

Dissertation

„Free Software, Free Society“?

**Über die Reproduktion von Differenz in der Praxis von Free/Libre Open
Source Software-Communities**

vorgelegt von

Silke Meyer

zur Erlangung des akademischen Grades

Doktorin der Philosophie

2011

Freie Universität Berlin

Fachbereich Politik- und Sozialwissenschaften

Otto-Suhr-Institut für Politikwissenschaft

Gutachten

Erstgutachten: Prof. Dr. Wolf-Dieter Narr

Zweitgutachten: Prof. Dr. Sabine Hark

Datum der letzten mündlichen Prüfung: 18.06.2012



„Free Software, Free Society“? Über die Reproduktion von Differenz in der Praxis von Free/Libre Open Source Software-Communities von Silke Meyer ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 3.0 Deutschland Lizenz (<http://creativecommons.org/licenses/by-sa/3.0/de/>). Bitte beachten Sie, dass dies nicht unbedingt für die zitierten Werke Anderer gilt.



Abb. 0.1.: T-Shirt der *Free Software Foundation Europe*

Danke!

Ich versichere, die vorliegende Arbeit auf der Grundlage der angegebenen Quellen und Hilfsmittel selbstständig verfasst zu haben. Dennoch ist die Arbeit im Austausch mit vielen anderen entstanden. Dafür danke ich besonders

- Prof. Dr. Wolf-Dieter Narr und Prof. Dr. Sabine Hark für die Betreuung, die Diskussionen und für das Vertrauen, dass diese Arbeit es wert ist, geschrieben zu werden,
- der Rosa-Luxemburg-Stiftung für die 3-jährige finanzielle Unterstützung,
- allen Mitgliedern von FLOSS-Communities, die sich von mir haben beobachten und interviewen lassen,
- allen, die bei Colloquien und Methodenworkshops an der Interpretation meines Materials mitgewirkt haben,
- der „Dissen“-Gruppe bei *reflect! Assoziation für politische Bildung und Gesellschaftsforschung*. V: Fabian Georgi, Nina Knirsch, Dag Schölper, Martina Benz und Christian Schröder, für den steten Austausch und das Feedback zu Teilen der Arbeit, Christian Schröder für die geduldige Lektüre von Entwürfen und die Diskussionen darüber,
- Christoph Lang für die Beratung bei der Methodenwahl und die teilnehmende Beobachtung bei einer Keysigning-Party,
- allen, die mir das Hintergrundwissen zu Linux beigebracht haben, besonders Hendrik Naumann und Jürgen Neumann,
- Neelke Wagner für das kritische Feedback zur Gesamtarbeit in der Abschlussphase, Ingo Rau und Katrin Lang für die Hilfe bei den Rechtschreibkorrekturen,
- allen, die mich durch Pausengespräche inspiriert, mich unterstützt und ermutigt haben, besonders Jürgen Neumann, Kristina Dietz, Niels Spilker, Petra Rostock, Pia Garske, meiner WG und meinen Eltern.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Wer gestaltet Computertechnologien?	1
1.2. Vorüberlegungen zur Ethnografie	9
1.3. Begriffsklärungen	13
1.4. Aufbau der Arbeit	15
2. Einführung ins Forschungsfeld FLOSS-Communities	17
2.1. FLOSS-Communities als Teile einer Szene	17
2.1.1. Zum sozioökonomischen Hintergrund von FLOSS-Entwickler_innen	19
2.1.2. Die Rolle von Privatwirtschaft und Stiftungen in der Szene	23
2.1.3. Kooperation und interne Strukturen von FLOSS-Projekten	25
2.1.4. Anliegen und ‚Philosophie‘ von FLOSS-Communities	27
2.1.5. Die Verbindung von Anerkennung und Geschlecht	29
2.1.6. Lokale User Groups als Anlaufstellen	34
2.1.7. Community-Events als Vernetzungstreffen der Szene	36
2.2. Der historische Wissensbestand von FLOSS-Communities	40
2.2.1. Die Entstehung des Internet	40
2.2.2. Die Entstehung des Softwaremarktes	41
2.2.3. Kämpfe um die „Freiheit“ von Software	44
2.2.4. Kurze Reflexion	46
3. Theoretischer Zugang: Einführung in Bourdieus Praxistheorie	49
3.1. Eckpunkte für die Konzeptionalisierung des Forschungsfeldes	52
3.1.1. Bourdieus Feldbegriff	52
3.1.2. Die Reproduktion von Feldern über die Kapitalverteilung	54
3.2. Anhaltspunkte für die Untersuchung von Praxis	58
3.2.1. Die Strukturierung der Praxis durch den Habitus	59
3.2.2. Distinktion: Die Herstellung von Unterschieden über Stil	61
3.3. Theoretisch informierte Reformulierung der Forschungsfragen	63

4. Eine Sprache sprechen	65
4.1. Mitreden und mitlachen können	66
4.2. „Das ‚s‘ steht für Schäuble“ — Humoristisches Sprechen über Politik	73
4.3. Exkurs: Linux als politisches Projekt?	74
4.4. Vergeschlechtlichte Stereotype und sexistische Witze	84
4.5. Empirische Zwischenergebnisse	92
4.6. Theoretische Reflexion: Fachsprache und Humor als Voraussetzung für Teilhabe	93
5. Informelles Lernen in Linux User Groups	95
5.1. Annäherung an LUG-Treffen	97
5.2. Durch technischen Support „der Community etwas zurückgeben“	99
5.2.1. Hilfesuche als Herausforderung im informellen Kontext	99
5.2.2. Vorwissen und Annahmen über Vorwissen	106
5.2.3. Learning by doing or learning by watching?	110
5.2.4. Technischer Support als Selbstinszenierung vor Publikum	117
5.2.5. Maßstäbe für Wissen im informellen Lernkontext	124
5.2.6. Ausprobieren: Zentrale Lernpraxis und Problemlösungsstrategie	129
5.3. Empirische Zwischenergebnisse	134
5.4. Theoretische Reflexion: Zur sozialen Selektivität informeller Lernkontexte	139
6. Techniknutzung und -anpassung	147
6.1. Die Konstitution von technischer Expertise über textbasiertes Arbeiten	147
6.1.1. Distinktion gegenüber Nutzer_innen grafischer Programme	150
6.1.2. Spezialwissen auf die Spitze treiben: Der Texteditor <i>vim</i>	154
6.2. Ästhetik und Kreativität	163
6.2.1. Kreative Umnutzung von Hard- und Software	164
6.2.2. Einschreibungen ästhetischer Vorstellungen in Artefakte	171
6.3. Die Strukturierung der Zeit bei LUG-Treffen	179
6.3.1. „Dreieinhalb Stunden. Wahnsinn.“ — Zeitgefühle	179
6.3.2. Neugierde auf Neues: Spontane Erweiterungen des Lerngegenstandes	187
6.4. Empirische Zwischenergebnisse	189
6.5. Theoretische Reflexion: Technikverständnisse in FLOSS-Communities	191
7. Vertrauen und Sinn für Datenschutz	205
7.1. Technische Sicherheit in der Praxis	205

7.1.1. Umgangsweisen mit Passwörtern	206
7.1.2. „Ich bin root, ich darf das!“ — Zugriffsrechte	210
7.2. Die Formalisierung von Vertrauen beim Keysigning	212
7.2.1. Exkurs: Kämpfe um den Zugang zu Verschlüsselungstechnologien . .	217
7.2.2. Keysigning-Parties	223
7.2.3. Das Web of Trust	230
7.2.4. Keysigning-Parties als ordnende und hierarchisierende Praxis	237
7.3. Empirische Zwischenergebnisse	246
7.4. Theoretische Reflexion: Distinktion über formalisierendes Denken	249
8. Forschungsergebnisse und -perspektiven	255
8.1. Habitus und Kapitalformen in FLOSS-Communities	256
8.1.1. Konstitutive Praktiken von FLOSS-Communities	256
8.1.2. Vom homogenen Habitus zu Habitusfragmenten von Linux-Nutzer_innen	256
8.1.3. Die gegenseitige Konstitution von Kapital und technischen Artefakten in der Praxis	261
8.2. Zur Herstellung von Differenz in FLOSS-Communities	264
8.2.1. Wissen und Vorbildung	264
8.2.2. Geschlecht	265
8.2.3. Stilistische Distinktion über den Umgang mit technischen Artefakten	267
8.3. Fazit und Ausblick	270
Literaturverzeichnis	275
Materialverzeichnis	315
Bildnachweis	317
A. Glossar	319
B. Forschungsmethoden	327
B.1. Datenerhebung	327
B.2. Datenauswertung	331

1. Einleitung

1.1. Wer gestaltet Computertechnologien?

Meine erste Begegnung mit freier Software war die Installation des *Netscape*-Browsers, von dem ich etwa im Jahr 2000 gehört hatte. Er sei kostenlos, besser und aktueller als der *Microsoft Internet Explorer*. Ich nutzte ihn ohne das Wissen darum, was hinter der Idee freier Software steht, und betrachtete ihn als kostenloses Produkt. Von Linux und der Idee, der ‚Philosophie‘ freier Software, erfuhr ich etwa ein Jahr später von einem Freund. Er erzählte, dass man *Microsoft Windows* nicht nutzen müsse, dass es ein alternatives Betriebssystem gebe, an dem alle mitentwickeln könnten, das frei verfügbar und damit *auch* kostenlos sei. Man müsse allerdings „Ahnung haben“, um Linux zu benutzen. Abgesehen davon, dass ich bis zu dem Zeitpunkt noch nicht gewusst hatte, was ein Betriebssystem ist, gefiel mir die Idee aus politischen Gründen — kollektive Softwareentwicklung jenseits von einem großen Konzern? 2003 stand für mich fest, dass ich Linux aus diesen Gründen nutzen wollte. Mit der Installation von Linux und zwei Büchern begann für mich eine intensive Beschäftigung mit Computern, von denen ich zuvor nicht wusste, dass ich dafür Interesse aufbringen könnte. Ich verbrachte Tage, Wochen damit, Software auszuprobieren, zu vergleichen, zu verwerfen, meine Installation „kaputt zu spielen“ und neu aufzusetzen, im Internet zu lesen und Bekannte, die Linux nutzten, auszufragen. 2004 besuchte ich eine erste Veranstaltung der Szene, die der Gegenstand dieser Arbeit ist. Sie war für mich befremdlich: Ich konnte mit den Ausstellerständen nichts anfangen, den Gesprächen nicht folgen und nicht ins Gespräch kommen. Das auf Seite iv abgebildete T-Shirt mit der Aufschrift *Free Software, Free Society*, das mir dort begegnete, warf in mir die Frage auf, wie das gemeint ist: Soll freie Software zu einer befreiten Gesellschaft führen und glauben die Szenegänger_innen daran, die das T-Shirt tragen?

Auch nach diesem ersten Kontakt zur Szene blieb meine Überzeugung von dem Ansatz: Linux wurde und wird als Gegenentwurf zu so genannter „proprietärer“ Software entwickelt. Freie Software erhebt den Anspruch, transparent zu sein: Sie wird mit ihrem

→ *Quellcode* veröffentlicht. Das bedeutet, dass jede_r¹ die Konstruktionsweise der Software einsehen kann. Darüber hinaus ist sie kostenlos und darf (bzw. soll explizit) kopiert, weitergegeben und verbessert werden. Es wurden eigene Lizenzen (wie die → *GNU General Public License*) geschaffen, um diese Eigenschaften der Software auch für geänderte Versionen zu erhalten: Wer freie Software verändert und wieder veröffentlicht, muss dies unter den gleichen Lizenzbedingungen tun, so dass auch → *Derivate* nicht vereinbart werden können (vgl. zu freien Lizenzmodellen Grassmuck 2004: 275 ff.). Freie Software wird nicht von einem Konzern entwickelt, sondern meistens von selbstorganisierten „Communities“, die mein Untersuchungsgegenstand sind. Ich bezeichne sie im Folgenden als FLOSS-Communities. FLOSS steht für *Free/Libre Open Source Software*.² „Community“ ist eine Selbstbezeichnung der Gruppen, die sich für die Entwicklung und Verbreitung freier Software einsetzen und die ich in Kapitel 2 genauer bestimme. Diese FLOSS-Communities, ehrenamtliche Einzelpersonen und Softwareentwickler_innen, die von Firmen dafür bezahlt werden, entwickeln gemeinsam freie Software. Die Kooperation erfolgt in großem Maße, aber nicht ausschließlich über das Internet. Dort steht auch die sehr große und ausdifferenzierte Auswahl an Software, Dokumentation und Supportforen bereit. Der Begründer der ‚Philosophie‘ freier Software, Richard Stallman, hat das Teilen von Quellcode stark politisch-moralisch aufgeladen: Es sei ein fundamentaler Akt von Freundschaft zwischen Programmierer_innen. Die Vermarktung von proprietärer Software verbiete es, andere als Freund_innen zu behandeln (vgl. Stallman 1985). Ein zentrales Anliegen von FLOSS-Communities ist demgegenüber die offene Weitergabe von Wissen: Computerprogramme werden als öffentliche Güter angesehen und sollen gemeinwohlorientiert entwickelt werden. Die Communities verstehen sich als offen für alle, die mitarbeiten möchten — ein Anspruch, an dem ich sie messe.

In den letzten zehn bis fünfzehn Jahren haben rege wissenschaftliche und politische Diskussionen über diese Produktionsweise stattgefunden. Forscher_innen haben sich vor allem dafür interessiert, was Menschen motiviert, unbezahlt zu arbeiten (vgl. z. B. Ghosh et al. 2002; Krishnamurthy 2006; Lin 2006; Bonaccorsi/Rossi 2006; Michlmayr 2007). Sie haben gefragt, wie sich FLOSS-Projekte intern organisieren (siehe etwa Crowston/Howison 2006; Brand/Schmid 2006; O’Neil 2009) und ob diese Produktionsweise auf andere gesell-

¹Der Unterstrich, die so genannte *Gender Gap* ist ein Versuch, eine geschlechtergerechtere Sprache zu verwenden. Er symbolisiert einen Raum zwischen der männlichen und der weiblichen Form und soll Raum für unterschiedliche Selbstverortungen in Bezug auf Geschlecht lassen. Der Vorschlag geht auf Steffen Kitty Herrmann zurück (<http://arranca.org/ausgabe/28/performing-the-gap>).

²Diese Bezeichnung wird in der Literatur häufig verwendet, um verschiedene Strömungen unter einem Begriff zu fassen. Diese Strömungen werde ich später erläutern.

schaftliche Bereiche übertragbar ist (vgl. *Oekonux*³, Siefkes 2007; kritisch Nuss/Heinrich 2002). Die Diskussion mit der größten Reichweite fand über den Umgang mit geistigem Eigentum digitaler Güter statt, in der es *auch*, aber nicht in erster Linie um freie Software ging (vgl. Nuss 2006: 89 ff.; Perelman 2002; Drahos/Braithwaite 2002; May 2000). Die bisherige FLOSS-Forschung bezieht sich vor allem auf die *Entwicklung* freier Software, wobei Entwicklung mit Programmieren und programmiernahen Tätigkeiten wie Tests und Fehlerbehebung gleichgesetzt wird.

