

Quintela – Der Einstieg in OSS-Projekte



Seminar „Ausgewählte Beiträge
zum Software Engineering“

Sommersemester 2006

Der Einstiegsprozess in Open Source Software Projekte

Luis Quintela García

quintela@inf.fu-berlin.de

Advisor: Christopher Oezbek

17. Juli 2006

Zusammenfassung

Der Erfolg von Open Source Software Projekten hängt von der Anzahl ihrer Mitglieder ab. Dafür ist der Einstiegsprozess von neuen Projektmitgliedern entscheidend. Diese Arbeit betrachtet den Einstiegsprozess und seine Optimierungsmöglichkeiten.

Inhaltsverzeichnis

1. Die Aufnahme von neuen Teilnehmern in OSS-Projekte
2. Struktur eines OSS-Projekts in Hinblick auf den Neueinstieg
3. Die virtuelle Identität des Projektteilnehmers in der sozialen Struktur eines OSS-Projekts
4. Interessendurchsetzung und politische Vorgänge in OSS-Projekten
5. Einfluss gewinnen in Entwickler-Kreisen von OSS-Projekten
6. Schlussfolgerungen

1. Die Aufnahme von neuen Teilnehmern in OSS-Projekte

Diese Arbeit widmet sich dem Thema des Einstiegs und Einflussnahme-Prozesses von Neulingen in Open Source Software (OSS) Projekten. OSS-Projekte produzieren in der Regel frei verwendbare Software und zeichnen sich dadurch aus, dass sie meistens auf freiwilliger Arbeit basieren, aus einer breit verteilten Teilnehmerschaft bestehen und die erzielte Softwareherstellung fast ausschließlich über Internet mit zentralen Server-Systemen (CVS¹, Mailinglisten) koordinieren. Der Einfluss eines jeden Teilnehmers auf das Projekt verhält sich proportional zu dessen CVS-Zugriffsrechten². Die Zugriffsrechte werden verteilt nach den sozialen Vorschriften, die in der Community gelten.

In diesem Kontext zeigt sich die Aufnahme von neuen Mitgliedern als ein zentraler Punkt in der Entwicklung eines jeden OSS-Projekts, nach Duchenaut (2005) „the socialization of newcomers is particularly crucial for the Open Source movement. First, the success of a project appears to be highly correlated with its size. (Duchenaut, 2005, S. 326)

Um den Einstiegs-Prozess von neuen Teilnehmern in OSS-Projekte zu thematisieren, werde ich die Natur der sozialen Interaktion zwischen OSS-Projektteilnehmern im Zusammenhang mit ihrer Identität als Projektteilnehmer diskutieren.

2. Struktur eines OSS-Projekts in Hinblick auf den Neueinstieg

Ein geläufiges Modell für die Betrachtung der sozialen Struktur von OSS-Projekten ist das Modell der Rollenhierarchie. Eine Ausprägung dieses Modells (vgl. z.B. Maas, 2004) ergibt fünf Schichten:

- core-developers
- Wartungsschicht
- Patchers
- Bug- Berichterstatter

1 Concurrent Versions System: beinhaltet den aktuellen Code des Softwareprodukts der Community. Das erlaubt Änderungen an der Software durch individuelle Teilnehmer und dadurch die Entstehung einer neuen Version des Produktes.

2 „Zugriffsrechte“ meint hier nicht nur die technische Zugriffsrechte, sondern zugleich die Rechte auf CVS-Zugriff, die aus der sozialen Interaktion der Community erteilt werden.

Quintela – Der Einstieg in OSS-Projekte

- Dokumentierer

Nach solch einer, eher statischen, Betrachtungsweise ernährt sich ein OSS-Projekt von den Beiträgen seiner Mitglieder und das spezifische Gewicht solcher Beiträge hängt ausschließlich davon ab, welche Rolle der Teilnehmer im Projekt spielt, d.h. welcher Schicht er aktuell angehört. In dieser Sichtweise würde die soziale Beförderung³ eines Mitglieds allein von der Qualität seiner Beiträge abhängen. Demzufolge würden Einsteiger je nach ihren Lernerfolgen weiter befördert werden.

