begleitet, der auf diesen achtet. Auch wenn dies heute abwegig und abergläubisch erscheint, so steckt in dieser Annahme doch eine gewisse Weisheit, die hier ihre Verwendung finden wird.

2.1 Direkte Reziprozität

Bleiben wir bei dem Gedanken der Gegenstände-begleitenden Geister (in Maori: *hau* genannt). Wird der Gegenstand an jemanden verschenkt, so wird auf dem Beschenkten die Last des *hau* wirken, der einen zwingt, sich für das Geschenk zu revanchieren (Mauss, 1990). Nun ist dieser Geist in der Maori-Welt eine Metapher für ein tief in uns verwurzeltes Verständnis des gegenseitigen Miteinanders, das uns verpflichtet, uns für Gaben erkenntlich zu zeigen. Mit anderen Worten: Wir werden früher oder später ein äquivalentes Geschenk zurück geben oder uns anderweitig revanchieren müssen. Dieses Prinzip, die »Regel der Gegenseitigkeit«, wird als Reziprozität bezeichnet.

Die Tatsache, dass es überhaupt zum *Geben* kommen kann, liegt in der Reziprozität begründet, denn wir wissen um diese gesellschaftliche Norm und erwarten bei einer Gabe entsprechende Erwidern. Nach Mauss trennen wir uns nie komplett, d.h. weder physisch noch psychisch, von den Gaben, sondern behalten eine Verbindung aufrecht: die implizite Erwartung von Reziprozität.

Bei der *direkten* Reziprozität wird von der beschenkten Person erwartet, dass sie sich bei der Schenkenden revanchiert. Ein einfaches Beispiel ist wohl das Geburtstagsgeschenk, das jemanden verpflichtet dem Schenkenden zu dessen Geburtstag ebenfalls ein entsprechendes Geschenk zu machen. Diese Pflicht findet sich auch im Sprachgebrauch wieder: Nicht umsonst ist der Ausdruck „Ich bin Ihnen sehr verpflichtet“ zum Synonym für „Danke“ geworden (Cialdini, 2003). Eine wesentliche Voraussetzung für das In-Kraft-Treten der Reziprozität ist die Annahme der Gabe. Doch wer möchte schon

in jemandes Schuld stehen? Trotzdem entspricht es nicht der gesellschaftlichen Norm, Gaben – z.B. die Einladung zu einem Essen – abzulehnen. Am eindrucksvollsten illustriert das meiner Meinung nach die deutsche Sprache selbst (vgl. Mauss): Im Germanischen existiert das Wort *Gift* im Sinne einer Gabe, eines Geschenkes (heute noch in „Mitgift“ auffindbar). Die andere Bedeutung des Wortes Gift ist jedem geläufig. Somit sind Geschenke nicht immer positiv assoziiert.

Die Verpflichtung, die mit einer Gabe einhergeht, muss nicht immer gegenüber einer einzelnen Person abgegolten werden. Ebenso ist es denkbar, sich bei einer Gruppe (z.B. einer Familie, Organisation oder Partei) erkenntlich zu zeigen. In vielen Religionen und Kulturen war oder ist es immer noch üblich, sich gegenüber Gott oder der Natur (denn niemand sonst hat je mehr *gegeben*) erkenntlich zu zeigen. So beschreibt Mauss den Brauch der Tschuktischen, die die Reste des Erntedank-Festopfers ins Meer werfen oder „in alle Winde“ verstreuen, um sie der Natur zurückzugeben. (S. 42)

2.2 Generalisierte Reziprozität

Vielfach reicht die *direkte* Reziprozität nicht aus, um menschliches Verhalten in Bezug auf Gaben zu erklären. An dieser Stelle kommt die *generalisierte* Reziprozität ins Spiel. Ein viel zitiertes Beispiel ist die Beziehung zwischen Eltern und deren Kindern. So schreibt Stegbauer (2002, S. 85):

Eine ganze Reihe an Untersuchungen zeigt, dass die ältere Generation ihre Kinder und Enkel weit stärker unterstützt, als diese jemals an Gegenleistung von den Kindern und Enkeln erwarten können.