Ich möchte zur Entwicklung meiner Fragestellung drei Forschungsergebnisse hervorheben, die mit meinem Interesse an möglichen politischen Anliegen von FLOSS-Communities verbunden sind: Sabine Nuss diskutiert, ob die alternative Produktionsweise freier Software antikapitalistisch ist oder ob sie wegen des offenen Umgangs mit geistigem Eigentum subversives Potenzial hat, und verneint dies: Die prekarierte und oft unbezahlte Arbeit der Entwickler_innen passe gerade in die neoliberalen, marktorientierten Veränderungen von Arbeitsbedingungen. Freie Software sei durch das ausdifferenzierte Lizenzrecht nach einem bürgerlichem Eigentumskonzept verregelt. Zudem sähen die meisten Beteiligten ihre Tätigkeit nicht als politisch an, verfolgten also keine subversiven Absichten (vgl. Nuss 2006). Auch das soziale Miteinander in freien Softwareprojekten wurde kritisch hinterfragt. Patricia Jung (2006) benennt sexistische Diskriminierung in der männlich dominierten „Hackerszene“ als Problem und Dawn Nafus et al. (2006) beleuchten die geschlechtlich segregierte Arbeitsteilung in FLOSS-Projekten, innerhalb derer Frauen häufig die wenig prestigeträchtigen Aufgaben übernehmen.⁴

Gleichzeitig verfügen viele Mitglieder von FLOSS-Communities über ein Fachwissen, das gesamtgesellschaftlich und politisch relevant ist. Mit diesem Wissen können sie die aktuellen politischen und sozialen Auseinandersetzungen um die Gestaltung von Informations- und Kommunikationstechnologien mitführen. In diesem erst entstehenden Politikfeld geht es darum, das zukünftige Internet zu gestalten, es geht um Deutungs- und die Datenhoheit in virtuellen Räumen, darum, Nutzungsmöglichkeiten und -freiheiten auszuhandeln. Auch wenn sich — wie Nuss schreibt — große Teile der Communities nicht als politisch verstehen, gibt es FLOSS-Communities, die ihr Fachwissen für die Beeinflussung dieser Diskussionen oder für politische Aktionen nutzen, wie ein Blick auf Netzpolitik verdeutlichen kann. Die Diskussionen über Informations- und Kommunikationstechnologien stehen im Spannungsfeld von staatlicher Regulierung und

³<http://oekonux.de/> und <http://keimform.de/>, Abruf: 10.11.2011

⁴Auf diese Arbeiten werde ich im Laufe der Arbeit noch genauer zurückkommen.

Überwachung, Bürgerrechten, von der Vernetzung und Mobilisierung politischer Bewegungen und von kommerziellen Interessen. Staatliche Ermittlungsbehörden ermitteln gezielt und verdeckt in sozialen Netzwerken wie *Twitter* oder *Facebook*⁵. Die Vernetzung politischer Bewegungen über diese Portale wird unter mehr oder weniger klaren Umständen beschnitten.⁶ Die wohl aufsehenerregendsten Beispiele sind die politischen Umstürze in Ägypten und Libyen, deren Regierungen die Länder vorübergehend ganz vom Internet abschnitten, um die Vernetzung Protestierender und die Veröffentlichung von Nachrichten zu behindern.⁷ Der Skandal um die von Regierungen eingesetzten „Trojaner“ zum Ausspionieren privater Computer wirft nicht nur die Frage auf, was staatliche Behörden grundsätzlich dürfen sollten, sondern auch, wie solche Skandale aufgedeckt werden können. Dazu braucht eine Gesellschaft technisch versierte und kritische Instanzen, die feststellen können, was die eingesetzten Technologien können.⁸ Hier treffen technische Machbarkeit, staatliche Überwachungsversuche, deren verfassungsrechtliche Einschränkungen und die Überprüfung ihrer Einhaltung aufeinander. Ähnliches gilt für das Internet als kommerziellen Raum. Wie die kommerziellen Diensteanbieter mit den hochgeladenen Datenprofilen umgehen, ist nur in Teilen transparent für die Nutzer_innen. Einer der zentralen Punkte dieser Diskussionen um die Gestaltung des Internet ist seine Topologie: Sie tendiert zu starker Zentralisierung bei wenigen Diensteanbietern und ist damit relativ leicht zu vereinnahmen.

In der netzpolitischen Gemengelage kommen verschiedene technikbezogene Kompetenzen ins Spiel, die die Akteur_innen benötigen. Die Vorschläge von Politiker_innen zeigen oft, dass sie selbst sehr wenig von den Technologien verstehen, um die es geht. Dass alle etablierten Parteien diesen Themenbereich so lange vernachlässigt haben, war eine Bedingung für die Entstehung und die Wahlerfolge der Piratenpartei. Sie hat das The-

⁵vgl. die Kleine Anfrage von Ulla Jelpke (MdB, DIE LINKE) (http://www.ulla-jelpke.de/uploads/KA_17_6100_facebook.pdf, Abruf: 10.11.2011)

⁶Ein Beispiel aus dem deutschen Kontext ist die *Facebook*-Seite der Anti-Castor-Bewegung „Castor schottern“, deren Betreiber_innen sich kurz vor dem Castor-Transport nicht mehr einloggen konnten und bereits veröffentlichte Beiträge vermissten (die tageszeitung, 19.10.2010, <https://www.taz.de/!60003/>, Abruf: 10.11.2011). *Facebook* schob dies auf technische Probleme (die tageszeitung, 20.10.2010, <https://www.taz.de/!60054/>, Abruf: 10.11.2011).

⁷Zum Thema Internetzensur geben die *Reporter ohne Grenzen* jährliche Bericht heraus (vgl. <http://www.reporter-ohne-grenzen.de/publikationen/themenberichte/>).

⁸Im Herbst 2011 nahm der *Chaos Computer Club (CCC)* die Rolle einer solchen Instanz ein: Ihm wurde von einem Anwalt eine Festplatte mit einer Überwachungssoftware übergeben, die in Ermittlungsverfahren eingesetzt wird. Die Innenministerien von Bund und einigen Ländern beriefen sich auf das Urteil des Bundesverfassungsgerichts, das die Überwachung grundsätzlich erlaubte, den Umfang jedoch definierte. Der CCC wies nach, wie weit die eingesetzte Software darüber hinausging.

ma auf die politische Agenda gesetzt, für das internet- und technikaffine Wähler_innen den anderen Parteien die Kompetenz absprach und teilweise immer noch abspricht (vgl. Niedermayer 2010: 843 ff., 851). Die damalige Bundesfamilienministerin Ursula von der Leyen (CDU) schlug 2009 z. B. das so genannte Zugangserschwerungsgesetz vor, das Websites mit Kinderpornografie vonseiten der Provider blockieren sollte. Es kam nicht nur zu harscher grund- und datenschutzrechtlicher Kritik, die der Ministerin den Spitznamen „Zensursula“ einbrachte, sondern die vorgeschlagene Maßnahme ist technisch unzureichend und daher sehr einfach zu umgehen.⁹ Das Gesetz blieb nur kurz in Kraft. Ein anderes Beispiel ist ein Vorschlag der Verbraucherschutzministerin Ilse Aigner (CSU): Sie brachte 2010/11 einen so genannten „digitalen Radiergummi“ auf die Agenda, der online veröffentlichten Daten ein digitales Verfallsdatum einschreiben sollte. So sollte den Nutzer_innen ein „Recht auf Vergessen“ eingeräumt werden. Die so genannte „Internetgemeinde“ quittiert derartige Regulierungsvorschläge stets mit großem Spott, nicht nur, weil sie die staatliche Regulierung per se ablehnt, sondern auch, weil die Vorschläge aus technischer Perspektive sinnlos sind. Es bedarf demnach auch einer gewissen technischen Kompetenz, um sich zu netzpolitischen Fragen äußern zu können.

Im Zusammenhang mit der zentralisierten Netzwerktopologie und der Kompetenzvermittlung kommt die freie Software wieder ins Spiel, die unter genau diesem Aspekt eine kritische Gegenbewegung darstellt: Die Communities setzen sich gegen proprietäre Software und Formate und für die Wiederverwendung und Umnutzung von Wissen und Technologien ein. Die zur Verfügung stehende freie Software kann verwendet werden, um eigene Kommunikationsinfrastrukturen aufzubauen. Die existierenden Alternativen zu den verbreiteten großen und bekannten Diensten sind oft in Form von freier Software realisiert worden. Der Ansatz, der dabei verfolgt wird, sind dezentrale Lösungen: Ein Beispiel ist Software für dezentrale soziale Netzwerke, die Menschen auf ihren eigenen Servern aufbauen können, die aber trotzdem mit den anderen Servern, die die gleiche Software haben, kommuniziert. Dabei sollen Nutzer_innen die Hoheit über ihre Daten behalten können. Zudem sind dezentrale Softwarelösungen weniger angreifbar. Aus dieser Perspektive positionieren sich die Softwareprojekte, die diese Software entwickeln, sowohl gegen staatliche Überwachung als auch gegen die intransparente kommerzielle Nutzung von Daten.

Außer Softwareentwicklung ist thematisch fokussierter Netzaktivismus eine Aktionsform,

⁹vgl. <http://www.heise.de/newsticker/meldung/Internetwirtschaft-Sperrung-von-Kinderporno-Seiten-nutzlos-209424.html>, Abruf: 13.12.2011

in die auch Mitglieder von FLOSS-Communities involviert sind. Projekte wie *Telecomix*¹⁰ oder *freifunk*¹¹ machen sich die Möglichkeit zunutze, eigene Infrastrukturen aufzubauen, um alternative Vernetzungen herzustellen. Ein anderes Beispiel für Netzaktivismus ist die Website *hatr.org*, die sich (queer-)feministische Blogger_innen ausgedacht haben: Mit einer Erweiterung für die → *Weblog*-Software *WordPress* sammeln sie Hasskommentare, die sie auf ihre kritische Berichterstattung erhalten, auf einer gesonderten Website. Dort haben sie Werbung geschaltet; die Einnahmen kommen Projekten zugute, über die die Beteiligten gemeinsam entscheiden. Während diese Projekte sich auf kreative, konstruktive Weise der Software bedienen, gibt es im Bereich Netzaktivismus ebenso Sabotageakte. Das jüngst bekannteste Beispiel sind *Anonymous*, die aus einem zunehmend politischen Protest heraus Websites vorübergehend unerreichbar machen.¹² Für all diese Formen von politischer Einmischung ist also erstens ein entsprechendes technisches Wissen erforderlich, zweitens spielen die technischen Artefakte selbst eine Schlüsselrolle in den Interventionen. In der vorliegenden Arbeit betrachte ich die Communities und ihre Mitglieder als *potenzielle* Akteur_innen in diesen aktuellen politischen und sozialen Auseinandersetzungen um Technikgestaltung.

FLOSS-Communities haben den Anspruch, das Wissen über Technologien frei miteinander zu teilen, was in den Lizenzen zum Ausdruck kommt, unter die sie ihre Arbeiten stellen. Dies ist ein emanzipatorischer Anspruch: Menschen sollen in die Lage versetzt werden, Software selbst mitzugestalten, statt sie als gegeben hinzunehmen. Der Wissensvermittlung in FLOSS-Communities kommt deshalb eine Rolle in den Auseinandersetzungen um Technologiegestaltung zu, die meines Erachtens bisher vernachlässigt wurde. In der vorliegenden Arbeit interessiert mich, wie in den Communities das Wissen vermittelt wird, das auch Hintergrund netzpolitischer Interventionen ist (oder werden kann). Was bedeutet dieser Anspruch in der *Praxis* der Gruppen? Werden dort tatsächlich Lernen und Wissenstransfer zwischen Individuen aus ganz verschiedenen sozialen Kontexten erleich-

¹⁰Das Projekt *Telecomix* hat mehreren hundert Ägypter_innen Internetzugang verschafft, als das Land 2011 vom Internet abgeschnitten wurde. Die Aktivist_innen nahmen ihre alten Modems wieder in Betrieb, schickten die anrufbaren Telefonnummern per Fax nach Ägypten und leiteten die Verbindungen, die aus Ägypten kamen, ins Internet weiter. Erst als wenig später Libyen abgeschnitten war, wurden diese Telefonnummern für die Libyer_innen gesperrt.

¹¹Die Initiative *freifunk* baut unkommerzielle WLAN-Netze auf, die ans Internet angebunden sein können, aber auch als eigene Netzwerke betrieben werden können.

¹²Hinter dem Decknamen *Anonymous* verbergen sich Unbekannte, die sich seit 2008 politisch äußern, oft im Dienste der Redefreiheit. Sie wurden 2010 sehr bekannt, als unter diesem Namen → *Server* zum Zusammenbruch gebracht wurden. Dies waren zunächst Server von Banken, die der Whistleblower-Plattform *Wikileaks* den Zugang zu ihren Konten gesperrt hatten (vgl. zu *Anonymous* Coleman 2011).

tert, wie Huysman/Lin (2005: 56) schreiben? Da die bisherige FLOSS-Forschung sich fast ausschließlich auf die Softwareentwicklung konzentriert, untersuche ich mit Linux User Groups (LUGs) einen anderen Ausschnitt des weiten Forschungsfeldes. LUGs sind mehr oder weniger fest institutionalisierte lokale Gruppen, in denen sich Linux-Nutzer_innen gegenseitig bei technischen Problemen helfen, das eigene Wissen in Kurzvorträgen für andere aufbereiten, kurz: LUGs sind Plattformen für Wissenstransfer. Sie basieren in der Regel auf Begegnungen von Angesicht zu Angesicht, wenn es auch oft parallel dazu eine elektronische Kommunikationsinfrastruktur gibt. Eine Prämisse meiner Arbeit ist – beruhend auf eigener Erfahrung –, dass die Beteiligten sich das Wissen *praktisch* aneignen, „hands on“, beim Machen. Aus diesem Grund habe ich mich für eine ethnografische Untersuchung der Praxis von Linux User Groups entschieden.

Die oben angeführten kritischen Forschungsergebnisse werfen die Frage nach Differenzen und Hierarchiebildung in LUGs auf: Wie werden dort Unterschiede oder sogar Ausschlüsse hergestellt? Was zählt in der Praxis? Wie wirken sich soziale Ungleichheiten und gesellschaftliche Herrschaftsverhältnisse in den Interaktionen aus? Ich gehe davon aus, dass in der Praxis gleichzeitig technisches Wissen vermittelt und soziale Positionen ausgehandelt werden. Mich interessiert, *wie* dies geschieht und welche Rolle die allgegenwärtigen technischen Artefakte in solchen sozialen Differenzierungsprozessen spielen. Da ich Wissensaneignung als Voraussetzung dafür betrachte, dass Individuen oder Gruppen zur Technologiegestaltung befähigt werden, steckt in der Frage nach sceneinterner Differenzierung auch eine Machtfrage: Es geht, so meine Einordnung des Themas, um Kompetenzerwerb in einem der wichtigsten Technologiebereiche, die unsere Gesellschaft derzeit prägen. Von Informations- und Kommunikationstechnologien wie Computer und Internet hängen große Teile unseres Alltags ab — und damit auch von Personen, die in der Lage sind, diese Technologien zu bedienen, zu konfigurieren, zu reparieren. Der offene Umgang mit dem entsprechenden Wissen ist deshalb so wichtig, um sich dieser Artefakte bemächtigen zu können. Ich habe die Praxis im Umgang mit technischem Wissen als Gegenstand dieser Arbeit gewählt, weil ich davon ausgehe, dass sich in ihr gesellschaftliche Machtverhältnisse niederschlagen. Sie zu erkennen und zu reflektieren, ist eine Voraussetzung für Veränderung, für einen selbstbestimmten Umgang mit den Maschinen, die wir unser Leben organisieren lassen.

Die Arbeit basiert demnach auf einigen Vorannahmen und Hypothesen:

1. Demokratische Technikgestaltung ist nur mit einem gewissen Vorwissen möglich.

FLOSS-Communities haben den Anspruch, mehr Menschen das nötige Wissen zu vermitteln und könnten so einen selbstbestimmteren Umgang mit Informations- und Kommunikationstechnologien vermitteln.

2. FLOSS-Communities konstituieren sich über gemeinsame Praktiken. Dabei bilden sie einen Habitus im Umgang miteinander und mit technischen Artefakten heraus.
3. In der Praxis der Communities wird mehr als nur „reines“, technisches Wissen vermittelt. In denselben Prozessen werden auch soziale Beziehungen ausgehandelt. In diesen sozialen Beziehungen geht es um Anerkennung, die darüber hergestellt wird, wer die Kontrolle und die Deutungsmacht über die Universalmaschine Computer hat.
4. Der Umgang mit technischen Artefakten ist ein wesentlicher, nicht wegzudenkender Bestandteil der Praxis und verdient gesonderte Aufmerksamkeit.