In Erweiterung dazu führt Ducheneaut (2005) auf empirischer Grundlage ein dynamisches Modell zur Betrachtung der sozialen Struktur von OSS-Projekten ein, das auf sozial-technischen Netzwerkstrukturen anstatt auf „sozialen Schichten“ basiert.

Um die Sozialisierung von Einsteigern in OSS-Projekten im Detail zu erforschen, führte Ducheneaut eine empirische Untersuchung der Python-Community durch, in der er die über elektronische Systeme zustande gekommenen Interaktionen der Projektmitglieder aufzeichnete. Er verwendete dafür einen Browser, der die Zusammenhänge/ Kontakte zwischen Teilnehmern untereinander, Teilnehmern und Code sowie Code und Code graphisch darstellte. So konnte er die Diskussionsstränge in der Community mit ihren Teilnehmern und dem daraus tatsächlich implementierten Code in Zusammenhang bringen und dies mit Hilfe von Netzwerkgraphiken darstellen.

Seine Ergebnisse zeigten, dass erfolgreiche Einsteiger besonders viele Kontakte mit anderen Teilnehmern pflegten. Dabei waren die Kontaktpersonen selektiv ausgewählt, so z.B. Entwickler deren Quellcode der Einsteiger verwenden wollte oder Personen, die einen besonderen Einfluss in Entscheidungsprozessen des Projekts hatten. Die Kontakte zu Entwicklern (bzw. maintainers) von Code, an deren Modulen der Einsteiger interessiert war, bestanden insbesondere daraus, ihnen Patches für ihre Module anzubieten. Weiterhin waren besonders viele Kontakte (Lesezugriffe) vom Einsteiger auf den Quellcode dieser Entwickler und umgekehrt zu erkennen und schließlich ein häufiges Zusammenwirken des Codes des Einsteigers mit dem Quellcode dieser Entwickler.

Im Mittelpunkt dieser Betrachtung steht ein Netzwerk von Individuen, die sich gegenseitig fast ausschließlich über ihre elektronisch gespeicherten Beiträge (Code, Diskussionsbeiträge, Übersetzungen, Dokumentation, Homepage-Gestaltung etc.)

³ Als Erwerb von mehr Zugriffsrechten

Quintela – Der Einstieg in OSS-Projekte

wahrnehmen können. Das Zusammenspiel zwischen den Beiträgen und ihren Produzenten, bestehenden und neuen Beiträgen, Produzenten von neuen Beiträgen und bestehenden Beiträgen bzw. ihren Autoren ergeben ein stark verlinktes Netz, über das das Endprodukt des OSS-Projekt bestimmt wird.

Ducheneaut (2005) nennt das ein hybrides Netzwerk, um zu betonen, dass das Netzwerk nicht nur aus Individuen, sondern auch aus materiellen Ressourcen besteht.

I think it is crucial to take the hybrid nature of OSS more seriously, and to simultaneously investigate the complex networks of people and material resources constitutive of OSS projects. This has consequences for the analysis of socialization in these projects: instead of a linear progression from role to role based purely on the quality of code contributions ... I argue we need to analyze OSS socialization as the active creation and maintenance of strong links between individual participants and the socio-technical network of a project. (Ducheneaut, 2005, S. 328)

Mahendran (2002) geht noch weiter und betont, dass es sich bei diesem Netzwerk letztendlich um Text im virtuellen elektronisch erzeugten Raum handelt:

The multiple components [of an Open Source project] that at first seem to be hard material are in essence text... This distributed network of people and things is constructed through the materialization of language... There is a hybridism of dialogue and code, where the dialogue is directly embedded in the code (Mahendran, 2002, S. 13–14).