Stegbauers Erklärungsansatz der *Intergenerationen-Reziprozität* geht davon aus, dass die einzelnen Gaben aufgrund der langen Zeitspanne der Unterstützung an Bedeutung verlieren und auf die Familie generalisiert werden können. Da die Kinder die Gaben ihrer Eltern nicht erwidern können, so werden sie doch

wiederum ihre Kinder genauso unterstützen, wie ihre Eltern das für sie getan haben.

Eine andere Form der *generalisierten* Reziprozität ist die Generalisierung über ein gemeinsames Merkmal. So führt Stegbauer als Beispiel eine ethnische Minderheit an, deren Mitglieder bedürftige Mitglieder dieser Gruppe in starken Maße unterstützen, ohne die Personen zu kennen.

2.2.1 Reziprozität und Altruismus

Führt man die Generalisierung weiter, so kann diese auf die Menschheit übertragen werden. Das wäre eine Möglichkeit, Altruismus und Wohltätigkeit zu erklären. Gouldner (1984) verfolgt dabei einen einfachen Ansatz: „Wenn du möchtest, dass dir von anderen geholfen wird, musst du ihnen helfen.“ (Stegbauer 2002, S. 97 nach Gouldner (1984) S. 102). Nach Cialdini (2003) ist das gegenseitige Helfen eine sozial-evolutionäre Eigenschaft, die es einer Person ermöglicht, anderen Menschen etwas zu schenken, ohne dass das Geschenk für die Person verloren geht, weil hier das Prinzip der Reziprozität greift. So ist jeder, dem einmal freigebig geholfen wurde, in der Regel auch bereit, dies für bedürftige Personen ebense zu tun. Das erklärt auch, warum arme Menschen grundsätzlich eher bereit sind, zu spenden, als das bei reichen der Fall wäre: die Armen haben in ihrem Leben bereits mehr Unterstützung seitens der Gesellschaft oder Individuen erfahren.

Es kann und soll hier nicht abschließend geklärt werden, ob es echtes altruistisches Verhalten wirklich geben kann. Vielmehr soll darauf hingewiesen werden, dass dieses Verhalten durch verschiedene andere Ursachen motiviert wird: Sei es durch Angstmotivation (Mauss)², soziale Erwünschtheit oder Image-Pflege (Stegbauer). Auch diese Motivationen lassen sich letztendlich wieder durch Reziprozität begründen: *Indirekte* Reziprozität (Stegbauer)

²Insbesondere gibt es in verschiedenen Religion Opfergaben, um den oder die Götter zu beschwichtigen. Aus einer anderen Perspektive betrachtet, kann auch Freigebigkeit als eine solche Opfergabe gewertet werden. So schreibt Mauss: „Freigebigkeit ist obligatorisch, da sich andernfalls die Nemesis [...] an dem Übermaß an Glück und Reichtum rächt.“ (Mauss, 1990, S. 47)

wird dabei von dritten Personen abgegolten. Beispielsweise verbessern Firmen, die großzügige Spenden leisten, ihr Image, was u.U. sogar in unproportional großem Maße von den Kunden honoriert wird.

3 Reziprozität und Open-Source-Software

Die Idee der Freien Software wirkt auf den ersten Blick wie purer Altruismus, bei dem Software-Entwickler ohne Gegenleistung mitwirken, um die Gesellschaft zu verbessern, in dem der Allgemeinheit Software zur Verfügung gestellt wird, die jeder kostenfrei nutzen kann. Dass das allein so nicht stimmt, zeigt bereits die Tatsache, dass sich vielfach auch Organisationen bei der Entwicklung von Open-Source-Software (OSS) engagieren. Deren Motive sollen in dieser Arbeit allerdings nicht betrachtet werden, vielmehr liegt das Augenmerk auf den Motiven der Individuen, die freiwillig und ohne Bezahlung an OSS mitwirken.

3.1 Motivatoren

Vielfach ist der Einstieg in OSS bedingt durch einen technischen Mangel eines bestehenden Programms oder durch das Fehlen bestimmter Funktionalitäten oder gar durch die Nichtexistenz eines benötigten Programms.

