

Dieser Artikel ist Teil des
Open Source Jahrbuchs 2006

Bernd Lutterbeck
Matthias Bärwolff
Robert A. Gehring (Hrsg.)

Open Source
Jahrbuch 2006

Zwischen Softwareentwicklung und Gesellschaftmodell

erhältlich unter www.opensourcejahrbuch.de.

Die komplette Ausgabe enthält viele weitere interessante Artikel. Lob und Kritik zu diesem Artikel sowie weitere Anregungen können Sie uns einfach und unkompliziert mitteilen per E-Mail oder auf www.opensourcejahrbuch.de/feedback/.

Ist ein Open-Source-Projekt nur kooperativ? – Die Koordination der Zusammenarbeit im KDE-Projekt

ANDREAS BRAND UND ALFONS SCHMID



(CC-Lizenz siehe Seite 499)

Die Koordination der freiwilligen Zusammenarbeit in Open-Source-Projekten als virtuellen Organisationen wird bisher überwiegend als kooperativ gesehen. In diesem Beitrag wird am Beispiel des KDE-Projekts untersucht, welchen Stellenwert die Kooperation, aber auch Hierarchie und Wettbewerb bei der projektinternen Allokation der Arbeit in einem Open-Source-Projekt haben. Dabei unterteilen wir die projektinterne Allokation der Arbeit in Arbeitsaufteilung, Arbeitsleistungserstellung und Arbeitszusammenführung. Als wesentliches Ergebnis stellt sich heraus, dass zwar die Kooperation dominiert, Hierarchie aber ebenfalls eine hohe Bedeutung bei der Koordination hat. Wettbewerb ist vorhanden, spielt aber eine untergeordnete Rolle. Kooperation ist eher bei der Arbeitsverteilung und der Arbeitsleistungserstellung zentral, Hierarchie ein wichtiges Steuerungsmedium bei der Arbeitszusammenführung. Wettbewerb besteht in Teilbereichen, wie z. B. bei der Erlangung von Reputation zur Besetzung von Positionen.

Schlüsselwörter: Open-Source-Projekt · virtuelle Zusammenarbeit · Hierarchie
· Koordination · Kooperation · Wettbewerb

1 Einleitung – Der Begriff „Open-Source-Projekt“

Open-Source-Projekte¹ haben als innovative Art der Softwareerstellung seit den neunziger Jahren eine starke Verbreitung gefunden. So gilt Linux inzwischen als ernstzunehmender Konkurrent von Microsoft Windows. Damit hat aber nicht nur das

¹ Es wird hier die Verwendung von Open Source als Kennzeichnung aller Free-Software-/Open-Source-Projekte verwendet, da sich diese Umschreibung als Oberbegriff für die zwei Richtungen „Open Source“ und „Free Software“ durchgesetzt hat.

öffentliche, sondern auch das wissenschaftliche Interesse an Open-Source-Projekten zugenommen.

Open-Source-Projekte sind virtuell-textbasierte, vorwiegend informale Gruppen, bei denen mehrere freiwillige Akteure zusammen über das Internet Software entwickeln. Sie unterscheiden sich von anderen virtuellen Gruppen mit Geselligkeitszielen und Kommunikation über ein einziges Medium wie Foren, Usenet oder Mailinglisten durch ihren Arbeitsbezug und ihre Nutzung verschiedener Kommunikationsmedien (vgl. Götzenbrucker und Löger 2000; Lerner und Tirole 2000).

Die Koordination der Arbeit in Open-Source-Projekten wurde aber bisher theoretisch und empirisch wenig thematisiert und auch nicht systematisch untersucht (vgl. z. B. Jørgensen 2001; Mockus et al. 2002). Zwar wird in der sozialen Open-Source-Bewegung Kooperation als vorherrschende Koordinationsform angeführt (vgl. Raymond 1998, 1999), es finden sich aber auch hierarchische und wettbewerbliche Elemente in Open-Source-Projekten (vgl. Rossi 2004; Ghosh 1998; Lerner und Tirole 2000). Zumeist werden dabei altruistische Motive der Beteiligten betont, aber auch die Idee eines wettbewerblichen Tauschmarkts von Ideen gegen Reputation hat sich in der Debatte weitgehend etabliert. Zusätzlich stellt sich Hierarchie als Rangordnung von „Leadern“ dar, die verschiedene Verantwortlichkeiten in Projekten übernehmen.

Hier knüpft der vorliegende Beitrag an, wobei wir einen neuen Weg einschlagen. Im Rahmen einer Fallstudie über das Open-Source-Projekt „K Desktop Environment“ (KDE) wird empirisch für die projektinterne Allokation der Softwareerstellung eruiert, inwieweit neben der Kooperation auch Hierarchie und Wettbewerb als Koordinationsformen vorhanden sind, welche Bedeutung Hierarchie und Wettbewerb neben der Kooperation haben und welche Kombination der drei Koordinationsformen besteht (vgl. Brand und Schmid 2004). Wir untersuchen hier die Koordination der projektinternen Allokation der Arbeit. Dabei unterteilen wir die Betrachtung in drei Unterkategorien:

- Projektinterne Zuordnung von Arbeitsaufgaben und Arbeitskräften
- Transformation von Arbeitsvermögen in Arbeitsleistung (Leistungserstellung)
- Zusammenführung der Arbeitsleistungen zu einem stabilen Release

Die projektexterne Allokation, also den Ein- und Austritt in das bzw. aus dem Projekt, untersuchen wir nicht. Zunächst skizzieren wir im nachfolgenden Abschnitt kurz die konzeptionelle Grundlage für unsere empirische Untersuchung mit der Unterscheidung der Koordinationsformen Wettbewerb, Hierarchie, Kooperation. Im Abschnitt 3 referieren wir die Ergebnisse aus der Fallstudie bezüglich der Bedeutung der drei Koordinationsformen bei der projektinternen Allokation in dem untersuchten Open-Source-Projekt. Der Abschnitt 4 schließt den Beitrag mit einem Resümee.