Dieser Komplex wird mit folgenden Forschungsfragen untersucht.

1. Mittels welcher Praktiken konstituieren sich Linux User Groups?
2. Welches implizite, praktische Wissen und welches technische Wissen werden in diesen Praktiken mobilisiert und vermittelt?
3. Wie sind technische Artefakte in diese Praktiken involviert?
4. Wie und entlang welcher Kategorien wird in diesen Praktiken Differenz hergestellt?

Mein übergeordnetes Erkenntnisinteresse gilt dabei der Frage, wie der Anspruch von FLOSS-Communities, Menschen zur Technologiegestaltung zu befähigen, in der Praxis umgesetzt wird. Die Frage nach Differenzkategorien impliziert immer auch eine Frage nach Machtverhältnissen. Die vorliegende Arbeit hat jedoch nicht das Ziel, eine theoretisch unterfütterte, systematische Untersuchung von Macht- oder Herrschaftsverhältnissen zu leisten. Mit der Analyse von Praxis möchte ich vielmehr eine empirische Grundlage dafür schaffen, dass diese Verbindung an anderer Stelle hergestellt und begrifflich präzisiert werden kann.

1.2. Vorüberlegungen zur Ethnografie

FLOSS-Communities werden in der vorliegenden Arbeit als Feld im Bourdieuschen Sinne begriffen, das von spezifischen, in der Auswertung aufzuzeigenden Habitus, von Kapital- und Praxisformen strukturiert wird (vgl. zur theoretischen Fokussierung Kapitel 3). Der feldspezifische Habitus und die dort relevanten Kapitalformen werden analysiert und herangezogen, um die Herstellung von Ausschluss, Differenz und Distinktion in FLOSS-Communities zu erklären. Detaillierte Ausführungen zu den eingesetzten Forschungsmethoden befinden sich im Anhang dieser Arbeit.

Die Praktiken der FLOSS-Communities werden ethnografisch rekonstruiert. Die Ethnografie erscheint mir aus folgenden Gründen dem Gegenstand angemessen: Erstens ist es das Anliegen einer Ethnografie, die Funktionsweise und die Relevanzen eines Forschungsfeldes von innen heraus zu verstehen, statt ein vorab erstelltes, wissenschaftliches Kategorienraster an das Feld heranzutragen. Ethnografie ist keine vorgefertigte Methode, sondern eine Forschungshaltung oder eine Rahmenstrategie (vgl. Amann/Hirschauer 1997; Krotz 2005). Die Studie wird feldspezifisch und situationssensibel angelegt, um dem Forschungsfeld auch als Außenstehende_r möglichst gerecht zu werden. Sie hat das Anliegen, die Praktiken und Logiken des Feldes durch extensive Feldaufenthalte (teilnehmende Beobachtung, Gespräche und Interviews) zu verstehen (vgl. Krotz 2005: 258 f.). In den Datenkorpus wird alles einbezogen, was erhoben werden kann. In dieser Arbeit werden Daten aus teilnehmender Beobachtung und Interviews, ebenso wie aus protokollierten informellen Gesprächen verwendet, dazu stellenweise Bild- und Textmaterial sowie Internetseiten. In diesem Sinne nutze ich die angeführte Literatur vorab zur Sensibilisierung bzw. Fokussierung für die Datenerhebung, nicht aber zur Vorab-Strukturierung des Gegenstandes entlang von theoretisch entwickelten Analysekategorien: Bourdieu lenkt zwar das Augenmerk auf Fragen nach Kapitalformen, Habitus und Differenz bzw. Distinktion, lässt jedoch offen, wie das Feld strukturiert ist, statt schon bestimmte Differenzkategorien oder Distinktionslinien hervorzuheben. Die in FLOSS-Communities relevanten Kategorien werden aus dem Datenmaterial herausgearbeitet, dann mit Theorien konfrontiert, was wiederum zur Entwicklung oder Ausdifferenzierung dieser Theorien führt.¹³

Zweitens arbeiten Ethnograf_innen nicht mit Typologien, um nicht über Gebühr zu homogenisieren. Die Komplexität von FLOSS-Communities soll entfaltet werden (vgl.

¹³Für diese Einordnung danke ich Anja Tervooren (Magdeburger Methodenworkshop 2009).

Amann/Hirschauer 1997), wobei ihre Heterogenitäten und Widersprüche stehen gelassen werden können. Es wird weder vorrangig auf eine schlüssige Interpretation hingearbeitet (vgl. Pritzlaff 2006: 127 ff.), noch der Anspruch erhoben, die Ergebnisse der Analyse eines notwendig beschränkten Materialkorpus auf „die Communities“ generalisieren zu können. In ethnografischen Arbeiten werden keine Ergebnisse mit Anspruch auf Repräsentativität erarbeitet, sondern es wird höchstens mit Vorsicht verallgemeinert. Drittens werden gesellschaftliche Kontexte nicht außer Acht gelassen.¹⁴

Viertens werden das Verhältnis zwischen Forschenden und Beforschten und die eigene Forscher_innenrolle in ethnografischen Arbeiten explizit thematisiert. Eine Ethnografie zu schreiben, bringt nämlich Rollenkonflikte mit sich: Mit welchem Recht und aus welcher Position spreche ich über FLOSS-Communities? Dass dieser Bedarf in der Literatur so explizit formuliert wird, lässt sich aus der Geschichte der Fachdisziplin erklären: Die Ethnografie kommt aus einer ethnologischen Forschungstradition, die von Kolonialismus, Ethnozentrismus und Positivismus geprägt war.¹⁵ Als Wissenschaftlerin konstruiere ich meinen Forschungsgegenstand ‚die irgendwie anderen‘ über die Abgrenzung gegenüber der Kultur und Gesellschaft, in der ich mich sonst bewege (vgl. Clifford 1993: 113). Die zwischenmenschlichen Beziehungen und persönlichen Erfahrungen, die die Forschung ermöglicht haben, werden oft verschleiert (vgl. Berg/Fuchs 1993b: 64), ohne zu reflektieren, dass Wissenschaftler_innen auch in die Dominanzstrukturen verstrickt sind, die sie untersuchen (vgl. Said 1989: 211).

„Die Verschriftlichung von ethnografischem Material ist immer mit sozialer Macht verbunden. Während der Feldforscher im Feld oft genug seinen Informanten völlig ausgeliefert ist, kann er am Schreibtisch sein Material beliebig gewichten, auslegen und interpretieren“ (Horstmann 1998: 6).

Aus dem Dilemma, dass ich Macht auf meinen Forschungsgegenstand ausübe, den ich eigentlich aus einer machtkritischen Perspektive untersuchen möchte, komme ich nicht heraus. Bourdieu spricht von unterschiedlichen sozialen Voraussetzungen von Forscher_in und Menschen im Forschungsfeld (vgl. Bourdieu 1999: 32): Ich trage mit meinen

¹⁴Darin unterschieden sich Ethnografien z. B. von Forschungsperspektiven, die sich auf den symbolischen Interaktionismus nach Herbert Blumer beziehen. Sie fragen danach, welchen subjektiven Sinn Individuen mit ihren Handlungen und ihrem Umfeld verbinden und verharren damit auf der Betrachtungsebene des Individuums (vgl. Flick 1999: 29 ff.). Für die vorliegende Untersuchung ist jedoch auch die Einbettung der Praktiken in breitere gesellschaftliche Kontexte relevant.

¹⁵An dieser Stelle werden verkürzend einige Punkte herausgegriffen, die meine Forscherinnenposition verdeutlichen sollen. Zur Geschichte der Ethnologie vgl. Berg/Fuchs (1993b) und Clifford (1993).

Forschungsfragen ein Problembewusstsein an die Community heran, indem ich Fragen und Thesen formuliere, von denen ich den Personen im Feld unterstelle, dass sie sie sich nicht stellen. Ich differenziere mich durch „wissenschaftliche“ Praktiken von ihnen, indem ich mir durch eine Anhäufung von Datenmaterial, das es in der Praxis des Feldes in der Form und Fülle nicht gibt, einen „Über-Blick“ (Berg/Fuchs 1993b: 31) verschaffe, den ich den Communities nicht zutraue (vgl. auch Bourdieu 1999: 151). Zudem schreibe ich für eine eher akademische Leser_innenschaft, nicht (in erster Linie) für die Rezeption im Forschungsfeld (vgl. Crapanzano 1986). In den Forschungsprozess fließen bewusste und unbewusste Entscheidungen und Interpretationen meinerseits ein, da bei Erhebungsmethoden wie der teilnehmenden Beobachtung die Datenerhebung und die Auswertung eng verbunden sind: Ein von mir verfasstes Protokoll enthält bereits *vor* der expliziten Auswertung Interpretationen der beobachteten Situation. Diese werden mir oft nur durch den kontrollierenden Blick anderer in Forschungswerkstätten oder Colloquien bewusst und nur dadurch kann ich sie meinen Leser_innen transparent und hinterfragbar machen. Mittels solcher Strategien beanspruche ich eine Definitionsmacht für die Entschlüsselung des Feldes und wissenschaftliche Autorität (vgl. Bourdieu 1999: 151). Dies ist ein aktiver Konstruktionsprozess, bei dem immer wieder die Frage ansteht, ob ich dem Forschungsfeld damit gerecht werde. Welche Strategien kann ich nun ins „Feld“ führen, um es möglichst fair zu betrachten und meine eigenen konstruierenden Eingriffe transparent zu machen?

Erstens gilt es zu explizieren, von welcher Warte aus ich mir das Feld aneigne, welche Geschichten und Politiken, welche normativen Annahmen meine Vertextlichung durchziehen: Mein grundsätzliches Anliegen ist eine Kritik an Macht- und Herrschaftsverhältnissen, die sich *auch* in Praktiken niederschlagen, die vom Anspruch her emanzipatorisch sein wollen. Einen selbstbestimmten Umgang mit Technologien und eine vergesellschaftete Technologiegestaltung sehe ich durch strukturelle soziale Ungleichheiten behindert, durch mehr oder weniger subtil wirkende Herrschaftsverhältnisse, die häufig Menschen entlang der Differenzkategorien *race*, *class* und *gender* klassifizieren und diskriminieren (vgl. etwa Klinger/Knapp 2005; Degele/Winker 2009; Lutz/Wenning 2001: 20). Diese Kritik speist sich aus meiner Auseinandersetzung mit strukturalistischen, postmodernen und feministischen Theorien, die die Strukturierung von Gesellschaft und von Wahrnehmung durch soziale Kategorisierungen und Normen hinterfragen. In Anlehnung an die Konzepte von *Doing Gender* (West/Zimmerman 1987) und *Doing difference* (West/Fenstermaker 1995) gehe ich davon aus, dass Geschlecht und andere Differenzkategorien von situierten

Individuen in der Praxis hergestellt werden, die immer in einen sozialen Kontext eingebettet ist. Diesen starken Fokus auf Differenz und Distinktion trage ich als Frage an mein Forschungsfeld heran. Damit konstruiere ich Differenzen, die im Feld selbst nicht gemacht werden, insoweit, als ich beobachtete Abgrenzungsmechanismen mit Differenzkategorien zusammendenke, statt die feldinterne Erklärung gelten zu lassen, das seien unterschiedliche Individuen, die einfach individuelle Entscheidungen getroffen hätten.

Diese Frage nach der Herstellung von Differenzen ist immer auch mit einer Frage nach Machtverhältnissen verbunden. Ich lege eine Untersuchung von Machtbeziehungen in FLOSS-Communities in dieser Arbeit *nicht* systematisch an, denke Macht jedoch mit Michel Foucault als dezentral entstehende, ständig fließende Kräfteverhältnisse, die allen Beziehungen innewohnen. Sie wirkt nicht repressiv, sondern zielt darauf ab, Leben zu ordnen und zu organisieren (vgl. Foucault 1977: 93 ff.; Foucault 1976a: 114 ff.). Auch Praxis ist damit ein Teil von Machtbeziehungen, die soziale Hierarchien und Dynamiken des Unterscheidens und des Ausschlusses aufrechterhalten.

Zweitens mache ich an dieser Stelle mein Forschungsanliegen deutlich. Hinter der Forschungsarbeit stehen mein persönliches Interesse an freier Software und die grundsätzliche Sympathie für die Idee dahinter. Meines Erachtens *hat* die Selbstorganisation in Communities politische und emanzipatorische Potenziale, die aufgrund bestimmter — im Folgenden zu untersuchender — Eigenheiten konkreter FLOSS-Communities nicht zur vollen Entfaltung kommen. Es ist mir ein Anliegen, meinen feministischen, politischen und sozialwissenschaftlichen Hintergrund im Sinne eines situierten Forschens dort einfließen zu lassen (vgl. Smith 1987) und ein Bewusstsein für problematische Praktiken zu schaffen. Während der Forschung habe ich mich gleichzeitig selbst in die Communities hineinbegeben und mir etliche Praktiken zueigen gemacht, die ich genauso infrage stelle, z. B. das Hantieren mit impliziten Maßstäben für Expertenwissen (vgl. Kapitel 5).

Drittens versuche ich demnach, mir meine Entscheidungen, Interpretationen und Meinungen bewusst zu machen, die einen besonderen Stellenwert dadurch erhalten, dass ich mich als Ethnografin selbst zum Erhebungsinstrument mache. Zudem hat sich im Forschungszeitraum mein Verhältnis zum Feld verändert, von einer Außenstehenden, zum Neuling, zur einer, die sich in den untersuchten Communities als Mitglied fühlt. Mein Forschungsinteresse und mein technologisches Interesse standen zunehmend gleichberechtigt nebeneinander. Ich habe, um in diesem Feld überhaupt eine „Revolution“ machen zu können, viele „Regeln des Spiels“ selbst inkorporiert und damit anerkannt (vgl. Bour-

dieu 1998b: 142). „Dissidenz und Partizipation sind (...) unauflöslich verknüpft: Teilhabe, ja Akzeptanz der herrschenden Spielregeln ist die paradoxe Voraussetzung für Veränderung“ (Hark 2005: 73). Dies wird im Verlauf der Auswertung gekennzeichnet. Für diese ständige Hinterfragung ziehe ich meine wissenschaftlichen Austauschpartner_innen heran. Viertens ist es kein Ziel dieser Arbeit, ein kohärentes Bild von in sich homogenen Communities zu konstruieren: Auch wenn ich nach konstitutiven Praktiken und ‚dem Habitus‘ frage, bleibe ich offen für Inkohärenzen und Widersprüche.

1.3. Begriffsklärungen

In der existierenden Forschung gibt es verschiedene Bezeichnungen, um das Themenfeld von freier und Open Source-Software auf den Begriff zu bringen. Auf den dahinter stehenden Namensstreit zwischen freier Software und Open Source-Software gehe ich weiter unten ein. In dieser Arbeit verwende ich die Abkürzung FLOSS, die für Free/Libre Open Source Software steht, werde also von FLOSS-Communities und FLOSS-Projekten sprechen. Der Zusatz „libre“ soll den Unterschied zu „free“ klar ausdrücken, da „free“ auch als „kostenlos“ verstanden werden kann, ohne dass die ‚Philosophie‘ freier Software gesehen wird. Auch wenn in meinen Erhebungen das Betriebssystem Linux im Vordergrund steht, ziehe ich teilweise Materialbeispiele heran, die sich auf andere freie Software beziehen. Zu Beginn des zweiten Kapitels setze ich mich mit angemessenen Bezeichnungen für das Forschungsfeld genauer auseinander. „Community“ ist eine Selbstbezeichnung im Feld. Ich werde in Anlehnung an Ronald Hitzler et al. (2001) von FLOSS-Communities als Teilen einer Szene sprechen und beide Begriffe synonym verwenden.