3.1.1 Lead Users

Ein Benutzer, der eine bestimmte Software benötigt, ist im Umfeld von Freier Software³ geneigt, selbst eine Lösung für sein Problem zu entwickeln. Diese Projektinitiatoren werden *Lead Users* genannt.

³ Was genau der ausschlaggebende Faktor ist, ist nicht klar. Möglicherweise trägt zu den Faktoren bei, dass es im Bereich der Freien Software üblich ist, im Rahmen seiner Möglichkeiten mitzuwirken. Darüber hinaus sind Linux-Benutzer in der Regel technisch versierter und eben eher dazu in der Lage, adäquate Lösungen zu entwickeln, was sich allerdings durch die zunehmende Verbreitung von OSS auch wieder ändert.

Da eine Veröffentlichung der Errungenschaften im Internet praktisch nichts kostet, zahlt sich die Veröffentlichung aus (vgl. Osterloh u. a., 2002), da sie anderen Benutzern ermöglicht von der Lösung zu profitieren und möglicherweise die Mit- und Weiterentwicklung durch andere anregt. Außerdem erreicht man dadurch auch einen gewissen Grad an Reputation.

3.1.2 Reputation

Im Open-Source-Bereich, trägt jede Änderung die Signatur des Entwicklers. Darüber hinaus gibt es in den Quelltexten der Programme immer Angaben zu den Autoren. Nun sind viele Menschen auf der Suche nach Ruhm und Anerkennung und somit erscheint es nur logisch, dass es eines der Ziele von Entwicklern ist, Anerkennung zu erhalten. In theoretischen Abhandlungen zu diesem Thema wird davon ausgegangen, dass Reputation der Hauptmotivationsfaktor von Open-Source-Entwicklern ist. Das konnte jedoch in einer großen Umfrage⁴ (Lakhani u. Wolf, 2002) widerlegt werden. Trotzdem wird jedoch oft vermutet, dass die Reputation im Open-Source-Bereich positive Auswirkungen auf das reale Leben hat. In wenigen Fällen fanden Entwickler allein aufgrund ihrer Mitwirkung an Open-Source-Projekten Arbeit, bekamen einen Beratungsvertrag oder das Angebot eines Verlages ein Buch zu verfassen. Das ist jedoch die Ausnahme⁵, denn die Namensangaben innerhalb des Quelltextes oder der *ChangeLogs* sind nur selten über das konkrete Projekt hinaus sichtbar. Aber auch die begrenzte Reputation innerhalb einer kleinen Entwicklergruppe kann bereits als Motivator dienen. Nach Raymond (1999) stehen Entwickler in Wettbewerbssituationen. Das bedeutet, die einzige Möglichkeit den Erfolg zu messen, ist die Reputation innerhalb

⁴ Lakhani u. Wolf führten im *Hacker Survey* auf SourceForge, der größten Plattform für Open-Source-Projekte, eine webbasierte Umfrage mit 654 Open-Source-Entwicklern durch.

⁵ Zumindest die direkte Wirkung von Reputation ist die Ausnahme. Indirekt kann man sich die Teilnahme an einem Open-Source-Projekt durchaus zunutze machen. Es kann sich in Bewerbungen als vorteilhaft erweisen, belegen zu können, dass man über soziales Engagement verfügt. Mitarbeit in Open-Source-Projekten beweist das mit Sicherheit.

der Gruppe.

Auch das ist mit Vorsicht zu betrachten. Lakhani u. von Hippel (2002) führten eine Studie, die die Änderungen an einem *Branch* (Entwicklungszweig) des *Apache*-Projekts analysierte. Sie fanden heraus, dass 82% der Mitwirkenden jeweils nur ein oder zwei Änderungen eingebracht hatten. Wären die Entwickler an starker Reputation interessiert, so hätten die Einzelpersonen wesentlich mehr Änderungen eingebracht.

Desweiteren gibt es einen weltweiten Mangel an IT-Fachkräften und jeder täte wohl besser daran, direkt Geld zu verdienen, als vorerst ohne Gegenleistung in Reputation zu investieren (vgl. Osterloh u. a., 2002).