3. Die virtuelle Identität des Projektteilnehmers in der sozialen Struktur eines OSS-Projekts

Die Tatsache, dass die einzelnen Projektteilnehmer einander vorwiegend über ihre jeweiligen elektronischen Beiträge kennen, impliziert, dass die Identität eines Mitglieds in der Community über seine elektronischen Beiträge definiert ist. „Mit virtueller Identität ist eine dienst- oder anwendungsspezifische, mehrfach in konsistenter und für andere Menschen wiedererkennbarer Weise verwendete, subjektiv relevante Repräsentation einer Person im Netz gemeint“ (Döring, 2003, S. 341).

Die Zugehörigkeit zur OSS-Community ist durch die Sichtbarkeit eines Mitglieds, d.h. durch seine virtuelle Präsenz bestimmt. Nur wer in der Community durch Beiträge sichtbar ist, existiert für die Community. Die Anzahl bzw. Qualität der Beiträge eines Mitglieds und daraus folgend sein Ruf modelliert seine virtuelle Identität.

Quintela – Der Einstieg in OSS-Projekte

Im Fall von Entwicklern wird ein potenzielles neues Mitglied erst sichtbar für die Community, wenn es sich in Mailinglisten/ Foren des jeweiligen OSS-Projektes äußert und z.B. seine Mitarbeitsbereitschaft signalisiert, sich über konkrete technische Themen äußert oder seine technischen Fertigkeiten schildert. In dieser Phase ist die virtuelle Identität des Neuen noch sehr schwach ausgebildet. Für die anderen Teilnehmer ist er immer noch ein Unbekannter. Mögliche Schritte für die Ausgestaltung der virtuellen Identität des Einsteigers sind z.B. Bugs zu lokalisieren und wenn möglich Patches dafür anzubieten oder sich an technischen Diskussionen über konkrete Themen sinnvoll zu beteiligen. Bis zu dem Zeitpunkt, zu dem ein Neueinsteiger CVS-Zugriffsrechte bekommt, sind jedoch die Möglichkeiten der Ausgestaltung seiner virtuellen Identität noch begrenzter, da er z.B. der Hilfe des Maintainers eines bestimmten Moduls bedarf, damit sein Patch-Code auf das CVS eingespielt wird und somit für die anderen Teilnehmer sichtbar ist. Dieser Umweg über den Maintainer stellt noch eine Hürde für die Ausgestaltung der eigenen virtuellen Identität des Einsteigers dar. Das bedeutet unter Umständen eine Einschränkung der Lösungsmöglichkeiten (und somit der Ausgestaltungsmöglichkeiten der virtuellen Identität) des Einsteigers, da der Maintainer letztlich die Entscheidung darüber trifft, welcher Code committed und somit für die anderen Projektteilnehmer sichtbar wird.

Das Phänomen, dass es einer gut entwickelten virtuellen Identität bedarf, um als Vollmitglied akzeptiert zu werden, wird z.T. explizit anerkannt. So ist z.B. auf der Web-Site des Python-Projekts vermerkt: „If the python-dev team knows who you are, whether through mailing list discussion, having submitted patches, or some other interaction, then you can ask for full CVS access.“ (Python, 2006).

Der Code eines Entwicklers auf dem CVS ist wesentlicher Bestandteil seiner virtuellen Identität, denn er hat in gewissem Sinne kommunikativen Charakter: In jedem Quellcode drücken sich u.a. technische Ansichten und Fertigkeiten des Entwicklers aus. „Email, code and databases constitute not simply the end products of OSS development efforts ... but also material means that OSS participants interact with and through“ (Duchenaute, 2005, S. 327).

Der Code eines Entwicklers drückt einerseits die technischen Ansichten des Entwicklers selbst aus, andererseits aber auch die technischen Ansichten aus Diskussionen in der Community, an denen er teilgenommen hat. Für einen Einsteiger besteht eine besonders subtile Hürde darin, diese technischen Ansichten

Quintela – Der Einstieg in OSS-Projekte

nachzuvollziehen, die Relevanz dieser technischen Entscheidungen/ Lösungen im Kontext der Entwicklung zu verstehen und zu erkennen, welche anderen Menschen daran beteiligt sind. „To work with this large and dispersed group, you'll have to learn who's the right person to answer a question [and] how to convince the others developers of the usefulness of a patch“ (Python, 2006).