Mit dem Begriff Differenz knüpfe ich an intersektionale Forschungsarbeiten an. Unter Differenz verstehe ich Unterschiede zwischen Personen oder Gruppen, die zu Diskriminierung führen. Differenzen sind sozial konstruierte, wirkmächtige Unterschiede, die mit Wertungen versehen und hierarchisiert werden. Unter dem Begriff Strukturkategorien werden meist nur Klasse, ‚Rasse‘, Geschlecht und Begehren gefasst. Diese Kategorien sind mit strukturellen Ungleichheiten verbunden. Differenz kann entlang dieser und vielfältiger weiterer Kategorien hergestellt werden und in den verschiedensten sozialen Kontexten mobilisiert werden (vgl. McCall 2005; Klinger/Knapp 2005; Degele/Winker 2007; Lenz 2010).

Distinktion stammt im Zusammenhang dieser Arbeit von Pierre Bourdieu. Der Soziologe verwendet ihn um zu zeigen, dass sich die sozialen Unterschiede im Bereich des Geschmacks fortsetzen. Mit Distinktion bezeichnet er solche Praktiken, mittels derer Akteur_innen sich wertend auf die Lebensstile anderer beziehen und sich selbst damit im sozialen Raum positionieren (vgl. Bourdieu 1987; Diaz-Bone 2002: 21 ff.).

Vor allem in den empirischen Teilen dieser Arbeit werden die Leser_innen mit der Fachsprache des untersuchten Feldes konfrontiert. Sie durchzieht nicht nur die Kommunikation der beobachteten Personen, sondern auch die Beobachtungsprotokolle, habe ich mich selbst doch soweit ins Feld begeben, dass ich sie mir in Teilen angeeignet habe. Dem ausschließenden Potenzial von Fachsprache und Szenehumor widme ich mich in einem eigenen Kapitel. Das hebt das Problem jedoch nicht auf. Die verwendeten technischen Fachbegriffe sind am Ende der Arbeit im Glossar aufgeführt.¹⁶ Mir ist bewusst, dass auch die Erläuterungen Leser_innen, die wenig oder nichts mit dem Forschungsfeld zu tun haben, nur bedingt helfen können, es ist jedoch ohne Kenntnis der Fachsprache möglich, die Arbeit zu verstehen.

Im Fließtext spreche ich oft von „Nutzer_innen“. Dies meine ich nicht als Gegensatz zu „Entwickler_innen“. Mir ist es wichtig, die häufig wahrgenommene Dichotomie von Nutzung versus Entwicklung zu hinterfragen. Sie wurde nicht nur von der feministischen Technikforschung kritisiert, weil sie technologische Konstrukteur_innen mythisch als geniale Schöpfer_innen darstelle und ihnen passive Rezipient_innen gegenüberstelle (vgl. Suchman 2002: 95; zitiert bei Bath 2009: 183 f.). Sie ist angesichts der wechselnden Rollen aller Personen in den Communities auch empirisch nicht haltbar, wie im Laufe der Arbeit deutlich werden wird. (In der Idee freier Software wird diese Dichotomie zwar aufgehoben, als Distinktion wirkt die Abgrenzung von reinen „Nutzer_innen“ hier jedoch trotzdem.)

Wenn ich von „Community-Mitgliedern“ spreche, gehe ich ebenfalls — wie deutlich werden wird — nicht von einer klaren Mitgliedschaft aus, die von einer klaren Nichtmitgliedschaft unterscheidbar wäre. Die Grenzen des Feldes sind fließend, so dass ich alle bei den Community-Treffen auftauchenden Personen als Mitglieder bezeichne.

¹⁶Die erklärten Begriffe sind im Fließtext durch einen vorangestellten Pfeil gekennzeichnet.

1.4. Aufbau der Arbeit

Zunächst stelle ich FLOSS-Communities als Forschungsfeld vor (Kapitel 2). Neben einer Einführung in die Strukturen wird der historische Wissensbestand der Communities angerissen. Dabei ordne ich FLOSS-Communities soziologisch als Szene ein, analog zu den Jugendszenen, die Hitzler et al. (2001) untersucht haben. Im dritten Kapitel richte ich den Blick auf die Konzepte von Pierre Bourdieu: Ich betrachte FLOSS-Communities als Feld in seinem Sinne und formuliere theoretisch inspirierte Fragen für die Auswertung des Datenmaterials. Im Empirieteil (Kapitel 4–7) untersuche ich vier Komplexe zentraler Praktiken: Die Sprache, den Wissenserwerb und seinen Kontext, den praktischen Umgang mit freier Software und weitere verbreitete Praktiken, die ich als formalisierende Praktiken begreife.

Im vierten Kapitel frage ich danach, wie Fachsprache die Zugehörigkeit zu FLOSS-Communities bestimmt. Humor identifiziere ich als integralen Bestandteil der Fachsprache. Witze über Technik, über Politik und über Frauen bzw. Sexualität haben eines gemeinsam: Sie spielen mit einem sehr technischen Verständnis von Gesellschaft und sozialen Beziehungen. Sie ironisieren den Glauben, alle Probleme seien mit technologischen Mitteln lösbar. Ein Frauenbild, in dem Frauen passive Objekte sexueller Begierde und technisch inkompetent sind, spiegelt sich in der Sprache ständig wider. Technische Fachsprache und ein technikbezogener Humor konstituieren Expertenwissen und gleichzeitig diejenigen, die nicht mitreden und mitlachen können.

Im fünften Kapitel geht es um die Praktiken der Wissensvermittlung in Linux User Groups. LUGs werden als informeller Lernkontext identifiziert, in dem technisches Wissen, aber auch ein „heimlicher Lehrplan“ vermittelt wird. Die in den Communities wahrgenommene Relevanz von technischem Wissen übt verschiedene Wirkungen aus: Alle Beteiligten eignen sich fortlaufend Wissen an, und zwar in den selbst geschaffenen informellen Kontexten, die LUGs bieten. Maßstäbe für Expertise sind ständige Verhandlungssache, sie sind situations- und themenabhängig. Einige setzen ihr Wissen wirkungsvoll zur Selbstinszenierung ein, um sich als Expert_innen und gleichzeitig andere als Neulinge interaktiv herzustellen. Der technische Support kann darüber hinaus als eine praktische Herstellung von Geschlecht gelesen werden. Anerkennung erhalten die Community-Mitglieder hier über eine situative, praktische Problemlösungskompetenz. Diese Kompetenz, die im Zentrum des technischen Supports der LUG steht, ist innerhalb der Gruppen sehr unter-

schiedlich ausgeprägt. Vor dem Hintergrund erziehungswissenschaftlicher Literatur diskutiere ich die soziale Selektivität von informellen Lernkontexten, wie LUGs es sind.

Im sechsten Kapitel arbeite ich Technikverständnisse von Linux-Nutzer_innen heraus. Dazu untersuche ich Praktiken der Techniknutzung und -anpassung anhand von drei Themenkomplexen: Zum einen geht es um den Einsatz textbasierter Programme, die einen hohen Lernaufwand bedeuten, aber auch mit Prestige verbunden sind. Hier wird gezeigt, wie und aufgrund welcher Eigenschaften Software als Distinktionsmittel eingesetzt werden kann. Zweitens rücke ich in Software eingeschriebene ästhetische Vorstellungen in den Blick, die als Skripte bzw. Genderskripte diskutiert werden, zu denen die Nutzer_innen sich verhalten. Drittens analysiere ich das spezifische Zeitgefühl, das viele Linux-Nutzer_innen bei der Bearbeitung technischer Probleme zeigen. Anhand dieser Themenkomplexe verdeutliche ich, dass die Beobachteten sich mit ihren Lernfortschritten ein selbstbestimmtes Verhältnis zu ihren Maschinen aneignen: Sie sehen sie als menschengemacht und veränderbar an, sind jedoch nicht für jede Einschreibung von Konnotationen gleichermaßen sensibilisiert.

Im siebten Kapitel richte ich den Blick auf Praktiken, die auf einer sehr grundsätzlichen Ebene ordnenden, formalisierenden und hierarchisierenden Charakter haben. Einerseits geht es um den Umgang mit Zugriffsrechten und Passwörtern als wiederkehrende inhaltliche Themen der Kleingruppenarbeit in LUGs. Hier werden Ordnungen und Hierarchien in Softwarekonfigurationen abgebildet. Andererseits beleuchte ich ein Ritual auf Community-Events, die Keysigning-Party, als Praxis, die die Anwesenden nach verschiedenen Kriterien ordnet. Beide Themenbereiche zielen inhaltlich darauf ab, durch starke Formalisierung Zurechenbarkeit von digitalen Daten zu realen Personen herzustellen. Die beobachtbaren Praktiken stehen jedoch häufig im Widerspruch zu dieser formalen Logik.

Im achten Kapitel trage ich schließlich die gewonnenen Erkenntnisse zusammen und identifiziere mögliche Anknüpfungspunkte für weitere Forschungsarbeiten.

2. Einführung ins Forschungsfeld FLOSS-Communities

Im Mittelpunkt dieser Arbeit stehen Linux User Groups und gruppenübergreifende Community-Events. Beide Kontexte meiner Datenerhebung stehen jedoch in einem größeren Zusammenhang, der in diesem Kapitel umrissen wird. Die Freie-Software-Szene ist in vielen, weltweit verteilten Communities organisiert, die sich um bestimmte Ideen, Projekte oder Veranstaltungen gruppieren. Zur Einführung in dieses Forschungsfeld gehe ich auf die bereits existierende Literatur kurz ein. Die zentralen Arbeiten in diesem Bereich ziehe ich in der Materialauswertung nochmals detaillierter heran.

2.1. FLOSS-Communities als Teile einer Szene

Der Begriff ‚Community‘ ist eine Selbstbezeichnung. In der Forschung gibt es verschiedene Ansätze, FLOSS-Communities zu denken, und verschiedene Bezeichnungen für sie. Christopher Kelty entscheidet sich in seiner Ethnografie über die kulturelle Bedeutung freier Software für den schwammigen Begriff *movement*: „It may be that *movement* is the best term (...), but unlike social movements, whose organization and momentum are fueled by shared causes or broken by ideological dispute, Free Software and Open Source share practices first, and ideologies second“ (Kelty 2008b: 113). Er grenzt diese Bewegung von sozialen Bewegungen ab, weil sie sich in erster Linie über gemeinsame Praktiken konstituiert, die Bedeutung von *movement* bleibt jedoch unklar. Sehr viele Autor_innen arbeiten mit dem Begriff der *Communities of Practice*. Sie beziehen sich damit auf eine lerntheoretische Publikation von Jean Lave und Etienne Wenger (Lave/Wenger 1991; vgl. z. B. Lin 2004b; Bolstad 2006; Lloyd 2007; Fried 2008; O’Neil 2009: 25 f.; Berdou 2007).¹ Die For-

¹Die Publikation von Lave / Wenger ist im erziehungswissenschaftlichen Kontext entstanden, wurde aber von verschiedenen anderen Disziplinen rezipiert und adaptiert (vgl. Fried 2008: 24). Interessanterweise setzte sehr schnell auch eine wirtschaftswissenschaftliche Auseinandersetzung mit dem Konzept von *Communities of Practice (CoP)* ein, in der Wissen als Produktionsfaktor und ein gelungenes Wissensmanagement in Unternehmen als wettbewerbsentscheidendes Differenzkriterium betrachtet wird (vgl. Zboralski 2007: 2). Zum Begriff *Community of Practice* und seinen verschiedenen Bedeutungen, Aneignungen und Übersetzungen vgl. ebenfalls Zboralski (2007: 26 ff.).

schung zu FLOSS-Communities ist sich darüber einig, dass Lernen und Kompetenzerwerb in Communities sehr relevant ist (vgl. z. B. Glott et al. 2007; Weller/Meiszner 2008; Ghosh/Glott 2005; Huysman/Lin 2005; Sowe et al. 2008). Aus diesem Grund greifen viele auf das Konzept zurück. Unter *Communities of Practice* versteht Wenger Gruppen, die sich — ob auf der Arbeit oder in der Freizeit — durch drei Dimensionen auszeichnen: Die Mitglieder teilen ein Unterfangen (1), das sie ständig neu definieren und verhandeln. Dazu gehen sie verbindliche Beziehungen (2) ein. Auf Dauer entsteht dadurch ein geteilter Bestand an Ressourcen (3), z. B. Wissen, Routinen, ein Sprachgebrauch (vgl. Wenger 1998: 73 ff.; Fried 2008: 25). Lave / Wenger zielen mit dem Konzept auf eine bestimmte Lernkonstellation ab, die „legitime periphere Partizipation“, in der Neulinge soweit in die Praxis hineinwachsen, bis sie selbst zu vollen Community-Mitgliedern werden (vgl. Lave/Wenger 1991: 53). Der in der FLOSS-Forschung breit rezipierte Ansatz wird in Kapitel 5 ausgeführt und auf seine Angemessenheit für Linux User Groups hin befragt. Dabei wird sich zeigen, dass er diese Lernverhältnisse nur zum Teil erfasst — aus diesem Grund verwerfe ich *Communities of Practice* als Bezeichnung für die untersuchten Gruppen.

Ich werde vielmehr die Selbstbezeichnung als „Communities“ beibehalten. Um sie soziologisch zu denken, schlage ich vor, sie in Anlehnung an *Leben in Szenen* von Ronald Hitzler, Thomas Bucher und Arne Niederbacher als Teile einer größeren Szene zu verstehen (Hitzler et al. 2001). Hitzler et al. beziehen sich zwar auf Jugendszenen, ihr Ansatz passt jedoch auch in diesem Kontext: Sie stellen die Vergemeinschaftung in Szenen in Zusammenhang mit der heute hochgradigen Individualisierung. Szenen rekurrerten — im Gegensatz zu herkömmlichen „Sozialisationsagenturen“ wie der Kirche oder Verbänden — nicht mehr so stark auf ähnliche Traditionen oder soziale Lagen der Mitglieder (vgl. Hitzler et al. 2001: 17 f.). Als Szene betrachtet, sind FLOSS-Communities „[t]hematisch fokussierte kulturelle Netzwerke von Personen, die bestimmte materiale und/oder mentale Formen der kollektiven Selbststilisierung teilen und Gemeinsamkeiten an typischen Orten und zu typischen Zeiten interaktiv stabilisieren und weiterentwickeln“ (ebd. 20). Die Communities sind thematisch auf die Entwicklung und Verbreitung von freier Software fokussiert. Der Begriff ist nicht eindeutig: Zum einen bezeichnet er „die FLOSS- oder Linux-Community“ allgemein. So wird auf der Website des Linuxtags 2008 „die Community“ als die „engagierte Gemeinschaft von Entwicklern und Nutzern“ beschrieben:

„Die vielbeschriebene → GNU/Linux-Community ist der zentrale Motor für die Weiterentwicklung freier Software. Diese Gemeinschaft sichert den Fortbestand und den

Fortschritt des freien Betriebssystems GNU/Linux und die darauf aufsetzende Entwicklung für Freie Software“.²

Zum anderen ist mit einem engeren Verständnis von „Community“ eine bestimmte Gruppe gemeint, die aktiv ein bestimmtes Softwareprojekt mitentwickelt, die sich als Austauschplattform von Nutzer_innen einer bestimmten Software trifft oder die aktiv an der Verbreitung von freier Software arbeitet. Die Vielfalt der Communities ist groß, es gibt in der Forschung jedoch einen Konsens über bestimmte Gemeinsamkeiten: Communities konstituieren sich über die gemeinsame Arbeit an einem Projekt, sie teilen gemeinsame Werte und Ziele und sie kooperieren in erster Linie über das Internet.³ FLOSS-Communities zeichnen sich durch eine hohe Anzahl aktiver Freiwilliger und konstante Zusammenarbeit von Programmierer_innen und Nutzer_innen aus (vgl. Haralanova 2010: 27 f.). Dass Software von Communities kollaborativ entwickelt wird oder sich Communities um Softwareprojekte bilden, ist ein Unterschied zur proprietären Softwareentwicklung, wo ein abgeschlossenes Team am unveröffentlichten → *Quellcode* arbeitet (vgl. für detailliertere Ausführungen z. B. Grassmuck 2004: 233 f.; als szeneeinternes Dokument siehe Raymond 2000; dazu kritisch Bezkourov 1999). Große bekannte Community-Projekte sind → *Distributionen* wie *Debian-Linux*, freie → *Webbrowser* wie *Mozilla* oder *Firefox* oder der → *Webserver Apache*. Nicht jede freie Software entsteht in Communities. Die Mehrheit der Projekte wird von einer einzigen oder sehr wenigen Personen entwickelt und bekommt wenig Aufmerksamkeit (vgl. Krishnamurthy 2002).