3.1.3 Wissensvermehrung

Durch das Lesen von offenen Quelltexten kann sich ein Entwickler neues Wissen aneignen. Außerdem lernt er beim Lösen von Problemen für eine spezielle Software selbst einiges hinzu. Die Änderungen, die er veröffentlicht, werden von anderen gelesen und geprüft. Dadurch werden Fehler entdeckt, die dann dem Autor mitgeteilt werden. Auch dadurch wird ein Lerneffekt erzielt. Die Beteiligung an der Entwicklung von OSS kann also als persönliche Fortbildungsmaßnahme gesehen werden.

3.1.4 Spaßfaktor

Die größte intrinsische Motivation bei Open-Source-Entwicklern geht von der Freude am Programmieren aus. Vielfach (und hier spreche ich aus Erfahrung) sind die Entwickler Tüftler, die Freude an der Herausforderung empfinden, ein Problem zu lösen. Sie beschreiben oft ein so genanntes Flow-Erlebnis, bei dem sie in der Programmiertätigkeit aufgehen. So geben 72% der Entwickler an während des Programmierens das Zeitgefühl zu verlieren (Lakhani u. Wolf, 2003). Nachdem ein Programm oder ein Teilprogramm in der Entwicklung abgeschlossen wurde, steht meist die Befriedigung etwas geschaffen zu haben das funktioniert („something that works“ Osterloh u. a., 2002). Eine Ursache für den Spaß an OSS und der damit verbundenen Produktivität und Kreati-

vität in Flow-Erlebnissen ist z.B. die Tatsache, dass der Programmierer sich das Projekt bzw. die Aufgabe selbst ausgesucht hat. Der Entwickler bearbeitet nur, was ihn interessiert, hat also im Gegensatz zum Erwerbstätigen keine stringenten Zielvorgaben oder einzuhaltende Zeitpläne.

3.2 Reziprozität in Lizenzen

OSS baut auf die freie Verfügbarkeit der Programmquelltexte. Deshalb ist es von Bedeutung, die Verfügbarkeit auch nachhaltig zu garantieren. Trotzdem soll in bestimmtem Umfang verhindert werden, dass sich Dritte an den Programmen bereichern.

Die wichtigste der Open-Source-Lizenzen ist die *GNU General Public License (GPL)* (Stallman, 1989). Diese Lizenz erlaubt explizit die Verbreitung und Nutzung der Software einschließlich des Quelltextes. Außerdem darf die Software verändert und die Veränderung auch verbreitet werden, aber nur, wenn jedes Derivat des Programms wieder unter den selben Bedingungen verbreitet wird. Die Lizenz wird daher als *reziprozitäre* Lizenz bezeichnet, da sie fordert, dass die Veröffentlichung von Änderungen wieder unter der GPL stehen muss. Das garantiert einerseits die Freiheit der Software, wird aber andererseits auch kritisiert, da sie die Freiheit anderer einschränkt: Die Lizenz hat einen „viralen Effekt“ (Osterloh u. a., 2002, S. 14), weil jedes andere (gerade auch kommerzielle) Programm, das die Software oder Teile von ihr benutzen will, gezwungen ist, nachfolgend das gesamte Programm offen zu legen. Das ist für die meisten software-produzierenden Firmen (wie z.B. Microsoft) nicht erfüllbar, da der Quelltext für diese Firmen den Status eines gut zu hütenden Firmengeheimnisses einnimmt.

3.3 Reziprozität als Motivator

Diese Zwangsmaßnahme der GPL ist wesentlicher Bestandteil der OSS und in vielen Bereichen die Ursache, dass der Quelltext offen gelegt wird. Im Folgenden soll es aber nicht um die rechtliche Verpflichtung der Einhaltung der

Lizenz, sondern um die persönliche Motivation Einzelner aus Gründen der Reziprozität gegenüber der Open-Source-Community gehen. Die Argumentationsidee ist dabei einfach: Freie Software ist frei, wird also ohne erwartete Gegenleistung zur Verfügung gestellt und jeder kann diese Software nutzen. Durch kostenlose Software spart man gegenüber kommerziellen Produkten Geld,⁶ bekommt also im Endeffekt ein Geschenk, und das sollte, wie im ersten Teil dieser Arbeit bereits dargelegt wurde, die Reziprozität hervorrufen.