2 Kooperation, Hierarchie und Wettbewerb als Koordinationsformen

Ausgangspunkt unserer Überlegungen ist, dass die Arbeit in einem Softwareprojekt entweder durch Kooperation, Hierarchie oder Wettbewerb koordiniert wird. Diese Koordinationsformen werden als gleichwertig betrachtet und sind von der Organisationsform Open-Source-Projekt zu trennen. Dabei entfernen wir uns von den gängigen Vorstellungen in der ökonomischen und soziologischen Literatur.² Eine weitgehend akzeptierte Begriffsdefinition von Kooperation, Hierarchie und Wettbewerb liegt u. W. bisher nicht vor, weswegen wir diese jeweils im Folgenden definieren.

2.1 Kooperation

Koordinationsmechanismus bei der Kooperation ist die Verhandlung. Das Verhandlungsergebnis besteht in einem Konsens bez. eines gemeinsamen Ziels. Das angestrebte Ziel ist eine Win-Win-Situation bzw. Besserstellung mindestens einer Person. Eine opportunistische Schädigung ist zwar möglich, wenn sie festgestellt wird, wird die Kooperation jedoch abgebrochen.

Drei Kooperationsarten bestimmen die Zielerreichung, wobei sie sich in der Dimension Entfernung im sozialen Raum, d. h. Äquivalenz der Gaben, Zeit, Raum und Person, unterscheiden:³ strategische Kooperation, reziproke Kooperation und gruppenbezogene Kooperation.

Erfolgt die strategische Kooperation z. B. von Preisverhandlungen zeitpunkt- und raumbezogen, d. h. einmalig und direkt zwischen zwei Akteuren z. B. auf zwei Marktseiten (vgl. Mahnkopf 1994, S. 71 f.), so wird bei der reziproken Kooperation die genaue Gleichwertigkeit sowie Zeit und Raum des Tauschs aufgelöst (vgl. Stegbauer 2002, S. 46, 54 ff.; Bourdieu 1998; Blau 1964). Die gruppenbezogene Kooperation bezieht sich auf eine generalisierte Reziprozität mit „impliziten“ Verhandlungen bzw. Kommunikation von z. B. Hilfsbedürftigkeit innerhalb der Gruppe (wie Familie oder Sportverein) zwischen gleichberechtigten Gruppenmitgliedern, wobei zusätzlich von der Person abstrahiert wird, d. h. eine direkte Zuordnung der Gaben ist nicht möglich (vgl. Stegbauer 2002, S. 79).⁴

2 Für eine genauere Darstellung der Diskussion zu Koordinationsformen und Organisationsformen siehe Brand und Schmid (2005) bzw. Kawai und Schmid (2004, S. 250 ff.).

3 Vgl. hierzu Noteboom (2002, S. 63 ff.) und Blau (1964).

4 Altruistische Kooperation als spezieller Fall der gruppenbezogenen Kooperation basiert auf einem diffusen Zugehörigkeitsgefühl zu sehr großen Gruppen mit besonders großer sozialer Distanz, der Kooperationsnorm und Emotionen wie Mitleid bzw. Empathie. Es wird keine Gegenleistung für z. B. anonyme Kleinspenden erwartet, weswegen es weder eine Äquivalenz noch einen zeitlichen und räumlichen Bezug zu einer Gegengabe gibt (vgl. Stegbauer 2002, S. 98 ff.).

2.2 Hierarchie

Das übliche Verständnis von Hierarchie ist besonders auf Unternehmen bzw. bürokratische Organisationen über abgeleitete Verfügungsrechte über Produktionsmittel ausgerichtet und lässt sich nicht ohne weiteres auf ein Open-Source-Projekt übertragen, da nach der GPL-Lizenz⁵ kein Projektbeteiligter als Eigentümer angesehen werden kann. Wir subsumieren daher Hierarchie unter die soziale Grundkategorie der Macht, die über den Zugriff auf Ressourcen die Dynamik in einem Open-Source-Projekt adäquat beschreibt.

Voraussetzung für Hierarchie ist die Abhängigkeit von den persönlichen oder organisatorischen Ressourcen einer oder mehrerer Personen (vgl. Hradil 1980). Der Mechanismus der Hierarchie bzw. Macht bei der Entscheidungsfindung besteht nach Max Weber innerhalb einer sozialen Beziehung darin, den eigenen Willen auch gegen das Widerstreben einer oder mehrerer Personen durchzusetzen (Weber 1956, S. 157 ff.).⁶

Die Zielerreichung wird durch mehrere Machtarten entweder bestrafender, belohnender oder manipulierender Art bestimmt, wobei Ressourcen gewährt oder zurückgehalten/entzogen werden können. Dabei sind mit positiven Sanktionen Belohnungen wie monetäre Anreize und mit negativen Sanktionen Bestrafungen verbunden. Die manipulierende Macht dagegen steht mit der Beeinflussung der Kommunikation in Verbindung, worunter Drohungen, aber auch die argumentative Beeinflussung der (öffentlichen) Meinung verstanden werden können. Die manipulative Macht setzt sich dabei aus Expertenmacht (formales Bildungskapital oder informales symbolisches Kapital bzw. Reputationsmacht), Informationsmacht („Gatekeeper“-Funktion in einem Kommunikationsnetzwerk bzw. soziales Kapital), Positionsmacht (legitimierte Posten bzw. „objektiv“ zurechenbare soziale Merkmale⁶) und Identifikationsmacht (besonderes Charisma einer Persönlichkeit oder Identifikation mit einer Gruppe) zusammen (vgl. Raven 1992; Bourdieu 1987; Bourdieu 1998, S. 108; Hradil 1980).

2.3 Wettbewerb

Wettbewerb wird oft mit Marktwettbewerb gleichgesetzt. Wir verstehen Wettbewerb umfassender. Wettbewerb ist ein friedlicher Kampf um ein Ziel, Gut oder eine Ressource, das/die nur von einer Person oder Personengruppe vereinnahmt wird bzw. werden kann. Eine Person oder Personengruppe bekommt alles, die andere(n) nichts

5 Die *General Public Licence (GPL)* schützt den offenen Zugang zum Quellcode. Diese Lizenz legt durch eine geschickte Definition der Verfügungsrechte fest, dass ein veränderter Quellcode bei der Weitergabe wieder dem Projekt zugeführt wird. Damit soll eine proprietäre Aneignung des freiwillig geschriebenen Quellcodes mit nachfolgendem Einfordern von Lizenzgebühren ausgeschlossen werden (vgl. Jaeger und Metzger 2002).