2.1.1. Zum sozioökonomischen Hintergrund von FLOSS-Entwickler_innen

Szenen sind laut Hitzler et al. dynamisch und nicht statisch kartografierbar (vgl. Hitzler et al. 2001: 223). Demnach kann die quantitative Studie zu sozioökonomischen Hintergründen von FLOSS-Entwickler_innen von Rishab Ghosh, Rüdiger Glott, Bernhard Krieger und Gregorio Robles (Ghosh et al. 2002) nur als Momentaufnahme gelten. Sie ist zudem bereits älter und konzentriert sich (ohne dies zu benennen) auf Softwareentwicklung im Sinne von Programmierung. Sie beruht jedoch auf einem repräsentativen Sample von

²<http://www.linuxtag.org/2008/de/community.html>, Abruf: 29.07.2009

³Dies deckt sich mit einer Beschreibung des Community-Gedankens in einem meiner Interviews: „Und was das [→ *Debian-Projekt*, S. M.] zu 'ner Community macht, sind eigentlich die Mailinglisten, → *IRC*, Events und zusätzlich gibt's dann halt noch einmal im Jahr bei Debian die → *debconf*, wo sich alle Entwickler oder alle, die halt Zeit haben, treffen. (...) Aber, ja, die Community ist eigentlich die Kommunikation einfach im Netz, viele Leute sind da auch so dann befreundet, arbeiten zusammen“ (Int 2007a).

über 2.700 Entwickler_innen aus vielen Ländern und erlaubt damit eine Einschätzung darüber, wie die Communities sich zusammensetzen. Es ergibt sich insgesamt das Bild, dass wir es *hauptsächlich* mit jungen Männern aus den USA, Frankreich und Deutschland zu tun haben, die ein recht hohes formales Bildungsniveau und einen professionellen IT-Hintergrund haben. Die meisten Befragten waren 2002 zwischen 16 und 36 Jahre alt und seit der zweiten Hälfte der 1990er Jahre im Bereich freie Software aktiv. 70 Prozent hatten akademische Abschlüsse (vgl. ebd. 12). 60 Prozent lebten in festen Beziehungen, mehr waren ledig als verheiratet, nur 17 Prozent hatten Kinder. Den Autor_innen ist dies wichtig, weil es ein Klischee widerlege: „[T]he often-mentioned assumption that OS/FS⁴ developers are singles that are bored and have no partnership obligations and responsibilities is apparently not true“ (Ghosh et al. 2002: 11; vgl. auch Jung 2006: 245). Rund zwei Drittel waren Angestellte, während unter den Befragten kaum Erwerbslose waren, wohl aber etliche Selbstständige. 83 Prozent arbeiteten im IT-Sektor oder hatten an Universitäten mit Informationstechnologien zu tun. Zwei Drittel des Samples waren professionelle Softwareentwickler_innen, 16 Prozent Studierende, weitere waren etwa in der IT-Beratung tätig. Die Autor_innen heben hervor, dass die Befragten trotz hoher formaler Bildungsabschlüsse und Jobs im IT-Bereich insgesamt keine Top-Verdiener_innen seien: Die Einkommen der meisten liegen bei bis zu 3.000 US-Dollar im Monat (vgl. ebd. 12 ff.). In der Umfrage wird zwischen Nationalität und derzeitigem Wohnort unterschieden, französische Staatsbürger_innen und in Frankreich Lebende sind mit 15 beziehungsweise 16 Prozent am stärksten vertreten.⁵ Deutschland und die USA folgen vor weiteren EU-Staaten (vgl. Abb. 2.1).

Ghosh et al. eruieren, wie viel Zeit die Befragten in Projekte steckten: Rund 70 Prozent verbrachten maximal zehn Stunden pro Woche mit ihrem Engagement, und zwar weniger zu Hause als vielmehr am Arbeitsplatz, in der Universität oder Schule. Etwa die Hälfte entwickelte auch proprietäre Software (vgl. ebd. 20 f.). Die meisten waren in bis zu fünf verschiedenen Projekten aktiv, einige wenige in noch mehr (vgl. ebd. 30 f.). Das Engagement in einem FLOSS-Projekt ist damit im Leben der meisten Beteiligten eine „Teilzeit-Welt“, die für sie weder die Beteiligung an anderen Communities ausschließt, noch ihnen die „Sorge für Existenz, Sinnggebung, Geld“ abnimmt (Hitzler et al. 2001: 18). Die meisten hatten regelmäßigen Kontakt zu anderen Entwickler_innen, wobei Personen, die führen-

⁴OS/FS steht für „Open Source/Free Software“.

⁵Ghosh et al. geben hierbei zu bedenken, dass das auch an der Verteilweise der Online-Fragebögen liegen kann: Sie gingen den Projektorganisator_innen zu und wurde über sie per Schneeballprinzip weiterverteilt.

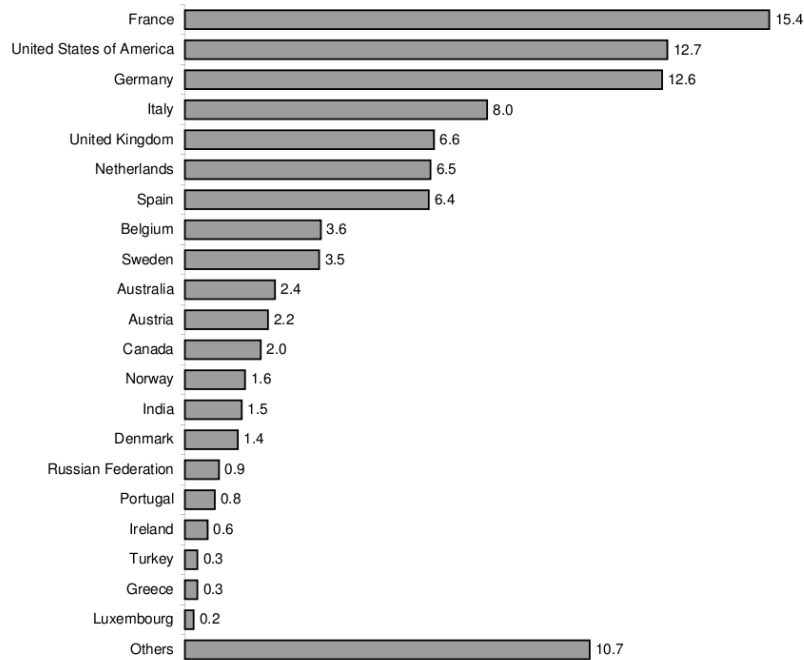


Abb. 2.1.: Wohnsitze von FLOSS-Entwickler_innen

de Positionen in Projekten einnahmen, auch signifikant mehr Kontakte geknüpft haben. Insgesamt zeigten die Befragten gute Kenntnisse von Szene-Strukturen und Schlüsselpersonen über ihre eigene Community oder ihr Projekt hinaus (vgl. Ghosh et al. 2002: 40 ff.). Die einzelnen Communities innerhalb der FLOSS-Szene öffnen sich zueinander, sie teilen Interessen und es gibt oft personelle Überschneidungen. Die Projektleiter_innen und Organisator_innen können mit Hitzler als communityübergreifendes Netzwerk von Eliten bezeichnet werden (vgl. Hitzler et al. 2001: 25 ff.).

Was die Motivationen der Community-Mitglieder angeht, sich in den FLOSS-Projekten zu engagieren, so stehen die individuellen Fähigkeiten eindeutig im Vordergrund, soziale Aspekte spielen eine nachgeordnete Rolle:

„Almost eight out of ten software developers started with OS/FS because they wanted to learn and develop new skills, and half of the sample claimed that they wanted to share their knowledge and skills with other software developers“ (Ghosh et al. 2002: 44).

Als weitere Motivationen nannten die Befragten die Beteiligung an neuen Kooperationsformen oder den Wunsch, Software anderer Entwickler_innen zu verbessern. Manche ar-

beiteten an einem Projekt mit, um an der FLOSS-Szene teilzuhaben oder weil sie proprietäre Software ablehnten. Etliche wollten auch ein konkretes Problem lösen, das sie mit proprietärer Software nicht lösen konnten, oder waren auf der Suche nach Hilfe bei der Umsetzung eigener Ideen. Ihre Motivationen verschoben sich mit der Dauer ihres Engagements: In der initialen Phase standen klar die Ausbildung individueller Fähigkeiten und der Austausch von Informationen und Wissen im Mittelpunkt. *Nachdem* die Befragten Erfahrungen in einer Community gesammelt hatten, rückten andere Motivationen in den Vordergrund. Sie schätzten vielmehr die gesamte Szene als Kontext, sahen die Chance, über das Engagement Geld zu verdienen, und hoben einen politischen Aspekt hervor: Mit der Dauer ihrer Involviertheit schätzten viele um so mehr, ein unkommerzielles Produkt herzustellen; für sie wurde es wichtiger, die Macht großer Softwarefirmen zu begrenzen (vgl. ebd. 44 f.). „To provide more freedom in software development and to serve as an institution where knowledge can be exchanged are the two most pronounced purposes of the OS/FS community“ (ebd. 49). Die Autor_innen der Arbeit heben hervor, dass viele der Entwickler_innen davon überzeugt seien, sie nähmen für sich mehr aus ihrem Engagement mit als sie hineinsteckten.

Meine Beobachtungen bestätigen, dass der Erwerb von Kompetenzen auch in Linux User Groups zentral ist: Gerade dort stehen der Wissenstransfer und das gegenseitige Helfen im Mittelpunkt. Der bei Ghosh et al. angeführte Eindruck, im Community-Leben mehr zu bekommen als man gibt, spiegelt sich auch in der verbreiteten Redewendung „der Community etwas zurückgeben“ wider: Personen engagieren sich als Gegenleistung für die Software oder die Hilfestellung, die sie in der Gruppe erhalten haben. FLOSS-Communities sehen sich jedoch explizit nicht als Lieferanten einforderbarer Produkte — eine Anspruchshaltung gegenüber den Aktiven gilt als verpönt. Hitzler et al. unterscheiden zwischen Aufklärungsszenen, Selbstverwirklichungsszenen und hedonistischen Szenen. Aufgrund der starken Vorrangstellung von Wissenstransfer in den Communities lassen sich FLOSS-Communities als Selbstverwirklichungsszene bezeichnen: Die Ausbildung persönlichen Könnens steht im Vordergrund, persönliche Freiheit und Selbstbestimmung nehmen einen wichtigen Stellenwert ein. Dagegen wird selten ein grundsätzlicher Weltveränderungsanspruch erhoben, wie dies etwa in politischen Aufklärungsszenen der Fall sei (vgl. Hitzler et al. 2001: 224 ff.). „World domination“ für freie Software wird zwar in der FLOSS-Szene öfter als Fernziel genannt, jedoch stets mit einem zwinkernden Auge. Hitzler et al. geben zu bedenken, dass jede Szene in dem Sinne exklusiv sei, dass sie den Mitgliedern längerfristigen Kompetenzerwerb abverlangt. Auch wenn bloßes Interesse zum

Eintritt ausreiche, könne „[v]olle Teilhabe (...) erst durch Aneignung und kompetente Anwendung szenetypischen Kultur-, Know Hows“ (ebd. 22) erreicht werden. Dies sei „grundsätzlich nicht ad hoc erlernbar“ (ebd. 215). *Wie* voraussetzungsvoll die Teilhabe an FLOSS-Communities durch die Rolle technischen Fachwissens ist, wird in den folgenden Kapiteln sichtbar.

Bezüglich der Entlohnung erhoben Ghosh et al. (2002), dass knapp die Hälfte der Befragten kein Geld mit ihrer Arbeit für FLOSS-Projekte verdienen. Jeweils zehn bis zwanzig Prozent gaben an, für die Entwicklung oder Administration freier Software oder für Support-Leistungen eine *direkte* Entlohnung zu erhalten. Weitaus weniger erhielten *indirekt* Geld über ihr Engagement, weil sie z. B. aufgrund der Erfahrungen in FLOSS-Communities eine bezahlte Stelle gefunden hätten oder bei der Lohnarbeit auch freie Software entwickelten, obwohl dies in der Stellenausschreibung nicht erwähnt worden sei (vgl. ebd. 63 ff.).

2.1.2. Die Rolle von Privatwirtschaft und Stiftungen in der Szene

Die Privatwirtschaft muss im Kontext von FLOSS-Communities erwähnt werden, sie spielt in dieser Arbeit jedoch nur eine marginale Rolle. Sie kommt insofern zum Tragen, als sich für viele Community-Mitglieder die Beschäftigung mit Linux zwischen Hobby und Professionalisierung bewegt. Sehr viele Aktive arbeiten auch im IT-Bereich (vgl. Ghosh et al. 2002: 12 f.), viele andere äußern dies als Berufswunsch. Auch Erwerbslose versuchen teilweise, sich dort weiterzubilden oder sich in der Community einen Ruf zu erarbeiten, um ihre Chancen auf einen Job oder Quereinstieg in IT-Berufe zu erhöhen. IT-Profis, die mit freier Software arbeiten, finden in der Community den Austausch mit Kolleg_innen.

Firmen sind jedoch nicht nur potenzielle Arbeitgeber, sondern sie tragen auf verschiedene Weise zur Entwicklung freier Software bei: Sie geben z. B. den Quellcode eigener, vormals proprietärer Software lizenzrechtlich frei und lassen ihn von Communities weiterentwickeln. Das bekannteste Beispiel ist die Veröffentlichung des *Netscape* → *Webrowsers*, der im so genannten „Browserkrieg“ um Marktanteile dem Konkurrenten *Microsoft Internet Explorer* im Jahr 1998 unterlegen war (vgl. z. B. Moody 2002: 182 ff.). Der Quellcode wurde vom *Mozilla*-Projekt aufgegriffen und weiterentwickelt. Inzwischen basieren weitere bekannte Programme auf *Mozilla*, z. B. *Firefox* oder *Camino*.

Firmen machen durch Sponsoring oder Spenden für Community-Events oder Gruppen auf sich aufmerksam oder sie präsentieren sich mit Ständen auf Events. Sie sind nicht zu-

letzt maßgebliche Entwickler und Nutzer freier Software. Auf Seiten der Systemadministration, also im Bereich der Verwaltung kompletter IT-Landschaften mit diversen → *Servern* und vielen Nutzer_innen, sind freie Werkzeuge verbreitet. Es gibt starke personelle Überschneidungen zwischen Systemadministrator_innen, die sie nutzen, und den Communities. Teilweise werden Softwareentwickler_innen von Firmen gezielt für die Arbeit an freien Softwareprojekten bezahlt oder können einen Teil ihrer Arbeitszeit dort hineinstecken (vgl. Ghosh et al. 2002: 63 ff.). Dahinter können wirtschaftliche Erwägungen stehen, z. B. dass die Anpassung einer freien Software durch eine bezahlte Arbeitskraft langfristig weniger kostet als die Lizenzgebühren für ein proprietäres Produkt, das darüber hinaus Einschränkungen mit sich bringt. Es kann auch eine gezielte Marketingstrategie sein, auf freie Software zu setzen. So werben manche in der Szene aktive Firmen gerade mit der Transparenz des Produktes freie Software für Kund_innen, denen der Quellcode mit ausgeliefert wird. Darüber hinaus wird mit der Garantie so genannter offener Standards geworben. Das bedeutet, dass die von der Software produzierten Daten zu anderen Programmen kompatibel sind, die ebenfalls offene Standards nutzen.⁶ Yuwei Lin hat die Kooperation zwischen Firmen und Communities untersucht. Im Unterschied zu Arbeiten, die diese Zusammenarbeit spieltheoretisch denken, fragt sie danach, was konkrete Praktiken wie gegenseitige Hilfe für hybride Innovationsmuster bedeuten. Sie stellt unter den Akteur_innen eine große gegenseitige Unterstützung und hohen Informationsfluss bei sehr verschiedenen sozio-technischen Hintergründen fest (vgl. Lin 2004b: 186, 190 f., 198). Lin resümiert, dass die Zusammenarbeit zwischen Unternehmen und FLOSS-Communities ständig ausgehandelt wird. In diesem Innovationsmodell könnten Ressourcen beider Sphären genutzt werden: Die Communities böten Raum für Experimente und informelle Kommunikation, während der private Sektor den Entwicklungsprozess eher stabilisiere, standardisiere und auf Märkte bringe. Sie benennt auch die erhöhte Wahrscheinlichkeit, dass Konflikte z. B. über ethische oder moralische Fragen zwischen den Akteur_innen auftreten, die für das Modell Risiken bergen (vgl. ebd. 218).