Das ist auch tatsächlich der Fall. So bekam ich für meine Arbeit an `wput` (Fritsch, 2005) regelmäßig Dankes-E-Mails. Gerade während ich an diesem Absatz schrieb, erhielt ich eine E-Mail aus Israel:

Thanks for the program. It's made my life a lot easier.

Im Wesentlichen mochten die Benutzer mein Programm, weil es einen Bereich abdeckte, für den damals noch keine Software zur Verfügung stand und da ihnen das Programm Arbeit abnahm, wollten sie sich wenigstens durch ein „Danke“ erkenntlich zeigen, was eben zitierte E-Mail perfekt illustriert. An dieser Stelle ist aber festzuhalten, dass darüber hinaus gehende Angebote der Mithilfe am Programm sehr selten waren.⁷

Viele Open-Source-Projekte rufen zu Spenden auf, um Entwickler, Hardware oder Werbung bezahlen zu können, was gerade bei größeren Projekten wie z.B. dem Webbrowser *Firefox* oder auch der *Wikipedia*⁸ sehr erfolgreich

⁶Statt Microsofts Betriebssystem *Windows* zu kaufen, kann man z.B. das kostenlose *Linux* benutzen, selbiges gilt für *OpenOffice* anstelle von *Microsoft Office*.

⁷ Konkret gab es eine Person, die sich als Maintainer um die Verwaltung von `wput` im Debian Projekt gekümmert hat, eine weitere Person die kleine Entwicklungsbereiche übernommen hat und eine Hand voll Menschen, die bei der Übersetzung geholfen haben.

⁸ Die Wikipedia ist an sich kein Open-Source-Projekt, kommt dem jedoch sehr nahe, da die Nutzer auch dort ohne Gegenleistung ihr Wissen einbringen. Auch die Wikipedia ist ein gutes Beispiel, um die Reziprozität zu illustrieren, da dort die anfängliche Investitionsschwelle viel niedriger ist und insgesamt viel mehr Menschen überhaupt die Möglichkeit haben, sich an dem Projekt zu beteiligen. Bei OSS ist der Kreis der Mitwirkenden in der Regel nur auf Programmierer beschränkt.

ist. All dies sind Formen *direkter* Reziprozität. Der Benutzer sagt den Entwicklern „Danke“ oder „bezahlt“ für die Software in Form einer Spende. Eine direkte weitere Beteiligung am Projekt erfolgt nur in seltenen Fällen.

Bei der *generalisierten* Reziprozität stellt sich das Problem, dass sie schwer messbar ist. Bei informellen Umfragen in meinem Freundeskreis fand ich heraus, dass die Open-Source-Gemeinschaft als etwas Großartiges erleben. Viele fühlen sich auch verpflichtet etwas zurückzugeben und würden, wenn sie eine gute Idee hätten, von der sie denken, dass sie der Open-Source-Gemeinschaft nützt, auch eigene Quelltexte als Open-Source zur Verfügung stellen. Einige haben dies bereits getan.

Ich selbst schrieb in meiner Arbeit über `wput` damals (2005), dass einer meiner Motivationsfaktoren an `wput` zu arbeiten, das Gefühl war, der Open-Source-Gemeinschaft etwas schuldig zu sein. Und in der Tat findet sich dieses Motiv sehr häufig auch bei anderen Entwicklern, wird jedoch selten explizit genannt. Mark Shuttleworth der Gründer des *Ubuntu* Projekts (eine der derzeit führenden Linux-Distributionen), wird in einem Buch über *Ubuntu* wie folgt zitiert und bestätigt damit dieses Motiv:

Meine Investition in Ubuntu (zumindest das Geld, das wir für Open-Source-Entwicklung und Tools wie Launchpad für Open-Source-Entwickler, ausgeben) kann ich persönlich philanthropisch begründen, weil ein Großteil meines Glücks und meines Wohlstands nur durch die Verwendung von Open-Source-Tools entstanden ist. Ich schätze mich glücklich, einen Teil davon der Community zurückgeben zu können. (Fischer, 2007, Anhang C nach Shuttleworth)