6 Eng verwandt mit der Macht ist die Herrschaft (Weber 1956), die als institutionalisierte Macht breite Zustimmung erfährt und als legitim angesehen wird. Die Herrschaft spiegelt sich in der Legitimität der institutionalisierten Positionen wider, wird hier aber nicht untersucht.

oder das, was ihrer Leistung entspricht. Wettbewerb erfolgt durch einen Vergleich von Qualitäten oder Quantitäten, von Ziel und Zielerreichung. Die Zielerreichung kann mit oder ohne Zutun der zu Vergleichenden erfolgen. Entweder kann ein Ranking über die Bewerteten erstellt werden oder diese müssen an einem Bietprozess mit verschiedenen Versteigerungsarten teilnehmen (vgl. Arentzen und Lörcher 1997; Brand und Schmid 2005).

3 Projektinterne Allokation von Arbeiten im KDE-Projekt

Das untersuchte Open-Source-Projekt KDE⁷ entwickelt einen Desktop für Linux-Betriebssysteme. Es besteht aus einer weltweit verteilten Gruppe von ca. 1000–1200 Menschen, die ihren räumlichen Schwerpunkt in Europa, besonders in Deutschland haben. Diese Gruppe ist demografisch homogen, der überwiegende Teil der Projektmitarbeiter ist männlich, zwischen 20 und 30 Jahre alt und hat oder absolviert eine IT-Ausbildung. Neben dem in diesem Artikel untersuchten, dominierenden Bereich der Softwareentwicklung gibt es noch andere nachrangige Tätigkeitsbereiche, in denen z. B. technische Dokumentare/Übersetzer arbeiten. Das Open-Source-Projekt besteht einerseits aus wenigen wichtigen und zentralen Unterprojekten mit hoher Mitarbeiterzahl zwischen 20 und 50 Personen, andererseits aus vielen kleinen Ein-Personen-Unterprojekten. Die Softwareerstellung findet aber normalerweise in kleinen und kleinsten Gruppen von im Durchschnitt vier Personen statt. Die wichtigsten Werkzeuge stellen die Mailinglisten für die Kommunikation und das Dateiablage-/Versionsmanagementsystem, in dem der Quellcode lagert, für die Produktion dar (vgl. Brand und Schmid 2004).

Das Vorgehen bei der Untersuchung des KDE-Projekts stützte sich auf verschiedene Methoden. Nicht nur die Homepage an sich, sondern auch dortige Selbstauskünfte von 62 Personen und 55 beantwortete Fragebögen von Open-Source-Projektmitgliedern wurden dabei zum Einstieg analysiert. Auch die Mitglieder kamen in elf halbstandardisierten Interviews zu Wort, wobei weitere informelle Gespräche auf Konferenzen im Zuge teilnehmender Beobachtungen erfolgten (vgl. Brand und Schmid 2004, 2005; Becher et al. 2005).

Wir haben uns bei der Untersuchung auf die Koordination bei der projektinternen Allokation in dem Projekt konzentriert. Darunter wird hier erstens die Arbeitsverteilung, d. h. die Zuordnung der Arbeitsaufgaben auf die jeweilige Arbeitskraft verstanden. Es werden aber nicht nur Arbeitsaufgaben/-bereiche an die Projektbeteiligten, sondern auch Posten verteilt. Zweitens subsumieren wir darunter die Leistungserstellung, d. h. die konkrete Transformation der Arbeitskraft in Arbeitsleistung, also

⁷ Wir wählten das KDE-Projekt wegen seiner durch Vorrecherchen bekannten ausgeprägten kooperativen Ausrichtung und des einfachen Zugangs zu den Programmierern. Dabei basiert dieser Artikel auf der Fallstudie von Brand und Schmid (2004) und einem Arbeitspapier (Brand und Schmid 2005). Diese können bei den Autoren angefordert werden oder sind auf der Homepage (<http://www.soz.uni-frankfurt.de/arbeitslehre/pelm.html>) zu finden.

den Prozess der Umwandlung in eine individuelle Arbeitsleistung in Form eines Arbeitspakets. Schließlich umfasst sie drittens die Arbeitszusammenführung, d. h., wie die einzelnen Arbeitsleistungen der einzelnen Open-Source-Mitarbeiter zu einem gemeinsamen Produkt zusammengefügt werden.

Leistungserstellung und Arbeitszusammenführung erfolgen im Allgemeinen jederzeit in einem Open-Source-Projekt (vgl. Jørgensen 2001, 328 ff.; Mockus et al. 2002), d. h. auch im KDE-Projekt. Verschiedene parallel ablaufende Tätigkeiten, wie Programmieren, Testen, Releaseerstellung etc., können jeweils eher der Leistungserstellung oder der Arbeitszusammenführung zugeordnet werden. Zur Vereinfachung der Diskussion setzen wir die Leistungserstellung mit der dynamischen, freien Phase und die Arbeitszusammenführung mit der Phase der Erstellung eines stabilen, offiziellen Releases gleich.

In der Leistungserstellung entsteht zunächst ein sich ständig in Veränderung befindliches dynamisches und somit unfertiges Entwicklungsrelease mit uneingeschränkten und großen Quellcodeveränderungen. Diese Phase beinhaltet die Leistungserstellung, d. h. das freie und iterative Programmieren an einer lokalen Kopie von Teilen des Quellcodes in einem Versuchs- und Irrtumsprozess. Aus dem Entwicklungsrelease wird in der Arbeitszusammenführung eine abgeschlossene statische, möglichst fehlerfreie offizielle Endversion erstellt, wobei das gemeinsame Softwareprodukt in lauffähiger Maschinensprache mit Dokumentationen und Übersetzungen für die Öffentlichkeit zum Herunterladen freigegeben wird (vgl. Brand und Schmid 2004, 2005; Becher et al. 2005).

Im Folgenden referieren wir die zusammengefassten Untersuchungsergebnisse für die Koordination der Arbeit in dem Open-Source-Projekt. Die Aufteilung in die drei Phasen Arbeitsaufteilung, Leistungserstellung und Arbeitszusammenführung erfolgt aus analytischen Gründen, um eine primäre Zuordnung der Ausprägungen zu erreichen. Allerdings ist die Trennung nicht immer eindeutig möglich, was bei der Gleichzeitigkeit der drei Funktionen in diesem Open-Source-Projekt auch nicht anders zu erwarten ist. Bei der Arbeitsverteilung werden dabei nicht nur Aufgaben, sondern auch Posten mit spezifischen Aufgaben von den Mitarbeitern übernommen.