Es gibt etliche Stiftungen, die Projekte oder Gruppen aus der Szene unterstützen. Ein Beispiel ist die *Ubuntu Foundation*, die der Multimillionär Mark Shuttleworth 2005 gegründet hat. Sie finanziert z. B. Entwickler_innen für die Fortentwicklung von *Ubuntu*. Die *Free Software Foundation* hat Richard Stallman 1985 gegründet. Ihre Aufgabe ist in erster Linie

⁶Ein Unterbereich offener Standards sind offene Dateiformate, in denen Dateien ohne rechtliche Einschränkungen gespeichert werden können. Für Office-Dokumente ist ein standardisiertes offenes Format „OpenDocument“. Textdateien im „freien“ Office-Programm OpenOffice.org werden standardmäßig in diesem Format gespeichert, nicht als Microsoft-formatierte Dokumente.

die Unterstützung des *GNU*-Projektes in finanzieller, personeller, technischer und juristischer Hinsicht. Die Stiftung hat weltweit einige Schwesterorganisationen. Die *Mozilla Foundation* koordiniert seit 2003 die Entwicklung des gleichnamigen Browsers und einiger seiner → *Derivate* und bezahlt ebenfalls Softwareentwickler_innen. Volker Grassmuck erklärt die Gründung von Stiftungen oder nicht profitorientierten Unternehmen damit, dass es für bestimmte Kooperationen zwischen Communities und Firmen einer Rechtsform bedarf, z. B. um Spenden annehmen oder sich an Industriekonsortien beteiligen zu können (vgl. Grassmuck 2004: 247 f.).

2.1.3. Kooperation und interne Strukturen von FLOSS-Projekten

In diesem Abschnitt soll die Kooperation innerhalb von FLOSS-Projekten kurz angerissen werden. Zahlreiche Forschungsarbeiten haben sich mit diesem Themenkomplex beschäftigt. Er ist für meine Fragestellung nur in Teilen relevant, weil Softwareprojekte, die regelmäßig auf die Veröffentlichung stabiler Software hinarbeiten, anders organisiert sind als LUGs. Zudem ist die Landschaft der FLOSS-Projekte so ausdifferenziert, dass kaum übergreifende Forschungsarbeiten existieren. In der verbreiteten Wahrnehmung dominieren große Softwareprojekte, an deren Entstehen, Weiterentwicklung und Verbreitung viele Menschen beteiligt sind. Solche großen Projekte sind beispielsweise → *Distributionen* oder sehr verbreitete Projekte wie der → *Webserver Apache*, an dem neben Communities auch die Privatwirtschaft beteiligt ist. Unter den Distributionen ist *Debian* das vermutlich am häufigsten untersuchte Projekt, weil es nicht nur sehr groß ist, sondern weil es sich einen expliziten Sozialvertrag gegeben hat, den *Debian Social Contract*. Darin sind einige gemeinsame Richtlinien festgehalten, etwa, dass *Debian* ausschließlich freie Software aufnimmt und wie die auszusehen hat, dass die Bedürfnisse von Nutzer_innen Priorität haben oder dass Fehler in einem öffentlich einsehbar System verwaltet werden.⁷ *Debian* ist hinsichtlich aller Prozesse stark verregelt: Jährlich wird ein_e Projektleiter_in gewählt, für die Aufnahme neuer Beitragender gibt es einen vorgesehenen Ablauf, der einem Mentoring ähnelt (zur Organisation bei *Debian* zwischen Demokratie und Bürokratie vgl. O’Neil 2009: 129 ff.; O’Mahony/Ferraro 2007; zu Ethik Coleman/Hill 2005; Krafft 2010; zur Distribution *Ubuntu* siehe Lloyd 2007).⁸

⁷vgl. http://www.debian.org/social_contract

⁸Fallstudien zu anderen konkreten Projekten finden sich z. B. im Sammelband von Feller et al. (2005).

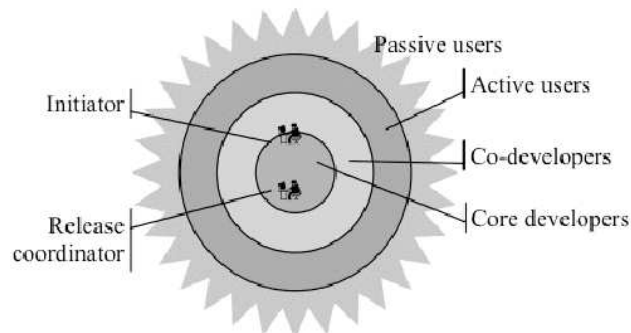


Abb. 2.2.: Teamstruktur eines FLOSS-Projektes

Crowston/Howison (2004) arbeiten aus Fallstudien zu FLOSS-Projekten ein Zwiebelmodell (Abb. 2.2) heraus, das zeigen soll, wie freie Softwareprojekte aufgebaut sind. Im Zentrum eines Projektes stehe ein Kern von Programmierer_innen, die den größten Teil des Quellcodes zum Projekt beitragen und den Überblick über die Entwicklung des Projektes haben. Um diesen Kern herum befänden sich weitere Entwickler_innen („co-developers“), die → *Patches* einschicken, also maßgeblich daran beteiligt sind, Fehler zu beheben und neue Funktionalitäten einzubauen. Ihre Beiträge würden vom Kernteam überprüft und übernommen. Der noch größere Kreis von „aktiven User_innen“ programmiere nicht selbst, sondern teste die Software, berichte Fehler und mögliche Einsatzfelder der Software. Schließlich gebe es die „passiven User_innen“, die die Software einsetzen, sich aber nicht in den Kommunikationskanälen des Projektes (Mailinglisten, Foren u.ä.) äußerten (vgl. Crowston/Howison 2004: 7). Hitzler et al. denken die Involviertheit von Personen in Szenen ähnlich zentralistisch: Szenen gruppierten sich um die Mitglieder der Organisationselite, die in engem Kontakt mit den ständig in der Szene Präsenten ständen. „Um diese beiden Gruppen herum formieren sich alle ‚normalen‘ Szenegänger (mit abnehmendem Engagement in der Szene nach außen hin)“ (Hitzler et al. 2001: 28). Eine Szene sei dabei — wie auch bei FLOSS-Communities — nicht abgekoppelt betrachtbar, sondern die Organisationseliten seien Teil größerer Netzwerke, in denen sich ähnliche Szenen oder Gruppen überregionaler Szenen miteinander vernetzten. In FLOSS-Communities entsteht diese Vernetzung online und auf Community-Events. Die Übergänge zwischen Szeneeliten, sehr Aktiven, ‚normalen‘ Szenegänger_innen und Publikum seien stets fließend: „Grundsätzlich gilt, daß scharfe Gruppen- oder Szenegrenzen (...) nicht existieren. Gerade eine solche Unschärfe bzw. eine solche Offenheit und Durchlässigkeit macht Szenen aus“ (ebd. 28). Zudem gebe es in den meisten Communities keine formale Mit-

gliedschaft, sondern die Kooperation beruhe auf freiwilliger Selbstbindung, die jederzeit widerrufen werden könne (vgl. ebd. 22).

Die Frage nach Szeneeliten führt zur Rolle von bekannten Einzelpersonen, so genannten ‚Gurus‘, in FLOSS-Communities. Einzelpersonen können in Communities oder szeneweit große Bekanntheit erlangen, die nicht nur die Anerkennung Gleichgesinnter bedeutet, sondern Einladungen als Referent_in auf Community-Events, mediale Aufmerksamkeit oder die Vermittlung bezahlter Arbeitsstellen nach sich ziehen kann. Beispiele für solche Mitglieder des Szenekerns sind der Mitbegründer des → *GNU*-Projektes Richard Matthew Stallman oder Kernel-Entwickler Linus Torvalds. Beide stehen für zwei verschiedene Strömungen innerhalb der Szene: Wer sich auf Stallman bezieht, wendet sich gegen jedes Verwenden proprietärer Anteile in freien Betriebssystemen. Die meisten Distributionen enthielten unfreie Anteile und seien abzulehnen.⁹ Diese Bezugnahme zeigen manche Community-Mitglieder explizit in ihrem Sprachgebrauch, wenn sie — wie von Stallman gefordert — von „GNU/Linux“ sprechen (vgl. Moody 2002: 92). In den Räumen einer LUG hing Stallmans Konterfei als Gemälde, mit den Hörnern eines Gnus versehen. Linus Torvalds, der Namensgeber von „Linux“, hat 1991 die erste Version des Betriebssystemkerns von Linux geschrieben und koordiniert das → *Kernel*-Projekt bis heute mit einem eher zentralistischen Führungsstil (vgl. Grassmuck 2004: 227). Er wird der Open-Source-Bewegung zugeordnet, die die Veröffentlichung des Quellcodes nicht politisch oder moralisch auflade (vgl. Nuss 2006: 77 f.). Seine ersten Kernelversionen standen nicht unter einer freien Lizenz (vgl. Torvalds/Diamond 2001: 103 ff.; Moody 2002: 48 f.). In der Szene ist er keine so starke Identifikationsfigur wie Stallman. Neben den beiden gibt es weitere szeneweite Eliten sowie „Gurus“, die sich thematisch spezialisiert haben.

2.1.4. Anliegen und ‚Philosophie‘ von FLOSS-Communities

Der ideelle Hintergrund von FLOSS-Communities stellt einen wichtigen Bezugspunkt dar und wird ‚die Philosophie‘ genannt. Einige programmatische Texte von Mitgliedern der Community-Eliten sind sehr bekannt, so das *GNU Manifesto* von Richard Stallman. Stallman war Mitte der 1980er Jahre Programmierer am *Massachusetts Institute of Technology (MIT)* in Cambridge/Massachusetts in den USA. Gemäß seiner Darstellung haben ihn die

⁹Es existiert sogar ein Programm für die → *Kommandozeile*, das einen Computer nach unfreier Software durchsuchen kann. Es heißt *vrms* — „virtual Richard Matthew Stallman“.

rechtlichen Einschränkungen, die mit proprietärer Software verbunden sind, so sehr gestört, dass er seine Stelle aufgeben habe, um ein freies Betriebssystem zu entwickeln. Er ist einer der Begründer des → GNU-Projektes, hat die Lizenz → *GNU General Public License* mitentwickelt und die *Free Software Foundation* mitgegründet. Seine Hauptaussage ist, dass „ethisches“ Programmieren nur mit komplett freier Software möglich sei. Damit macht er eine klare Dichotomie zwischen freier (moralischer, in freundschaftlicher Zusammenarbeit entstandener) Software und proprietärer (unmoralischer, restriktiver) Software auf (vgl. Stallman 1985). Die Firmen *Microsoft* und *Apple* sind wichtige „Feindbilder“, die der Abgrenzung dienen. Diese Konstruktion wird von breiten Teilen der Community geteilt und sehr häufig aufgegriffen, z. B. in Witzen. Die ‚Philosophie‘ von freier Software steht in einem Spannungsfeld zwischen Ethik und nicht durch Patente eingeschränktem Markt. Jede_r könne Änderungen an freier Software vornehmen oder eine_ Programmierer_in dafür bezahlen, die Software auf die eigenen Bedürfnisse zuzuschneiden. Das *GNU Manifesto* räumt damit der Tatsache Raum ein, dass viele von Dienstleistungen in diesem Bereich leben (vgl. dazu kritisch Yeats 2006: 64). Um die Doppeldeutigkeit des englischen Wortes „free“ gab es Diskussionen. Es sollte vermieden werden, dass mit „free software“ in erster Linie kostenlose Software assoziiert wird, damit die anderen „Freiheiten“ der Software wahrgenommen werden. Stallman formulierte dazu das in der Szene berühmte Zitat „free as in freedom, not as in free beer“. Man muss damit rechnen, korrigiert zu werden, wenn man den unentgeltlichen Charakter der Software hervorhebt.

Die ‚Philosophie‘ der Szene ist stark von der so genannten Hackerethik geprägt. Sie ist mit Szeneeliten wie dem Technik-Redakteur Steven Levy verbunden, der die zuvor mündlich überlieferten Grundsätze 1984 als erster ausformuliert hat (Levy 2010 [1984]). Sie lauten: Der Zugang zu Computern sollte unbegrenzt sein, Informationen sollten frei verfügbar sein. Hacker_innen werden dazu aufgerufen, Autoritäten zu misstrauen und Dezentralisierung anzustreben. Sie sollten Hacker_innen nach ihren Taten beurteilen, nicht nach Kriterien wie Alter, Ausbildungsgrad, Herkunft, Hautfarbe, Ethnizität oder Position.¹⁰ Mit Computern kann man Kunst und Schönheit schaffen und sie können das Leben zum Besseren verändern (vgl. Levy 2010: 27 ff.). Eric Raymond, FLOSS-Entwickler der ersten Stunde, pflegte bis 2003 das Online-Dokument *The Jargon File* (Raymond 2003), das eine Art Lifestyle-Beschreibung ist und der Hackerethik zugerechnet wird. Die Grundsätze gehen auf die 1960er Jahre zurück, als sich Studierende in US-Universitäten gegen den restrikti-

¹⁰Der *Chaos Computer Club* (CCC), der sich ebenfalls auf die Hackerethik bezieht, erweitert die Kriterien wie folgt: „nicht nach üblichen Kriterien wie Aussehen, Alter, Rasse [sic!], Geschlecht oder gesellschaftlicher Stellung“ (<http://www.ccc.de/hackerethics>).

ven Zugang zu teuren Rechenkapazitäten auflehnten (vgl. Imhorst 2004: 23 ff.). Nafus et al. sehen in FLOSS-Communities *keine* einheitliche Identifikation mit dieser Hackerethik, wohl aber rege Diskussionen darüber. Gerade Neulinge setzten sich zur Orientierung in der Szene damit auseinander. Die Hackerethik sei insgesamt als Abgrenzung vom gesellschaftlichen Mainstream zu verstehen (vgl. Nafus et al. 2006: 23 ff.).