3.4 Bewertung der Rolle der Reziprozität

Die Reziprozität ist ein Motivationsfaktor für Open-Source-Entwickler, jedoch stellt sich die Frage, wie stark dessen Einfluss ist. Harte Zahlen sind schwer zu bekommen, doch glücklicherweise finden sich einige im bereits erwähnten *Hacker Survey* von Lakhani u. Wolf (2002). Von 684 befragten

Intellectually stimulating	45%
Improves skill	41%
Work functionality	34%
Code should be open	33%
Non-work functionality	30%
Obligation from use	29%
Work with team	20%
Professional status	17%
Other	16%
Open Source reputation	11%
Beat proprietary software	11%
License forces me to	0.2%

Tabelle 1: Übersicht der Motivationen nach Lakhani u. Wolf (2002)

Open-Source-Entwicklern gaben 29% an, dass die „*obligation from use*“ – also das durch die Benutzung der Software entstandene Schuldgefühl – ein Motivationsfaktor für sie sei. Die Reziprozität steht damit auf Platz sechs nach u.a. intellektueller Stimulation (45%) oder Lernerfolg (41%), aber noch gefolgt von Faktoren wie Reputation (11%) oder Lizenzzwang (0.2%).

Ein weiterer nicht zu vernachlässigender Faktor ist die Rolle des Altruismus, der im *Hacker Survey* leider nur unzureichend erfasst ist. Darin wird lediglich nach ideologischen Motivationen gefragt, wonach 33% der Befragten angeben, dass ein Motivationsgrund der Glaube ist, dass Software quelloffen sein sollte. Dennoch zeigt sich hier deutlich, dass sich die Befragten eigentlich aus altruistischen Motiven beteiligen, eben um z.B. die Gesellschaft durch OSS voranzubringen. Außerdem vertreten sie jedoch eine Einstellung, die, wenn sich deren Gedanken verbreiten, ihnen auch selbst wieder Vorteile bringt, nämlich die Mitwirkung anderer an OSS. Im Wesentlichen handelt es sich um ein klassisches Beispiel für die *generalisierte* Reziprozität.

Auch andere Beispiele belegen das Altruismus-Motiv: Dem *Survey* nach

geben 83% der Befragten an, dass *Hacker* die primäre Community sei, mit der sie sich identifizieren. Und gerade diese Gemeinschaft zeichnet sich durch starke ideologische Grundsätze aus, die letztendlich den Altruismus belegen. Davon ausgehend, dass echtes altruistisches Verhalten nicht existiert (siehe Kapitel 2.2.1), ist ein Beleg für Altruismus automatisch auch ein Beleg für *generalisierte* Reziprozität.

In einer weiteren Umfrage⁹ (Aalbers, 2004), wird die Reziprozität im Durchschnitt mit 3.5 Punkten (auf einer Skala von 1 bis 5, wobei 5 der Höchstwert wäre) und Altruismus mit 3.7 Punkten bewertet. Beide Motive werden also sehr hoch eingeschätzt.

Damit ist klar, dass die Reziprozität für viele Entwickler ein wichtiger Motivationsfaktor ist, jedoch sollte die Wichtigkeit auch nicht überschätzt werden, denn die wirklich ausschlaggebenden Faktoren für die Mitarbeit an Open-Source-Projekten sind andere, wie vorab bereits beschrieben. Viele meiner Freunde verspüren diese Verpflichtung gegenüber der Open-Source-Gemeinschaft, ihnen fehlt es aber an Wissen, Ideen oder schlicht und einfach an Zeit, um selbst auch aktiv mitwirken zu können.

4 Schlussbemerkungen / Zusammenfassung

Reziprozität ist eines der Grundkonzepte unseres sozialen Umgangs. *Generalisierte* Reziprozität ermöglicht es uns zu geben, ohne direkt davon zu profitieren und verpflichtet andere sich an anderer Stelle zu revanchieren. Beide Konzepte spielen als Motivationsfaktoren für Open-Source-Entwickler eine wichtige Rolle, aber auch die sollte nicht überschätzt werden.