3.1 Arbeitsverteilung

Kooperation ist eine wesentliche Koordinationsform bei der Arbeitsverteilung von Aufgaben im KDE-Projekt, da ein gemeinsames Ziel angestrebt wird: die Entwicklung einer Software. Bei der Arbeitsverteilung der Postenbesetzung ist dagegen die Hierarchie zentral (s. u.). Von den Kooperationsarten überwiegt bei der Aufgabenverteilung die reziproke Kooperation, da vor allem die Entwicklungsrichtungen und -bereiche diskutiert werden. Bei den Abstimmungen wird über das Vorgehen und die verschiedenen individuellen Ziele bei der Arbeit gleichberechtigt verhandelt bzw. mittels besserem technischen Argument überzeugt, schreibt auch Ettrich (2004) als zentraler KDE-Entwickler. Dabei streben die „KDEler“ Konsens an; es gibt keine for-

Ist ein Open-Source-Projekt nur kooperativ?

malen (Abstimmungs-)Regeln. Bei den Absprachen legen die KDE-Projektbeteiligten ihre Arbeitsbereiche fest und/oder übernehmen Arbeitsbereiche freiwillig, wobei eine konsensuelle Zustimmung vorausgesetzt wird. Schnittstellen bzw. Interdependenzen zwischen Softwareteilen, z. B. mit den zentralen Bibliotheken, unterstützen dies durch eine gegenseitige Abhängigkeit. Die Wahrscheinlichkeit von Absprachen z. B. bei gemeinsamem Vorgehen steigt, wenn die Projektbeteiligten Reputation aufgebaut haben und somit von vornherein Vertrauen in die zukünftige Durchführung vorhanden ist (Interview 7).⁸

In vielen Fällen werden nur wenige oder gar keine Absprachen benötigt, etwa wenn KDE-Beteiligte im Rahmen der gruppenbezogenen Kooperation vernachlässigte Aufgaben,⁹ die normalerweise von anderen Projektbeteiligten bearbeitet wurden,¹⁰ oder Arbeitsaufgaben allgemeiner Unterstützungsanfragen in Mailinglisten oder Chats als Aushilfen übernehmen.

Entstehen Konflikte, versuchen die Kontrahenten zunächst, eine Einigung mittels überzeugender Argumente zu finden, erklärte ein zentraler KDE-Entwickler (Interview 6). Bei einem größeren Konflikt über längere Zeit, d. h. mehrere Tage, wird ein Konsens durch einen freiwilligen unparteiischen Schlichter aus dem KDE-Projekt herbeigeführt. Nach der überwiegend konsensuellen Einigung über die Arbeitsverteilung erfolgt die Handlungsdurchführung selbstverantwortlich.

Unterstützend wirkt die häufige altruistische Kooperation der projektexternen Endnutzer, die zwar nicht an der Arbeit in dem untersuchten Projekt beteiligt sind, aber einmalig Fehlerberichte, Wünsche oder programmierte Fehlerkorrekturen einsenden. Die hohe Anzahl von einmaligen Berichten führt zu einer sehr hohen Gesamtzahl aller Beiträge, auch während der Leistungserstellung (Interview 10).

Die hierarchische Koordination ist der Kooperation nachgelagert, da nur die manipulative Macht auftritt, die von Personen mit hoher Reputation ausgeht.¹¹ KDE-Mitarbeiter können über Sanktionen aufgrund der Virtualität und des Verlusts an intrinsischer Motivation die Übernahme von Aufgaben nicht erzwingen. Belohnungen in Form von Loben für die fortwährende Softwareerstellung und kommunikative Beteiligung durch Personen aus dem inneren Kreis der Reputationshierarchie kommen in der Kommunikation kaum vor, da die Belohnung durch Reputation erfolgt. Als zusätzliche Belohnungen können dabei die Vergabe einer E-Mail-Adresse der Domain kde.org oder die Aufnahme in Mailinglisten mit begrenzter Teilnehmerschaft

8 Die Autoren haben Interviews mit KDE-Mitgliedern geführt und verwenden im weiteren Verlauf des Artikels auch Selbstaussagen von der KDE-Homepage.

9 „I tend to look for things that no-one else is doing but are really in need of being done and just start working on it“ (Selbstaussage 29).

10 Der übliche Bearbeiter kann dies z. B. wegen Austritt, zeitlicher Abwesenheit, fehlender Kompetenzen etc. nicht ausführen (Interview 5).

11 Mit Arbeit kann Reputation erlangt werden, wie bei der Befragung festgestellt werden konnte: Die Antwortenden des inneren Kreises/Kernbereichs investieren mehr Arbeitszeit als die Mitarbeiter des äußeren Kreises.

gewertet werden (Interview 5).

Die manipulative Macht zeigt sich gerade bei KDElern mit hoher Reputation und/oder in Posten. Formales Bildungskapital wie Zeugnisse spielt dabei keine Rolle. Den höheren Einfluss auf die Entscheidungen haben sie einerseits durch die mit Reputation als symbolisches Kapital bzw. Reputationsmacht einhergehenden Erfahrungen bzw. das Wissen über das Projekt (Expertenmacht). Sie erkennen durch ihr Wissen Aufgabengebiete schneller und können diese besetzen bzw. darauf aufmerksam machen. Andererseits besteht der Einfluss der Posteninhaber durch die Verantwortung, die sie mit Postenübernahme eingehen. Dabei stehen die Posteninhaber auch in einer absteigenden hierarchischen Beziehung zueinander, die sich proportional zur Verantwortlichkeit über einen bestimmten Projektbereich, vom gesamtprojektbezogenen Posten (Releasekoordinatoren) bis hin zu (Unter-)Projektleitern, verhält. Oft übernehmen sie wichtige, aber unbeliebte Aufgaben. Es können aber Aufgaben nur vorgeschlagen und Mitarbeiter nur überzeugt werden (Interview 5/2). Die Aktionen müssen letztendlich vor den anderen Mitarbeitern argumentativ bestehen.