Innerhalb von FLOSS-Communities gibt es zwei verschiedene Strömungen, nämlich die Verfechter_innen von freier Software im Sinne Stallmans und die Anhänger_innen des Open-Source-Konzeptes. Bei der Spaltung der Szene geht es um die Frage, ob es in jedem Fall erlaubt sein sollte, quelloffene Software zu verändern, oder ob dies in bestimmten Fällen lizenzrechtlich unterbunden werden darf. Die Szene spaltete sich ab 1998 auf, als eine Gruppe um Eric Raymond die *Open Source Initiative (OSI)* gründete. Die Initiative wollte die Konfrontationshaltung gegenüber der Wirtschaft aufgeben. Sie hatte das Ziel, Firmen von der Veröffentlichung von Quellcodes zu überzeugen und diese Veröffentlichung zu einem Gütesiegel für Software zu machen. Stallmans Begriff „free software“ wurde von der *OSI* für zu ideologisch gehalten. Der Begriff „Open Source“ beinhaltet die grundsätzliche Zustimmung, dass quelloffene Software kommerziell verwendet werden darf (vgl. Nuss 2006: 78; Grassmuck 2004: 230). Die Bezeichnung „Open Source“ wurde erfolgreich in den Medien platziert und machte das Konzept der Quelloffenheit breiter bekannt. „Hinter der neuen Sprachpolitik standen auch Angriffe gegen die Person Stallmans“ (Grassmuck 2004: 231). Aus der Open-Source-Strömung wurde Kritik daran geübt, dass er Investor_innen verschrecke und innerhalb des *GNU*-Projektes einen zu kontrollierenden Führungsstil habe. Laut Volker Grassmuck ist die *OSI* mit ihrem Anliegen jedoch gescheitert: Die Quelloffenheit der Software, die sich dadurch zwar breiter etabliert habe, sei ohne Modifikationsfreiheit sinnlos (vgl. ebd.). Ghosh et al. (2002) haben in FLOSS-Projekten festgestellt, dass es sich bei diesen beiden Strömungen eher um eine ideelle Abgrenzung handelt. Die Anhänger_innen der unterschiedlichen Strömungen kooperierten in der Praxis dennoch. Tendenziell sei die Unterscheidung denjenigen wichtiger, die dem Konzept von freier Software anhängen. Nicht alle Beteiligten maßen der Unterscheidung überhaupt Bedeutung bei oder verorteten sich klar (vgl. Ghosh et al. 2002: 50 ff.).

2.1.5. Die Verbindung von Anerkennung und Geschlecht

Laut Ghosh et al. erwartete rund ein Drittel der Befragten von FLOSS-Communities persönlichen Respekt und Respekt für die eigenen Beiträge (vgl. Ghosh et al. 2002: 47). Aner-

kennung und auch Entscheidungsbefugnisse werden an das Engagement geknüpft. Dazu beziehen sich die Communities auf den für sie positiv besetzten Begriff ‚Meritokratie‘ (vgl. z. B. zum *Ubuntu*-Gründer Mark Shuttleworth Lloyd 2007: 105 ff.). Meritokratie bezeichnet die Organisation von Herrschaft oder von Entscheidungsstrukturen nach dem Leistungsprinzip und widerspricht damit egalitären Ansprüchen.¹¹ Die positive Konnotation kommt durch eine Abgrenzung von anderen Kriterien zustande, die in der Szene nicht als Legitimationsgrundlage von Positionen oder Befugnissen akzeptiert werden: Es ist sehr verbreitet, formalen Bildungsabschlüssen oder Titeln keine große Bedeutung beizumessen. Dadurch, dass viele FLOSS-Projekte hauptsächlich online kommunizierten, seien Geschlecht, Herkunft, Hautfarbe oder Alter der Beteiligten oft nicht einmal bekannt. Stattdessen würden die Community-Mitglieder einander anhand ihrer Beiträge zum Projekt einschätzen (vgl. Siefkes 2007: 16).¹² Zudem ist der Glaube verbreitet, dass jede_r sich in FLOSS-Communities einbringen kann; wer dies nicht tue, habe sich aus freien Stücken so entschieden.

Laut Ghosh et al. spielen Frauen im Sample der Untersuchung mit ca. einem Prozent „keine Rolle“ (Ghosh et al. 2002: 8). Auf dieses Ergebnis gibt es verschiedene Reaktionen, die im Folgenden vorgestellt werden. Christina Haralanova hinterfragt das Zustandekommen dieser Zahl und widerlegt sie. Patricia Jung, Dawn Nafus et al. sowie Yuwei Lin erklären die Unterrepräsentanz von Frauen mit szenespezifischen oder gesamtgesellschaftlichen sozialen Strukturen. Zudem identifizieren Haralanova und Jung es als Problem, dass Ghosh et al. mit ihrer breit rezipierten wissenschaftlichen Studie FLOSS-Communities als frauenfreies Umfeld reproduzieren (vgl. Haralanova 2010; Jung 2006: 241). Die Erklärungsansätze dieser Autor_innen liegen auf verschiedenen, wenn auch nicht trennscharfen Ebenen, nämlich auf der Ebene der geschlechtlich segregierten Arbeitsteilung, der vergeschlechtlichten Rollenzuschreibungen, auf der Ebene der Szenekultur und der der gesamtgesellschaftlichen strukturellen Ungleichheiten. Insgesamt lassen sich die Erklärungsansätze alle kaum von dieser gesamtgesellschaftlichen Ebene trennen; viele von ihnen sind in Bezug auf andere gesellschaftliche Bereiche entwickelt worden und sind auf FLOSS-Communities übertragbar.

Die geschlechtlich segregierte Arbeitsteilung ist eng mit den Anerkennungsstrukturen von FLOSS-Projekten verbunden: Viele Frauen engagierten sich für die wenig prestigeträchti-

¹¹Der Begriff wurde zuerst von Michael Young (1961) in dem Buch *Es lebe die Ungleichheit* verwendet.

¹²Christian Siefkes, ein Autor aus dem *Oekonux*-Zusammenhang, setzt sich damit auseinander, wie ‚Peer Production‘, also eine Produktionsweise, die Ebenbürtigkeit der Beteiligten voraussetzt, auf andere Gesellschaftsbereiche übertragen werden kann.

gen Aufgaben der Software-Dokumentation, der Bewerbung freier Software und der dazu nötigen Überzeugungsarbeit im beruflichen oder im privaten Umfeld oder bei organisatorischen Aufgaben innerhalb der Projekte. Prestigeträchtige Aufgaben seien dagegen die Programmierarbeit und quellcodenahe Hilfen wie das Einsenden von → *Bugreports* oder → *Patches* (vgl. Haralanova 2010: 124 ff.). Christina Haralanova spricht auf der Basis ihrer nicht repräsentativen Befragung von einem möglichen Frauenanteil von zehn bis 20 Prozent, der sich ergäbe, würde man alle Tätigkeiten in wissenschaftlichen Arbeiten erheben, statt sich nur auf das Programmieren zu beschränken. Sie geht von einem grundsätzlich breiteren Begriff von Beiträgen aus, indem sie die Unterscheidung aufhebt zwischen Engagement innerhalb der Szene und Engagement für die Verbreitung freier Software außerhalb der Szene: Die meisten von ihr befragten Frauen arbeiten *nicht* unmittelbar im IT-Bereich, teilweise sogar in ganz anderen Bereichen, z. B. in kulturellen oder künstlerischen Zusammenhängen, die mit freier Software zuvor nichts zu tun hatten. Sie setzten sich dort erfolgreich für den Einsatz von freier Software ein — ein Engagement, von dem die FLOSS-Communities oft nichts mitbekämen (vgl. Haralanova 2010: 122). Der Großteil der Befragten reproduzierte interessanterweise selbst die Vorrangstellung von Programmierenden: Sie siedelten *Code contributions* auf einer höheren Stufe an, die für sie unerreichbar sei und bleibe. Sie bezweifelten, dass sie selbst relevante Beiträge zur FLOSS-Bewegung leisteten, selbst wenn ihre Tätigkeitsbeschreibungen dazu im Widerspruch stünden (vgl. ebd. 109 f.).

Letztlich habe sich ein Begriff von ‚Entwicklung‘ etabliert, der die Aktivitäten innerhalb der Projekte unsichtbar macht, denen oft Frauen nachgehen:

„F/LOSS rewards the producing code rather than the producing software. It thereby puts most emphasis in a particular skill set. Other activities such as interface design or documentation are understood less 'technical' and therefore less prestigious. This has consequences (...) for the lower valuation of activities in which F/LOSS women often engage (...)“ (Nafus et al. 2006: 31).

Patricia Jung bezeichnet den in der Szene verwendeten Begriff Meritokratie deshalb als Euphemismus für diese Vorrangstellung von Programmierenden.¹³ Projekte mit meritokratischen Entscheidungsstrukturen sollten alle Aktiven in Entscheidungen einbeziehen,

¹³ „An der Spitze der Hierarchien in Open Source-Projekten stehen Programmierer, und selbst die Nomenklatur spricht Bände: Als Entwickler oder Developer gelten ausschließlich diejenigen, die Code schreiben, während alle anderen im besten Falle als Contributors, Beitragende, bezeichnet werden“ (Jung 2006: 241).

de facto hätten aber meist Programmierer_innen mehr Macht, da sie nur bestimmte Änderungsvorschläge umsetzten (vgl. Jung 2006: 241 f.; vgl. auch Nafus et al. 2006: 31). Nafus et al. zeigen, dass in Bezug auf Frauen in den männlich dominierten Communities die weiter unten erläuterten strukturellen Chancenungleichheiten ausgeblendet werden, da eine Teilnahme an den Projekten als individuelle Entscheidung wahrgenommen wird, die Frauen offenbar nicht getroffen hätten, wenn sie nicht präsent seien (vgl. Nafus et al. 2006: 17).

Zweitens werden vergeschlechtlichte Rollenzuschreibungen untersucht. Laut Nafus et al. verorten sich viele „Hacker“ außerhalb des gesellschaftlichen Mainstreams und sehen die FLOSS-Szene als Raum, in dem die sonst konventionellen Kommunikationsformen und Höflichkeitsregeln nicht unbedingt gelten. Viele bezeichneten sich selbst als sozial inkompetent, schrieben Frauen aber essentialisierend zu, soziale Wesen zu sein (vgl. Nafus et al. 2006: 26). Damit berufen sie sich auf vermeintliche Eigenschaften im „Wesen“ von Frauen, schreiben ihnen zu, soziale Aufgaben besser als technische zu erfüllen, und grenzen sie dadurch diskursiv aus dem angesehenen technischen Bereich aus. Aber auch die wenigen präsenten Frauen nehmen die Gender Gap nicht alle gleich wahr (vgl. Lin 2005a). Die von Haralanova interviewten Frauen assoziieren sich selbst klar mit sozialen Kompetenzen oder den ‚sozialeren Aspekten‘ von Technologien, so dass die Autorin so weit geht, den eigentlichen Beitrag von Frauen zu freier Software in der Vermittlung und Weiterverbreitung der Software und der damit verknüpften Ideen zu sehen. Eine veränderte Wahrnehmung von freier Software als sozialem Projekt könne mehr Frauen zum Engagement motivieren (vgl. Haralanova 2010: 109, 126 ff.). Sie befragt dieses Forschungsergebnis nicht daraufhin, ob es nicht selbst ein Stereotyp reproduziert, wenn es Frauen die Vermittlerrolle und das „Soziale“ zuschreibt. Solche Selbstzuschreibungen könnten wiederum gendersensiblen Frauen den Austausch und die Organisation erschweren (vgl. Lin 2005a). Überdies werden fehlende Mentor_innen bzw. Rollenmodelle oft als Erklärung für einen gering *bleibenden* Frauenanteil gesehen (vgl. Lin 2005a; Jung 2006: 246).

Ein dritter Erklärungsansatz ist die Szenekultur von FLOSS-Communities und dort besonders das Kommunikationsverhalten: Gerade in der Online-Kommunikation gebe es ein geradezu aggressives Redeverhalten. Vor allem die Community-Mitglieder, die dabei seien, sich einen Ruf aufzubauen, versuchten oft, sich durch so genannte → *Flamewars* sichtbar zu machen. Dies habe auf Neulinge eine ausschließende oder zumindest abschreckende Wirkung. Frauen betreffe dies besonders, weil sie häufig auf eine kürzere Erfahrung mit Computern zurückblickten als Männer und sich deshalb weniger selbst-

bewusst in technischen Fragen verteidigten (vgl. Nafus et al. 2006: 37 ff.). Ein anderes Symptom einer männlich dominierten Szene sei es, dass in Software und Dokumentationen in der Regel keine gendersensible Sprache, sondern lediglich die männliche Form verwendet werde, was Menschen, die sich geschlechtlich anders verorten, diskriminiere (vgl. Lin 2005a). FLOSS-Communities seien nicht nur von einer kooperativen Kultur und teilweise klaren Hierarchien geprägt, sondern auch von einer kompetitiven Kultur (vgl. Brand/Schmid 2006). Der Männlichkeitsforscher Michael Meuser nennt die „enge Verzahnung von Wettbewerb und Solidarität ein zentrales Element“ homosozialer Männergemeinschaften (Meuser 2008: 43).¹⁴

Auch offener Sexismus komme besonders in der Online-Kommunikation vor, „seltener im Direktkontakt z. B. auf User-Gruppen-Treffen“ (Jung 2006: 238; siehe auch Herring 1997). FLOSS-Communities unterlägen als Männerdomänen einer Dynamik „monogeschlechtlicher Gruppen“ (Jung 2006: 237): Wenn dort Frauen auftauchten, bekämen sie aufgrund ihres zugeschriebenen Geschlechts übermäßig viel Aufmerksamkeit, sexuelles Interesse, aber auch Interesse an Ratschlägen ‚von einer Frau‘. Die technischen Sachthemen rückten in der Kommunikation in den Hintergrund und Frauen, die kein Interesse an weitergehenden Beziehungen bekundeten, hätten es schwer, im Rahmen des FLOSS-Projektes zu kommunizieren oder Mentor_innen zu finden (vgl. Jung 2006: 237 f.; Nafus et al. 2006: 27 ff.). Für Jung ist die Szenekultur vor allem eine Erklärung für den unterschiedlichen Frauenanteil in FLOSS-Communities im Vergleich zur kommerziellen Softwareentwicklung, wo von einem Frauenanteil von rund 20 Prozent ausgegangen wird (vgl. Jung 2006: 236).

Schließlich stehen die genannten Erklärungsansätze mit weiteren gesamtgesellschaftlichen strukturellen Ungleichheiten in Zusammenhang, die sich auch in FLOSS-Communities zeigen: Frauen seien im technischen Bereich insgesamt unterrepräsentiert (vgl. zu Informatikstudiengängen im internationalen Vergleich Schinzel 2004a,b, 2005a,b). Frauen und Mädchen kämen später mit Computern in Berührung und bekommen ihre Kompetenzen eher im schulischen Rahmen vermittelt. Jungen dagegen machten ihre ersten Erfahrungen mit dem Computer oft in jungen Jahren mit ihren Vätern (vgl. Margolis/Fisher 2002: 24 f.). Das FLOSS-Umfeld sei jedoch stark auf Menschen angelegt, die langjährige Erfahrungen mit Computern haben. „In order to join women have a larger amount

¹⁴Meuser geht vor allem auf die Jugendphase ein, in der Männlichkeit über die Teilnahme an kompetitiven, teilweise den eigenen Körper aufs Spiel setzenden Spielen hergestellt würde, was Mädchen oder junge Frauen in dieser Form nicht zu lieben lernten (vgl. ebd.).

of catching up work to do, which they must do in an environment that almost exclusively values independent discovery“ (Nafus et al. 2006: 34 f.). Des Weiteren wird die ungleiche Verteilung der Haus- und Erziehungsarbeit angeführt, die häufig Männern mehr Zeit für ehrenamtliche Arbeit, z. B. stundenlanges Programmieren, lasse (vgl. zur ‚doppelten Vergesellschaftung‘ von Frauen Becker-Schmidt 2003; spezieller zu FLOSS-Communities Lin 2005a; Jung 2006: 244 ff.; Nafus et al. 2006: 40 ff.).