4. Interessendurchsetzung und politische Vorgänge in OSS-Projekten

Nach der Selbstdarstellung von OSS-Projekten können Entscheidungsprozesse auf demokratischer Basis bzw. auf Expertenbasis zustande kommen. Vor allem in der ersten Version ist das meiner Meinung nach eine idealisierende Beschreibung, die die Natur von virtuellen Identitäten in einem OSS-Netzwerk nicht berücksichtigt. Tatsache ist, dass in jeder OSS-Community elektronische Ressourcen zur Verfügung gestellt werden, auf die es nur einen beschränkten Zugriff gibt. Die Verwaltung von Zugriffsrechten auf Ressourcen (CVS, bestimmte Mailinglisten usw.) induziert ein Geflecht von Interessen um die Entfaltung von virtuellen Identitäten.

Interessen von einzelnen Projektteilnehmern können, müssen aber nicht, mit dem Wohl des Projektes übereinstimmen, aber sie haben immer zum Ziel, die virtuelle Identität eines Teilnehmers zu erhalten oder auszugestalten (d.h. möglichst viele persönliche „Spuren“ im Code im CVS, in Diskussionsbeiträgen in Mailinglisten oder in der Organisation des Projekts zu hinterlassen), da das die Mitgliedschaft in der Community sichert. "In online communities, control over artifacts is a central source of power and influence ... and textual resources are manipulated to project a participant's "virtual" identity"(Duchenaut, 2005, S. 327).

Das heißt die Ausgestaltung der virtuellen Identität eines Teilnehmers ist entscheidend für seinen Ruhm und Einfluss in der Community.

Im Gegensatz dazu stellen Änderungen oder gar die Entfernung von Teilen einer individuellen Beitragsmenge die virtuelle Identität eines Teilnehmers in Frage. „Ein abgeschwächter Sanktionsmechanismus ist der Ausschluss bzw. die Ablehnung von Arbeitspaketen aus dem Dateiablagensystem bzw. Release.“ (Brand, unbekanntes Datum, S. 31).

Jeder neue Beitrag zum Projekt kann potenziell Änderungen an

Quintela – Der Einstieg in OSS-Projekte

bestehenden Beiträgen bewirken bzw. ältere Beiträge überflüssig machen. So erzeugt die Überzeugungsarbeit eines Teilnehmers von der Güte seines neuen Vorschlags notwendigerweise Interessenkonflikte mit anderen Teilnehmern und kann deren Widerstand gegen den neuen Vorschlag hervorrufen.

So ist z.B. Entwicklern zur Anfangszeit einer Diskussion über einen neuen Vorschlag (Modul) noch nicht unbedingt klar, welche Softwareteile des Projektes davon betroffen sein könnten. Erst im Laufe der Diskussion kann jeder sich eine Meinung darüber bilden. Der aktuelle Entwicklungszustand des Projektes ist zudem das Ergebnis der Arbeit der Entwickler und erfüllt die damals angestrebten Funktionalitäten. Das kann zur Folge haben, dass das Netzwerk einen latenten Widerstand gegenüber Änderungen hervorbringt.

Das bedeutet, es ist „the entire network, designed to resist change, which must be weakened in strategic areas and eventually reconfigured if a participant’s contribution is to be accepted (Ducheneaut, 2005, S. 353).

5. Einfluss gewinnen in Entwickler-Kreisen von OSS-Projekten

In der Literatur werden verschiedene grundsätzliche, mehr oder weniger klar explizierte Anforderungen aufgezählt, die ein Einsteiger erfüllen muss, um in den Entwickler-Kreis der Community eines OSS-Projektes aufgenommen zu werden (vgl. von Krogh et al.; 2003, Brand, 2006; Ducheneaut, 2005).