Die hier genannten harten Zahlen sind trotz Professionalität und Repräsentativität der Umfrage mit Vorsicht zu betrachten. Wie bereits erwähnt, ist Freigebigkeit eine sozial erwünschte Eigenschaft. Menschen neigen dazu

⁹Bei dieser Umfrage ist zu beachten, dass die zu einer bestimmten Software gehörende Community befragt wurde. Insgesamt wurden 746 Personen befragt. Jedoch sind nur 21% von ihnen direkt an der Software-Entwicklung beteiligt. Andere Gruppen erstellen Grafiken oder Dokumentation zur Software und wieder andere helfen Benutzern im Support-Forum.

sich selbst entsprechend positiv darzustellen. Deshalb ist anzunehmen, dass bei der Beantwortung der Frage nach der Motivation mehr altruistische Motive angegeben werden, als dies tatsächlich der Fall ist.

Trotzdem beteiligen sich immer mehr Menschen an der Entwicklung von OSS und diese Entwicklung ist – altruistische Motive hin oder her – ein nachhaltiger Gewinn für die gesamte Gesellschaft, der gerade im Zuge der immer weiteren Kommerzialisierung des alltäglichen Lebens ein klaren Akzent gegen diese Entwicklung setzt.

Literatur

Aalbers 2004

AALBERS, Martine: *Blender Questionnaire: Motivation for participating in an online open source software community*. http://download.blender.org/documentation/bc2004/Martine_Aalbers/results-summary.pdf. Version: 2004

Cialdini 2003

CIALDINI, Robert B.: *Die Psychologie des Überzeugens*. Huber, Bern, 2003

Fischer 2007

FISCHER, Marcus: *Ubuntu GNU/Linux*. Galileo Computing, 2007

Fritsch 2005

FRITSCH, Hagen: *Wput. Ein vollautomatischer Konsolen-FTP-Client*, Ernst-Friedrich-Oberschule, Berlin, Besondere Lernleistung, 2005. <http://wput.rootboard.org/wput/wput.pdf.gz>

Gouldner 1984

GOULDNER, Alwin W.: *Reziprozität und Autonomie*. suhrkamp, 1984

Lakhani u. von Hippel 2002

LAKHANI, Karim ; HIPPEL, Eric von: *How open source softwa-*

re works: "freeüser-to-user assistance. <http://opensource.mit.edu/papers/lakhanivonhippelusersupport.pdf>. Version: 2002

Lakhani u. Wolf 2002

LAKHANI, Karim ; WOLF, Bob: *The Boston Consulting Group/OSTG Hacker Survey*. <http://www.ostg.com/bcg/>. Version: 2002

Lakhani u. Wolf 2003

LAKHANI, Karim ; WOLF, Bob: *Why Hackers Do What They Do: Understanding Motivation Effort in Free/Open Source Software Projects*. <http://opensource.mit.edu/papers/lakhaniwolf.pdf>. Version: 2003

Mauss 1990

MAUSS, Marcel: *Die Gabe. Form und Funktion des Austauschs in archaischen Gesellschaften*. suhrkamp, 1990. – (Original erschienen 1923-24: *Essai sur le don. Forme et raison de l'échange dans les sociétés archaïques*)

Osterloh u. a. 2002

OSTERLOH, Margit ; ROTA, Sandra ; KUSTER, Bernhard: *Open Source Software Production: Climbing on the Shoulders of Giants*. 2002

Raymond 1999

RAYMOND, Eric S.: Homesteading the Noosphere. In: *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly & Associates, 1999, S. 79–135. – Original erschienen online 1998

Stallman 1989

STALLMAN, Richard: *GNU General Public License*. <http://www.fsf.org/licensing/licenses/gpl.html>. Version: 1989

Stegbauer 2002

STEGBAUER, Christian: *Reziprozität. Einführung in soziale Formen der Gegenseitigkeit*. VS Verlag für Sozialwissenschaften, 2002