Institutionelle Unterstützung der manipulativen Macht bestehen durch die technische Infrastruktur, d. h., dass in wenigen Mailinglisten nur bestimmte KDEler mit hoher Reputation schreiben dürfen, wie in der „KDE core devel“-Mailingliste. Dadurch können bestimmte Problemsichten vordiskutiert und „vorentschieden“ werden (Interview 7). Eine Begrenzung ist, dass eine konsensuale Entscheidung oft nicht genau definiert ist und ein Freiwilliger den Spielraum nutzen und das Problem eher in seinem Sinne lösen kann.

Eine Koordination erfolgt auch durch die Steuerung der Aufmerksamkeit und strategische Kommunikation (Informationsmacht). Die KDE-Mitarbeiter mit hoher Reputation und Posteninhaber können Themen auf die Diskursagenda setzen, durch entsprechende Argumente verhindern oder wichtige Informationen aus sozialen Netzwerkkontakten bzw. aus ihrem Wissen weiterleiten oder auch nicht (Interview 5).

Die Identifikationsmacht des Projektleiters ist mit besonders freundlicher und konstruktiver E-Mail-Kommunikation verbunden, wobei die Aufgaben in KDE mit allgemein geteilten Werten verknüpft und manchmal ideologische Ziele definiert werden. Dabei stellt der Projektleiter ein Vorbild für die anderen dar, das Mitarbeiter anzieht. In Unterprojekten mit charismatischen Projektleitern besteht eine höhere intrinsische Motivation und Konflikte sind eher selten (Interview 4).

Bei der Arbeitsverteilung von Aufgaben spielt Wettbewerb eine geringe, wenn gleich auch signifikante Rolle, die sich aber eher auf die Motivation auswirkt. Die KDE-Projektbeteiligten wählen u. U. schwierigere und komplexe Arbeitspakete aus, um Reputation zu akkumulieren und einen zukünftigen Wettbewerbsvorsprung zu erhalten (Interview 8). Einen Wettbewerb bei der Übernahme eines Arbeitspakets aufgrund der Reaktionsgeschwindigkeit eines Projektbeteiligten gibt es nicht, da eine parallele Bearbeitung durch den offen zugänglichen Quellcode prinzipiell möglich ist.

Bei Diskussionen über größere Veränderungen treten manchmal zwei Parteien in

kleinem Rahmen auf, die um die Zustimmung der anderen Mitarbeiter zu einer der vorgeschlagenen Alternativen werben. Dabei vergleichen die Diskussionsbeteiligten die Argumente und wählen eine der Alternativen aus (Interview 3/9). Schweigen gilt dabei als Zustimmung, da davon ausgegangen wird, dass sich die Personen beteiligen, deren Interessen betroffen sind.

Bei der Besetzung von Posten mit spezifischen Aufgaben wie Projektleiter, Releasekoordinator etc. bestimmt der bisherige Posteninhaber seinen Nachfolger (Interview 3). Dabei spielt Wettbewerb um Reputation eine Rolle. Der Posteninhaber vergleicht den Grad der Reputation und wählt einen KDE-Mitarbeiter mit hoher Reputation aus. Beim Releasekoordinator z. B. wird mit Reputation Erfahrung erwartet, da er die Schwere eines Problems und die Wichtigkeit einer Software einschätzen muss. Sein soziales Kapital kann der Posteninhaber aber ausnutzen, um durch geschickte Wahl des Nachfolgers die weitere Entwicklung des (Unter-)Projekts zu beeinflussen. „Manipulativ“ ist dabei auch die freundlich-werbende charismatische Kommunikation zur Überzeugung eines adäquaten Nachfolgers. Die Wahl wird aber als allgemeiner Konsens betrachtet, wenn keiner etwas gegen den Nachfolger einzuwenden hat (Interview 8/10). Andere Projektbeteiligte aus dem inneren Kreis helfen dem bisherigen Posteninhaber bei der Suche nach Personen für diesen Posten durch eigene, unterstützende Suche, Erkundigungen oder Vorschläge, da die Besetzung wegen des hohen Arbeitsaufwands und hohen Erwartungsdrucks mögliche Nachfolger abschreckt (reziproke Kooperation).

Die bisherigen Positionsinhaber haben oft die Positionen als Gründer dieser Posten übernommen. Bisher gaben sie diese Posten wegen der erworbenen Reputation nicht auf, obwohl sie weniger für das Projekt arbeiteten. Ein Wettbewerb zwischen den Personen aus dem inneren Kreis scheidet aber wegen der überwiegend freundschaftlichen Verhältnisse und wiederholten persönlichen Treffen auf Konferenzen aus. Die nachfolgende Generation hat hohe Reputation akkumuliert und möchte diese Positionen übernehmen, weswegen z. T. neue, unwichtigere Posten als Ausweichmöglichkeiten geschaffen wurden (Interview 7). Der Wettbewerb zwischen den Generationen wirkt sich auf die alltägliche Softwareentwicklung nicht aus.

3.2 Leistungserstellung

Bei der Leistungserstellung überwiegt die gruppenbezogene Kooperation, da unaufgeforderte Hilfen bei der Softwareentwicklung häufiger auftreten als direkte Absprachen: „When it comes to KDE, the nicest thing is when somebody fixes bugs for me I introduced. However I really don't expect that, please don't get me wrong“ (Selbstaussage 52). Es kann außerdem jeder Fehlerberichte schreiben oder bei der Zusammenarbeit, bei Softwaretests und Softwareproblemen helfen, da das Fehlerberichtssystem und die meisten Mailinglisten für alle zugänglich sind (Interview 4).¹²

12 Siehe <http://developer.kde.org/source/anonsvn.html> bzw. <http://kde.org/maillinglists/>

Einerseits kontrollieren die KDE-Softwareentwickler sich selbst und andere durch die Nutzung der neuesten Quellcodeveränderungen im Versionsmanagementsystem (Peer-Review) und melden Fehler freiwillig über das standardisierende Fehlerberichts-system. Dabei wird kein Fehler einer Person zugeschrieben. Andererseits beantworten Projektbeteiligte mit dem nötigen Wissen unaufgefordert Anfragen in Mailinglisten oder Chaträumen zu Problemen mit dem Softwareprodukt, Werkzeugen etc. Außerdem bekommen sie erledigte Arbeiten zu ihrem Arbeitsbereich, wie Fehlerberichte oder -korrekturen, aus freien Stücken, da die andere Person auch zum Großprojekt gehört (Interview 3/9). Die reziproke Kooperation bezieht sich auf das gegenseitige Aushelfen bei kleinen Aufgaben und die temporäre Übernahme von kleinen Arbeit-spaketen nach persönlicher Anfrage. Projektbeteiligte z. B. bitten andere (oft beim Chatten), ihre Software zu kontrollieren bzw. zu testen.