So muss der Neuling grundsätzlich als Qualifikation/ Eintrittskarte nachweisen, dass er „wirklich Lust“ hat, dauerhaft am Projekt mitzuarbeiten. Diese Hürde besteht, weil die Entwickler des Projekts zwischen echten Mitarbeitern und „Trittbrettfahrern“ unterscheiden wollen, um zu entscheiden, in wen sie Aufmerksamkeit investieren (vgl. Brand, 2006). Weiterhin wird von Neulingen selbständiges Arbeiten gefordert, auch in der Hinsicht, dass ihre Beiträge den anderen Projektteilnehmern keine zusätzliche Arbeit verursachen (vgl. Brand, 2006). Selbst das Erlernen von Projekttechnologien soll möglichst selbständig geschehen und keine Belastung für andere Mitglieder sein, d.h. so wenige Fragen wie möglich stellen bzw. sich mit Fragen möglichst direkt an die zuständige Person wenden. Der Neuling muss außerdem das Produkt des Projektes gut kennen (vgl. Brand, 2006; Ducheneaut, 2005). Voraussetzung für die Aufnahme als Entwickler

Quintela – Der Einstieg in OSS-Projekte

ist außerdem, dass der Neuling bereits über entsprechende technische Fertigkeiten verfügt (vgl. Ducheneaut, 2005).

Zudem muss der Neuling die Konventionen sowie die geschriebenen und ungeschriebenen Regeln des OSS-Projekts berücksichtigen, Interaktions-Regeln in den Mailinglisten (z.B. welche Art von Kritik ist zulässig, wie geht man mit Kritik an der eigenen Arbeit um), Konventionen bezüglich des Codes und Umgang mit dem CVS, insbesondere sind die Regeln für die Nutzung des CVS dafür angelegt, dass es nicht zum Verlust von Codebeiträgen kommt (vgl. Brand, 2006; Python, 2006).

Obwohl bereits an dieser Aufzählung deutlich wird, dass für den erfolgreichen Einstieg in ein OSS-Projekt vielfältige unterschiedliche Kompetenzen und Fertigkeiten erforderlich sind (also nicht nur technisches Können, sondern auch eine gewisse soziale Kompetenz) wird der Einstiegsprozess in ein OSS-Projekt als ein bloßer Lernerfolgs-bezogener Vorgang analog zu klassischen Lehrausbildungen konzeptualisiert (vgl. von Krogh et al.; 2003). Ducheneaut (2005) weist jedoch darauf hin, dass der Neuling in einem OSS-Projekt nicht wie ein Lehrling im Handwerk den direkten Kontakt mit dem Meister hat, von dem er in umfassender Weise alle entscheidenden Fähigkeiten lernen könnte und der dann seinen Lernerfolg objektiv beurteilen könnte. Die Anforderung der selbständigen Einarbeitung und die Forderung, den aktiven Vollmitgliedern keine zusätzliche Arbeit zu machen, widerspricht dem Lehrling-Meister-Modell sogar explizit.

Aus all dem wird deutlich, dass dem Aufstieg von Einsteigern komplexere Vorgänge zugrunde liegen müssen, als eine reine Lernerfolgs-bezogene Beförderung in den Kreis der Entwickler, wie sie nach dem Schichtenmodell nahe gelegt wird.

Wie kommt dann aber die Beförderung zum Entwickler zustande, wenn es niemanden gibt, der den konkreten „Lernerfolg“ eines Einsteigers kontrollieren wollte oder könnte? Offenbar müssen dabei noch andere Mechanismen wirksam sein.

In der Untersuchung der Freenet-Community von von Krogh et al. (2003) wurde deutlich, dass ein relativ kleiner Anteil von erfolgreichen Einsteigern (etwa 16% in seiner Untersuchung) schon bei der Kontakt-Aufnahme mit der Community Erweiterungscode bzw. den Code für neue Tools beigelegt hatten. Das bezeichnet er als Geschenke an die Community und sieht darin eine mögliche Erleichterung der Überwindung der von Einsteigern zu überwindenden Hürde.

Quintela – Der Einstieg in OSS-Projekte

In Ergänzung der oben genannten grundlegenden Anforderungen an Einsteiger in OSS-Projekte postuliert Ducheneaut (2005) auf Grundlage seiner Forschung zum Python-Projekt, dass die Beförderung von Einsteigern über „politische“ Vorgänge geschieht. So müsse der Einsteiger beim Aufbau seiner virtuellen Identität so viele und gute Verbindungen im hybriden Netzwerk zu schaffen, dass schließlich ein Großteil der Developer den Eindruck gewinne, dass der Einsteiger präsent und kompetent sei und viel tue für das Projekt. Dabei geht es aber in erster Linie um den Eindruck der Core-Entwickler, also nicht im engeren Sinne um eine „objektive“ Qualität des Beitrags des Einsteigers, sondern um seine Interaktionsfähigkeiten im hybriden Netzwerk der OSS-Community.

Praktisch könnte das bedeuten, dass der Einsteiger ein Stück Quellcode schreibt, das mit dem Quellcode von Developer A zusammenspielen soll. Dabei hat der Einsteiger zunächst den Quellcode von Developer A genau betrachtet (Struktur, Strategie, Konventionen, ...) und mit A darüber diskutiert. Also kann A darauf vertrauen, dass der Einsteiger in technischer Hinsicht so handelt, wie es den Wertvorstellungen von A entspricht und somit wird A sich dafür einsetzen, dass der Quellcode des Einsteigers letztendlich auf das CVS gespielt wird. Allerdings ist dieses Beispiel mit nur zwei Beteiligten sehr vereinfacht. In der Realität wird es für den Einsteiger eine echte Herausforderung sein, zu erkennen, an wen er sich im Einzelfall wenden soll, welcher Code mit welchem anderen etwas zu tun hat und welche Zusammenhänge im vielgestaltigen hybriden Netzwerk er kennen muss, um zielgerichtet interagieren zu können. Hoffentlich wollte der Neuling nicht z.B. aus Versehen gerade einen „Bug“ reparieren und hat sich mit seiner Idee/ seinem Code gegen etwas gestellt, was aus den unterschiedlichsten Gründen (z.B. Softwarearchitektur, Kompatibilität von Komponenten, Gewohnheit, Konvention, ...) von wichtigen Projektteilnehmern so entschieden wurde:

If someone had a grandiose scheme in mind when they submitted the report, be prepared for wrath when you apply a simple solution. Realize that, in some cases, someone, somewhere is relying on the undocumented buggy behavior and your fixing it is breaking their code (Python, 2006).

Wie komplex dieser Vorgang der Schaffung eines solchen Netzwerks sein kann, wird aus Ducheneauts graphisch dargestellten Netzen der Interaktion des Einsteigers Fred in die Python-Community deutlich (Ducheneauts, 2005, S. 341-348).

6. Schlussfolgerungen

Aus dem bisher Diskutierten scheint der Einstiegsprozess in OSS-Projekte in solch komplexe soziale Zusammenhänge eingebettet zu sein, dass jedes schlichte Erklärungsmodell für den erfolgreichen Einstieg in die Vollmitgliedschaft vereinfachend und ungenau sein muss.

Trotzdem werden gewisse Erkenntnisse gewonnen, die eine Auflistung von möglicherweise erfolgreichen Vorgehensweisen erlauben. Diese Punkte sollten jedoch in jedem konkreten Kontext auf ihre Relevanz überprüft werden.