Außerdem gibt es Nachverhandlungen von gemeinsamen Absprachen bei der Arbeitsverteilung. Selten, bei Verständnisproblemen von Fehlerberichten seitens des KDE-Softwareentwicklers, wird versucht, über Rückfragen bez. des gemeldeten Fehlers das Problem gemeinsam mit dem Fehlerberichts-Endenden, z. T. externen End-nutzern, zu lösen (Brand und Schmid 2004).

Als Hierarchieelement steht die „bestrafende“ Macht im Vordergrund, ist aber seltener vorzufinden als die Kooperation. Belohnungen wie monetäre Anreize spielen keine Rolle. Die Bestrafung kann nicht bez. der Softwareerstellung erfolgen, da Projektbeteiligte aufgrund der GPL-Lizenz rechtlich nicht von der Veränderung der Software ausgeschlossen werden können. Dagegen kann ein Ausschluss aus dem Open-Source-Projekt durch Entzug des Schreibzugriffs zum Dateiablagesystem von zentralen Posteninhabern vorgenommen werden, wenn die Normen des Großprojekts verletzt wurden bzw. Quellcode gegen die gruppenspezifischen Interessen verändert wird.¹³ Ein Entzug des Schreibzugriffs bei einer Mailingliste wird nicht vorgenommen, da gegen die Norm der frei fließenden Informationen verstoßen wird. Daher dominieren textbasierte gegenüber technischen Sanktionen. Personen mit hoher Reputation ahnden hin und wieder auftretende irrelevante, beleidigende und themenun-spezifische Aussagen oder schlechten Quellcode durch textbasierte soziale Sanktionen. Ignorieren ist oftmals die Sanktion der Wahl, vor allem bei Verstößen gegen die „Netiquette“. Ignorieren hat aber seine Grenze, wenn Einzelmeinungen als Aussagen des Gesamtprojekts missverstanden werden können und damit die Außendarstellung des Projekts beeinflussen (Interview 3/10).

Posteninhaber haben eine Verhinderungsmacht von seltenen gravierenden, nicht-normenkonformen Handlungen wie dem Einbau ungewünschten Quellcodes. Neben den seltenen Fällen des Entzugs der Schreibberechtigung für das Dateiablagesystem, kann gegebenenfalls die Entfernung von Quellcodeveränderungen und das Zurückspielen der alten Softwareversion aus dem gemeinsamen Dateiablagesystem durch

¹³ Siehe zur genaueren Darstellung des Konflikts um „moralische“ Autorenrechte Brand und Holtgrewe (2005).

den Projektleiter (oder Projektbeteiligte als Posteninhaber) als weitere bestrafende Sanktion gesehen werden. Der Projektleiter kontrolliert stichprobenartig den eingestellten Quellcode, wenn z. B. die beschreibenden Kommentare (Logs) der Veränderungen Verdacht erregen.¹⁴ Der Projektleiter hat durch Androhung der Löschung einen verstärkten Einfluss und kann dadurch z. B. eine höhere Qualität erzwingen. Motivationsverlust und Abwanderung von Projektbeteiligten begrenzen den Einsatz von bestrafenden Sanktionen, weswegen diese Sanktionen eher selten verwendet werden (Interview 2/11). Posteninhaber nehmen zusätzlichen Einfluss auf die konkrete Programmierung durch ihre ständige Erreichbarkeit, Hilfsbereitschaft und Erfahrung (Expertenmacht).

Ein verhaltener, verdeckter Wettbewerb besteht durch den Vergleich des eigenen Arbeitsfortschritts mit dem Fortschritt der anderen Unterprojektbeteiligten bzw. deren Reputation, was sich auf die Motivation auswirkt. Ein Aufstieg in der Reputationshierarchie ist durch eine ständig hohe Qualität der Arbeit (z. B. Fehlerfreiheit des Quellcodes) bei ständiger Mitarbeit möglich (Interview 5). Die Offenheit des Versionsmanagementsystems erhöht zwar die Vergleichbarkeit, ein gezielter Vergleich wird aber nicht verfolgt. Die Komplexität der Software macht es zusätzlich schwierig, einen direkten Vergleich mit Mitarbeitern von anderen Unterprojekten anzustellen. Ein sehr seltener direkter Vergleich des eingeschickten Quellcodes nach der Qualität findet statt, wenn ein Unterprojektkollege und ein Unterprojektexterner zu einem Fehler je eine Korrektur einsenden (Interview 1).

3.3 Arbeitszusammenführung

Bei der Arbeitszusammenführung zu einem offiziellen Softwareprodukt spielt die legitimierte Position des Releasekoordinators mit seiner Verhinderungsmacht bei der Erstellung des Releases eine große Rolle. Damit ist die Hierarchie bei der Arbeitszusammenführung am wichtigsten. Der Koordinator stellt den Releaseplan mit gestuften Einschränkungen der Veränderungsmöglichkeiten im Versionsmanagementsystem auf, überwacht dessen Einhaltung und testet die Software auf Fehlerfreiheit. Bei Übertretung des Plans als sozialer Norm oder bei mangelnder Fehlerfreiheit kann er Quellcode Teile löschen oder ganze Programmblöcke aus dem Release ausschließen. Bei einem Ausschluss von Quellcode spielt die Wichtigkeit bzw. Zentralität der Programme eine Rolle. So wird der Releasetermin für zentrale Software, die wichtig für die Funktionsfähigkeit des Endprodukts ist, (um einige Wochen) nach hinten gerückt und nur fehlerhafte Teile der Software werden entfernt. Bisher wurden nur periphe-

¹⁴ Dieser Normalzustand gilt für Langzeitmitglieder des (Unter-)Projekts. Für neue Mitglieder gilt eine stufenweise Annäherung an diesen Zustand. Der Projektleiter kontrolliert bei Eintritt eines neuen Mitglieds in sein (Unter-)Projekt erst jede Quellcodezeile. Einige Zeit nachdem das Neumitglied qualitativ hochwertigen Quellcode geliefert hat, werden nur noch die Veränderungskommentare (Logs) auf Verdächtiges hin kontrolliert. Nach einem weiteren Zeitraum ohne Probleme kontrolliert der Projektleiter nur noch sporadisch. Es hat sich Vertrauen zu der Person eingestellt.

re und stark fehlerbehaftete Programme komplett aus den Releases ausgeschlossen (Interview 3/9).¹⁵ In ganz seltenen Fällen wird die Zugangsberechtigung für das Dateiablagensystem entzogen, wenn unabgesprochene Änderungen oder ungetestete Software kurz vor dem Releasetermin eingespielt werden.

Normalerweise werden Probleme und Konflikte zwischen Releasekoordinator, den Unterprojektleitern und anderen Mitarbeitern diskutiert und es wird eine pragmatische Lösung gesucht. Das gemeinsame Interesse, ein stabiles, fehlerfreies, offizielles Produkt zu erstellen, steht dabei im Mittelpunkt. Es drückt sich auch in der Identifikation mit der Gruppe aus, was sich z. B. in Marken und Logos für das Gesamtprojekt bzw. -produkt und bestimmte Software oder in dem Stolz bei der Vorführung auf Konferenzen widerspiegelt (Interview 11).

Bei der Lösungssuche nutzt der Releasekoordinator seine Expertenmacht und seine starke Informationsmacht, da bei ihm alle Informationen zusammenlaufen. Bei einigen Personen erhöht sein Charisma zusätzlich die Legitimität von Entscheidungen. Oft muss er als Schlichter bei Konflikten über die Qualität von Software zwischen zwei Unterprojektbeteiligten fungieren. In seltenen Fällen, wenn keine Lösung gefunden wird, entscheidet der Releasekoordinator als letzte Instanz über Problem- und Konfliktlösungen (Interview 11).

Kooperation tritt vor allem als reziproke Kooperation zwischen Projektbeteiligten sowie zwischen Releasekoordinator und Projektbeteiligten auf. Letztere z. B. helfen sich gegenseitig bei der Fehlerbeseitigung. Zusätzlich ergibt sich ein Wettbewerb der Beteiligten um die begrenzte Aufmerksamkeit des Releasekoordinators in Form von Hilfe bei Entscheidungen, Problemlösung und aktiver Mitarbeit bei der Fehlerbeseitigung (Interview 9/11). Dieser Wettbewerb wird durch die gruppenbezogene Kooperation einer Gruppe von Personen mit hoher Reputation gemildert, die ein hohes Verantwortungsgefühl für das Gesamtprojekt haben und dem Releasekoordinator Arbeit abnehmen. Die Unterprojektleiter nehmen dem Releasekoordinator auch Arbeit ab, da sie für das Unterprojekt und die Fehlerfreiheit der Software verantwortlich und ihm daher untergeordnet sind. Sie haben dadurch eine abgeleitete „Macht“ über die anderen Projektbeteiligten und kontrollieren, testen und lehnen Softwareveränderungen gegebenenfalls ab. Dies geschieht besonders in einem Zeitraum kurz vor dem Veröffentlichungstermin, um unbedachte Fehlerentstehung zu vermeiden.

4 Zusammenfassung

Im untersuchten Open-Source-Projekt KDE erfolgt die projektinterne Arbeitsallokation nicht nur kooperativ, sondern durch eine Kombination von Wettbewerb, Hierarchie und Kooperation; die Kombination ist dabei unterschiedlich ausgeprägt. Insgesamt dominiert zwar die Kooperation, Hierarchie hat aber ebenfalls Bedeutung

¹⁵ Das Ausschlusskriterium ist das Auftreten kritischer Fehler bei der Nutzung (z. B. Systemabstürze).

Ist ein Open-Source-Projekt nur kooperativ?

Koordinations- formen	Arbeits- verteilung von Aufgaben	Arbeits- verteilung von Posten	Leistungs- erstellung	Arbeits- zusammen- führung
Kooperation	+++	++	+++	++
Hierarchie	++	+++	++	+++
Wettbewerb	+	+	+	+

Tabelle 1: Bedeutung der Koordinationsformen bei der projektinternen Allokation, +++ „sehr hohe Bedeutung“; ++ „hohe Bedeutung“; + „kaum Bedeutung“

bei der Koordination. Wettbewerb ist vorhanden, spielt aber eine untergeordnete Rolle. Es ist anzunehmen, dass diese Elemente auch in anderen Open-Source-Projekten auftreten.¹⁶

Die Untersuchungsergebnisse verweisen auch auf ein komplementäres Verhältnis der drei Koordinationsformen (Tabelle 1). Als vorsichtige allgemeine Schlussfolgerung lässt sich aus der Fallstudie festhalten, dass bei der Arbeitsverteilung Kooperation dominiert, die Hierarchie eine eher geringe Rolle spielt. Bei der Leistungserstellung überwiegt zwar die Kooperation, doch bestehen auch hierarchische Elemente. Bei der Arbeitszusammenführung aufgrund der Kontrollnotwendigkeiten der Vereinheitlichung der Software und des Projektzusammenhalts spielt die Hierarchie eine wesentliche Rolle, bei etwas geringerer, aber vorhandener Bedeutung auch der Kooperation. Wettbewerb hat hier eine geringe Bedeutung und wirkt sich eher auf die Motivation der Beteiligten aus.

Kooperative, aber auch hierarchische Elemente der Koordination finden sich eher in der Organisationsstruktur bei der Besetzung von Stellen und der damit verbundenen Handlungsmöglichkeiten. Wettbewerb hat auch hier einen geringen Stellenwert.