Die idealtypischen Schritte für einen erfolgreichen Einstieg in ein OSS-Projekt wären somit (erweitert nach Ducheneaut, 2005):

1. Zunächst ein „passives“ Beobachten der Entwicklungsaktivitäten in der Community, um ein Vorverständnis der wichtigsten Strukturen, wichtigen Personen, wichtigen technischen Themen und wichtigen vorhandenen Software-Maschinen zu entwickeln und deren Zusammenwirken so weit wie möglich zu erschließen.
2. Inhaltlich sinnvolle Beteiligung an den technischen Diskussionen in den Mailinglisten. Damit Beginn des Aufbaus der eigenen virtuellen Identität und möglicher Netzwerk-Aufbau. Diese Aktivität muss sich allerdings über den ganzen Prozess hinziehen und gemäß der eigenen Interessen strategisch eingesetzt werden.
3. Berichten von Bugs und Vorschlägen möglicher Lösungen dafür. Dafür ist es sinnvoll, sich auf Module zu konzentrieren, die für das eigene Vorhaben wichtig sind, um durch die Bug-Suche ein tieferes technisches Verständnis der Modul-Architektur zu gewinnen. So kann der Lernaufwand minimiert werden.
4. CVS-Zugang bekommen und somit die Möglichkeit, Bugs direkt zu beheben bzw. in der Lage sein, eigene Module vorzuschlagen.
5. Mitarbeit in einem Teilprojekt bzw. dessen Übernahme. Hierbei ist es nötig, dass der Einsteiger schon am Anfang sich für dieses Teilprojekt interessiert hat. Ansonsten würde es zu einer Verzögerung kommen, weil er sich zuerst intensiv mit der Architektur dieses Teilprojekts vertraut machen müsste.

Quintela – Der Einstieg in OSS-Projekte

6. Entwicklung des eigenen Projekts, Unterstützung im Netzwerk dafür organisieren, es öffentlich bewerben, verteidigen und, wenn nötig, für Änderungen in anderen Teilen des Gesamtprojekts sorgen.
7. Das Einverständnis der Core-Members einholen und das Teilprojekt in die die Architektur des Gesamtprojekts eingliedern.

Das Problem an dem hier beschriebenen Einstiegsprozess ist, dass er sich viel zu komplex gestaltet, als dass der einzelne Einstiegswillige ihn ohne Weiteres planen und umsetzen könnte. Um diesen Prozess für Einsteiger leichter zu machen, schlägt Ducheneaut (2005) die Entwicklung von Software vor, die die oben genannten Zusammenhänge zwischen Menschen, Diskussionsbeiträgen, Code etc. so einfach wie möglich darstellt, so dass in jedem einzelnen Fall die Details der Interaktionen im Nachhinein nachvollziehbar sind und ein Einsteiger somit sein Vorgehen sinnvoller planen kann.

Quintela – Der Einstieg in OSS-Projekte

LITERATUR

Brand, A. (2006): Fallstudie horizontales elektronisches Arbeitsnetz/ Open Source-Projekt, <http://www.soz.unifr Frankfurt.de/arbeitslehre/pelm/docs/FALLSTUDIE%20Open%20Source%20V1.pdf> [Juli 2006]

Döring, N. (2003): Sozialpsychologie des Internet. Hogrefe-Verlag GmbH & Co, Göttingen.

Kamenzki, N. (2005): Einfluss Gewinnen, http://www.inf.fu-berlin.de/inst/ag-se/teaching/S-Komponenten-2005/KamenzkyN05-Einfluss_Gewinnen.pdf [Juli 2006], Freie Universität, Berlin.

von Krogh, G., Spaeth, S. und Lakhani K. (2003): Community, Joining, and Specialization in Open Source Software Innovation: A Case Study. Research Policy, Bd. 32, Nr. 7, S. 1217– 1241.

Latour, B. (1987a): The Pasteurization of French Society, with Irreductions. Cambridge, MA: Harvard University Press.

Maas, W. (2004): Inside an Open Source Software Community: Empirical Analysis on Individual and Group Level. In Proceedings of the 4th Workshop on Open Source Software Engineering. Edinburgh Scotland, S. 64–70.

Mahendran, D. (2002): Serpents and Primitives: An Ethnographic Excursion into an Open Source Community. Unpublished Masters thesis, University of California, Berkeley, Berkeley, CA.

Python (2006): The Python Project's Web Site, <http://www.python.org> [Juli 2006].