Literatur

- Arentzen, U. und Lörcher, U. (Hrsg.) (1997), *Gabler-Wirtschafts-Lexikon*, 14. Aufl., Gabler, Wiesbaden.
- Becher, C., Brand, A., Kawai, A. und Schmid, A. (2005), *Elektronische Arbeitsmärkte. Empirie der Struktur und Funktionsweise elektronischer Arbeitsmarktplätze in Teilarbeitsmärkten*, Rainer Hampp Verlag, München und Mering.
- Blau, P. M. (1964), *Exchange and Power in Social Life*, Wiley, New York, USA.

¹⁶ Hinzuweisen ist hierbei noch darauf, dass sich vor allem beim Wettbewerb und bei der Hierarchie die Instrumente der Koordination von den herkömmlichen wie dem „Marktwettbewerb“ und der „Unternehmenshierarchie“ unterscheiden.

- Bourdieu, P. (1987), *Die feinen Unterschiede. Kritik der gesellschaftlichen Urteilskraft*, Suhrkamp, Frankfurt.
- Bourdieu, P. (1998), *Praktische Vernunft. Zur Theorie des Handelns*, Suhrkamp, Frankfurt. Original von 1985.
- Brand, A. und Holtgrewe, U. (2005), Open Source Software Entwicklung und öffentliche Güter, in M. Moldaschl und N. Stehr (Hrsg.), 'Knowledge Economy. Beiträge zur Ökonomie der Wissensgesellschaft', Metropolis, Marburg.
- Brand, A. und Schmid, A. (2004), 'Fallstudie eines Open Source-Projekts als Arbeitsnetz'. <http://www.soz.uni-frankfurt.de/arbeitslehre/pelm/docs/FALLSTUDIE%20Open%20Source%20V1.pdf> [28. Jan 2006].
- Brand, A. und Schmid, A. (2005), 'Koordination in einem Open Source-Projekt. Arbeitspapier aus dem DFG-Projekt Struktur und Funktionsweise elektronischer Arbeitsmärkte'. http://www.soz.uni-frankfurt.de/arbeitslehre/pelm/docs/KOORDINATION_IN_OPEN_SOURCE-PROJEKT_Brand-Schmid.pdf [10. Jan. 2006].
- Ettrich, M. (2004), Koordination und Kommunikation in Open Source-Projekten, in B. Lutterbeck und R. A. Gehring (Hrsg.), 'Open Source Jahrbuch 2004 – Zwischen Softwareentwicklung und Gesellschaftsmodell', Lehmanns Media, Berlin, S. 179–192. <http://www.opensourcejahrbuch.de/2004/> [10. Feb 2006].
- Ghosh, R. A. (1998), 'Cooking-Pot-Markets: An Economic Model for the Trade in Free Goods and Services over the Internet', *First Monday* 3(3). http://firstmonday.org/issues/issue3_3/ghosh/ [10. Jan 2006].
- Götzenbrucker, G. und Löger, B. (2000), Online Communities. Struktur sozialer Beziehungen und Spieler Motivationen am Beispiel von Multi User Dungeons, in U. Thiedecke (Hrsg.), 'Virtuelle Gruppen. Charakteristika und Problemdimensionen', Springer, Wiesbaden.
- Hradil, S. (1980), *Die Erforschung der Macht: Eine Übersicht über die empirische Ermittlung von Machtverteilungen durch die Sozialwissenschaften*, Kohlhammer, Stuttgart.
- Jaeger, T. und Metzger, A. (2002), *Open Source Software. Rechtliche Rahmenbedingungen der Freien Software*, Beck, München.
- Jørgensen, N. (2001), 'Putting it all in the Trunk: Incremental Software Development in the FreeBSD Open Source Project', *Information Systems Journal* 11(4), S. 321–336. [10. Jan 2006].
- Kawai, A. und Schmid, A. (2004), Beschäftigungsflexibilität durch interne Jobbörsen. Wettbewerb und Hierarchie auf unternehmensinternen elektronischen Arbeitsmärkten, in O. Struck und C. Köhler (Hrsg.), 'Beschäftigungsstabilität im Wandel? Empirische Befunde und theoretische Erklärungen für West- und Ostdeutschland', Rainer Hampp, München und Mering, S. 247–265.
- Lerner, J. und Tirole, J. (2000), The Simple Economics of the Open Source Software, Working Paper 7600, NBER. <http://www.nber.org/papers/w7600> [10. Jan 2006].

Ist ein Open-Source-Projekt nur kooperativ?

- Mahnkopf, B. (1994), Markt, Hierarchie und soziale Beziehungen. Zur Bedeutung reziproker Beziehungsnetzwerke in modernen Marktgesellschaften, in N. Beckenbach und W. van Treeck (Hrsg.), 'Umbrüche gesellschaftlicher Arbeit', Schwartz, Göttingen. Soziale Welt Sonderband 9.
- Mockus, A., Fielding, R. und Herbsleb, J. (2002), 'Two Case Studies of Open Source Software Development: Apache and Mozilla', *ACM Trans. Software Engineering and Methodology* **11**(3), S. 309–346.
<http://www.research.avayalabs.com/techreport/ALR-2002-003-paper.pdf> [10. Jan 2006].
- Noteboom, B. (2002), *Trust. Forms, Foundations, Functions, Failures and Figures*, Edward Elgar, Cheltenham.
- Raven, B. (1992), 'A Power/Interaction Model of Interpersonal Influence: French and Raven thirty Years later', *Journal of Social Behavior and Personality* **7**, S. 217–244.
- Raymond, E. S. (1998), 'Homesteading the Noosphere', *First Monday* **3**(10).
http://firstmonday.org/issues/issue3_10/raymond [10. Jan 2006].
- Raymond, E. S. (1999), 'The Magic Cauldron'.
<http://www.catb.org/~esr/writings/magic-cauldron/magic-cauldron.html> [10. Jan 2006].
- Rossi, M. A. (2004), Decoding the „Free/Open Source (F/OSS) Software Puzzle“ - A Survey of Theoretical and Empirical Contributions, Working Paper 424, Università di Siena, dipartimento di economia politica. <http://opensource.mit.edu/papers/rossi.pdf> [10. Jan 2006].
- Stegbauer, C. (2002), *Reziprozität. Einführung in soziale Formen der Gegenseitigkeit*, Westdeutscher Verlag, Wiesbaden.
- Weber, M. (1956), *Wirtschaft und Gesellschaft*, J.C.B. Mohr (Paul Siebeck), Tübingen.