2.1.6. Lokale User Groups als Anlaufstellen

User Groups sind lokale Gruppen, die sich auf ehrenamtlicher Basis mit freier Software beschäftigen. Die Angaben über die Zahl der Linux User Groups (LUGs) in Deutschland schwanken zwischen 140 und 500 Gruppen.¹⁵ Daneben existieren Unix User Groups und viele auf eine bestimmte Software spezialisierte Gruppen. Beispiele sind eine lokale → *Ubuntu* User Group oder eine → *Typo3* User Group. User Groups und einige andere Treffpunkte sind innerhalb der Szene als verlässliche Orte, an denen zu bestimmten Zeiten Gleichgesinnte anzutreffen sind, wichtig (vgl. Hitzler et al. 2001: 24). Manche LUGs sind als gemeinnützige Vereine institutionalisiert oder an Universitäten angesiedelt, andere sind informelle Treffen oder Stammtische. LUGs treffen sich regelmäßig (wöchentlich, 14-tägig oder monatlich) und bieten ein Programm an. Programmpunkte sind oft Vorträge und so genannte „Install Parties“, auf denen Menschen bei der Linux-Installation auf ihrem Rechner geholfen wird. Die meisten Treffen sind für den Austausch über Fragen und Probleme aus der alltäglichen Arbeit mit Linux reserviert. Viele LUGs haben eigene Internetauftritte, wo sie auf ihre Mailinglisten oder Chatkanäle hinweisen. Dort ermutigen sie oft explizit Neulinge, mit ihren Fragen auf die Liste oder zu den Treffen zu kommen:

„Es gibt keine Aufnahmeprüfung; niemand sollte aus Angst, nicht genügend Ahnung vom Thema zu haben, keine intelligenten Fragen stellen zu können oder einfach nicht → *Geek* genug zu sein, vom Stammtisch fernbleiben.“¹⁶

„Alles in allem soll es aber auf keinen Fall ein ‚Freakverein‘ werden, sondern die CLUG ist explizit auch dafür da, um nicht so versierten LINUX-Anwendern Hilfen und Anregungen zu geben.“¹⁷

¹⁵vgl. <http://linuxwiki.de/LinuxUserGroup>, <http://www.pro-linux.de/lugs/> und <http://www.linux.de/groups/> (Abruf: jeweils 29.07.2009), vgl. auch die Liste in der monatlich erscheinenden Zeitschrift *Linux-User*.

¹⁶Website der LUG Jena (<http://www.lug-jena.de/>, Abruf: 29.07.2009)

¹⁷Website der LUG Chemnitz (CLUG) (<http://www.clug.de/uber-uns/>, Abruf: 29.07.2009)

Dabei schreiben mehrere Gruppen, der erste Schritte, um sich der LUG anzuschließen, sei es, die Mailingliste zu abonnieren. Das persönliche Kennenlernen komme danach (vgl. ebenfalls bzgl. Softwareprojekten von Krogh et al. 2003: 11).

„Ich will mitmachen, wie?

Du abonnierst am besten erstmal die Mailingliste und kommst dann zu unseren Treffen, um uns auch persönlich kennen zu lernen. Des Weiteren betreiben wir auch das BraLUG → *Wiki*, bei dem auch Du zum Inhalt beitragen kannst.“¹⁸

Zu Linux User Groups liegt sehr wenig Forschung vor. Richard Bagozzi und Utpal Dholakia werfen die Frage auf, warum die Beteiligten zu der unbezahlten Assistenz und Beratung bereit sind, obwohl das zeitaufwändige, alltägliche und wenig prestigeträchtige Aufgaben seien (vgl. Bagozzi/Dholakia 2006: 1100). Der für diese Arbeit interessanteste Punkt der Autoren ist, dass sie LUGs als einflussreiche *soziale* Zusammenhänge bezeichnen (vgl. ebd. 1110), in denen es den Beteiligten immer stärker um die sozialen Beziehungen geht, je länger sie dabei sind. Neulinge kämen oft mit konkreten technischen Problem zur LUG, die langjährigen Mitglieder kämen auch wegen der Gruppe selbst und hülften Neulingen, weil es ihnen Spaß macht *und* weil sie die neuen Linux-Nutzer_innen gerne kennen lernen (vgl. Bagozzi/Dholakia 2006: 1112). In Anbetracht des verbreiteten Klischees von ‚sozial inkompetenten Geeks‘¹⁹ ist es jedoch interessant, dass die Befragten die sozialen Aspekte von LUGs hier stark hervorgehoben haben.²⁰ Aus meiner Perspektive entsteht bei Bagozzi/Dholakia dennoch ein verzerrtes Bild von LUGs, weil sie die Relevanz

¹⁸Website der Brandenburger Linux User Group (BraLUG) (<http://www.bralug.de/>, Abruf: 29.07.2009), vgl. auch die Website der Bonner Linux User Group (<http://bolug.uni-bonn.de/about/#wiekannichmitmachen>, Abruf: 29.07.2009)

¹⁹Diese behauptete soziale Inkompetenz findet sich einerseits in Teilen der Literatur (vgl. Kelty 2008b: 22). Bereits 1977 beschrieb Joseph Weizenbaum „zwanghafte Programmierer“ als ungepflegte Männer, die sich schlecht ernähren, 20-Stunden-Schichten arbeiten und am liebsten neben ihrer Maschine schlafen (Weizenbaum 1977: 160 ff.). Auch das bekannte *Portrait of J. Random Hacker* befördert diese Wahrnehmung innerhalb und außerhalb der Szene (<http://www.catb.org/jargon/html/appendixb.html>). Andererseits fanden sich auch entsprechende Selbstbeschreibungen bei einigen Beobachteten („das schöne ist das man sich über die Technik dann doch was zu sagen hat, was soziale Inkompetenz ein wenig ausgleicht. Beispielsweise gibt es wirklich code der ist wunderschön. Und das kann man den Leuten dann auch sagen“ (Ch2 2009)).

²⁰Zu einem ähnlichen Ergebnis kam Jörg Strübing bei einer Untersuchung über Programmierer_innen bereits 1992. Er bezeichnet eine ausgeprägte Kollektivität als eines der auffälligsten Merkmale von Softwareentwickler_innen, die bei ihren Kolleg_innen sehr viel Wert auf Teamfähigkeit legten. „Die Feststellung der Kollektivität ist insofern von Interesse, als sie die Bedeutsamkeit der Erfahrung von Kooperationsanforderungen in der Programmierarbeit unterstreicht und damit die — vor allem populärwissenschaftlich gerne verbreiteten — ‚Vereinsamungsszenarien‘ für diesen Bereich gesellschaftlicher Arbeit, wenn nicht in das Reich der Fabeln, so zumindest in die Abteilung ‚Exotik‘ verweist“ (Strübing 1992: 155).

des Technologischen nicht registrieren. Der Spaß an der Beschäftigung mit den technischen Fragen *selbst* kommt in ihrer Arbeit nicht vor (vgl. zu Spaß als Erklärungsansatz der Motivation von FLOSS-Entwickler_innen Luthiger Stoll (2006), der sich jedoch nur auf Programmierende bezieht).

Yuwei Lin und Marleen Huysman haben die Yorker LUG untersucht. Sie konzentrieren sich jedoch auf die technischen Diskussionen auf der LUG-Mailingliste und stellen LUGs als Plattformen dar, die *in erster Linie* online kommunizieren (vgl. Lin 2004a; Huysman/-Lin 2005: 56 f.). Lin stellt das Potenzial der online archivierten Mailingliste heraus: Durch diese Kommunikationsform trügen die LUG-Mitglieder nicht nur zur Lösung eines *lokalen* technischen Problems bei, sondern gleichzeitig zum *globalen* Wissensbestand über freie Software. Dies bezeichnet sie als innovative Praxis im Umgang mit Wissen (vgl. Lin 2004a). Charakteristisch für das Lernen in LUGs seien die ständigen (Re-)Definitionen von Problemen, die *trial-and-error*-lastigen Aushandlungsprozesse von Lösungen und die Heterogenität der Gruppen, was z. B. soziale Bildungshintergründe, technischen Wissensstand oder Berufe angeht. Lin/Huysman gehen optimistisch davon aus, dass derartige „Online-Communities“ ohne feste Mitgliedschaft das Lernen und den Wissenstransfer zwischen Menschen aus ganz verschiedenen sozialen Kontexten erleichtern, geben aber auch zu bedenken, dass sie die Nachteile, die aus einer so losen Koordination erwachsen, nicht erheben können (vgl. ebd. 56).

Hier knüpfe ich mit meiner Fragestellung und mit meinen Erhebungsmethoden an: Die teilnehmende Beobachtung mit einem Fokus auf die Herstellung von Differenz beleuchtet genau diese von Lin/Huysman identifizierte Leerstelle. Ich konzentriere mich demnach ausschließlich auf die Offline-Kommunikation von LUGs. (Die Gruppen, in denen ich meine Erhebungen durchgeführt habe, nutzen zwar ebenfalls Mailinglisten und Chatkanäle. Darauf wurde jedoch bei den persönlichen Treffen kaum Bezug genommen.)

2.1.7. Community-Events als Vernetzungstreffen der Szene

Ein wichtiges Element von Szenen sind nach Hitzler et al. „Events“, also Veranstaltungen, bei denen „unterschiedliche Unterhaltungsangebote nach szenetypischen ästhetischen Kriterien“ zusammengestellt werden. Hitzler et al. nennen Events „interaktive Spektakel“ mit dem Anspruch, ein „totales Erlebnis“ zu bieten. „Die zumindest latente Funktion (...) eines Events ist die Aktualisierung, Herstellung und Intensivierung von Wir-Gefühl“ (Hitzler et al. 2001: 26). Bei Community-Events im FLOSS-Bereich wird die Vernetzung

unterschiedlicher Gruppen deutlich und diese Vernetzung wird dort gepflegt. Tatsächlich ist der Begriff „Event“ im Forschungsfeld selbst gängig. Über das Jahr verteilt gibt es allein in Deutschland etliche Termine: Dazu gehören die verschiedenen messeähnlichen Linuxtage, die in der Regel ehrenamtlich von lokalen Gruppen wie LUGs organisiert werden.²¹ Beispiele sind die Chemnitzer Linuxtage, der Dresdner Linux-Infotag, der Brandenburger Linux-Infotag oder der „große“ Linuxtag, der seit 2007 in Berlin stattfindet. Das LUG-Camp ist ein Zeltlager, das jeden Sommer stattfindet. Bekannte Konferenzen sind die *Free and Open Source Software Conference (FrOSCon)* bei Bonn oder das *Free and Open source Software Developers' European Meeting (Fosdem)* in Brüssel. Das Debian-Entwickler_innen-Treffen *debconf* findet jedes Jahr an einem anderen Ort auf der Welt statt. Auch die Veranstaltungen des *Chaos Computer Club* werden von der Community frequentiert, nämlich der *Easterhegg* an Ostern und der *Chaos Communication Congress* zwischen Weihnachten und Neujahr. Teile der Community besuchen auch thematisch spezialisiertere Konferenzen oder Workshops wie die *LinuxAudioConference* oder das *Wireless Community Weekend*, bei dem sich Initiativen treffen, die unkommerzielle Funknetzwerke aufbauen.

Community-Events ziehen meistens internationales Publikum an und haben gewisse Programmpunkte gemeinsam: Es gibt jeweils ein Vortragsprogramm, Workshops zu speziellen Themen und Ausstellerstände von Projekten (Softwareprojekten, Linux-Distributionen, LUG-Stände) und Firmen, die Dienstleistungen oder Software-Lösungen anbieten. Auch die bereits erwähnten politisch engagierten Gruppen treten auf den Events mit Infoständen auf. Jedesmal ist auch ein Stand präsent, an dem Fanartikel mit den Logos oder Maskottchen des Events oder bestimmter Projekte erworben werden können (vgl. Abb. 2.3). Das bekannteste Maskottchen ist der Pinguin *Tux*, der allgemein für Linux steht. Beliebte Fanartikel sind auch T-Shirts, die auf die Bedienung des Computers über die → *Kommandozeile* anspielen (vgl. zur Manifestation von Szenekultur in Codes, Symbolen und Outfits Hitzler et al. 2001: 22, 35). Unter den Ständen sind nicht nur Linuxprojekte, sondern auch andere freie Softwareprojekte vertreten, z. B. verschiedene → *BSD*-Betriebssysteme oder → *OpenSolaris*. Hier zeigt sich, wie fließend die Grenzen der Communities sind (vgl. Hitzler et al. 2001: 28). Die Events sind auch gemeinsame Treffpunkte von FLOSS-Communities und einem Spektrum explizit politischer Gruppen. Ihr Inhalt ist

²¹Der „große“ Linuxtag bildet eine Ausnahme. Er ist inzwischen so groß, dass der Verein Linuxtag e.V. einen großen Teil der Organisation an die kommerziell ausgerichtete Berliner Messe AG abgegeben hat. Das hat wiederum in der Community für Verstimmungen gesorgt, weil die Teilnahme am Event und vor allem am abendlichen „social event“ dadurch kostspielig wurde.

2. Einführung ins Forschungsfeld FLOSS-Communities

nicht in erster Linie freie Software, sondern die Einflussnahme auf Politik in den Bereichen Internet, Datenschutz und Bürgerrechte. Dazu gehören z. B. der *FoeBuD e. V.*, der *AK Vorratsdatenspeicherung* oder die Piratenpartei. Es gibt personelle Überschneidungen zu FLOSS-Communities, zudem nutzen die Mitglieder dieser Gruppen oft ebenfalls freie Technologien.

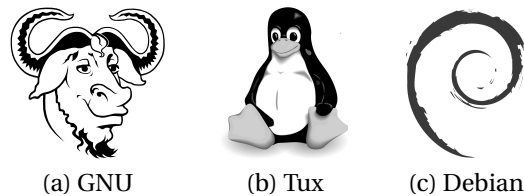


Abb. 2.3.: Maskottchen und Logos in FLOSS-Communities

Ein Programmpunkt auf Events sind Zertifizierungsprüfungen wie die des → *Linux Professional Institute* oder → *Typo3*-Zertifizierungen. Sie können von Veranstaltungsteilnehmer_innen zu günstigeren Preisen abgelegt werden als außerhalb von Linuxtagen. Die Zertifizierung des Wissens über freie Software wird klar mit beruflichen Zielen in Verbindung gebracht (vgl. Heinlein 2009: 15; Maaßen 2009: 15). Ein weiteres wiederkehrendes Element sind → *Keysigning*-Parties. Sie haben etwas mit verschlüsselter Online-Kommunikation zu tun. Bei persönlichen Treffen überprüfen die Beteiligten die Identität ihrer Kommunikationspartner_innen durch eine Passkontrolle und lassen sich die Authentizität ihrer ‚öffentlichen Schlüssel‘ bestätigen, um sicher zu gehen, dass sie online mit den Personen kommunizieren, von denen sie es denken (vgl. die Analyse in Kap. 7).

Auf Community-Events findet abends ein so genanntes ‚social event‘ statt, auf dem gemeinsam gegessen wird und es expliziten Raum für das Kennenlernen gibt. Hier wird die Hierarchisierung innerhalb der Communities besonders deutlich: Je nach Event kann der Zugang zu diesem Teil der Veranstaltung bestimmten Community-Eliten vorbehalten sein. Bei einem bestimmten Event sind nur die Aktiven (die Vortragenden, die Organisator_innen und Helfer_innen) dazu eingeladen. Bei einem anderen Event kostet das ‚social event‘ gesondert Eintritt, der für die auf der Veranstaltung Engagierten vergünstigt ist. Die Szene-Eliten bauen nicht nur die Communities auf, sondern verschaffen sich dadurch auch Privilegien, z. B. in Form von besonderen Zugangsberechtigungen (vgl. Hitzler et al. 2001: 27). Ein weiteres Beispiel sind die elektronischen Eintrittskarten, die beim ‚großen‘ Linuxtag an Szene-Eliten verschickt werden. Wer ein solches e-Ticket hat, bekommt die Möglichkeit, weitere Personen einzuladen. Dies bedeutet, dass diejenigen, die gut vernetzt sind, kostenlosen Zugang zum Event haben. „Rein kommt, wer drin ist“ (Hitzler et

al. 2001: 29). Derart gestaffelte Mitgliedschaften deuten bereits vor einer genauen Datenerhebung darauf hin, dass innerhalb der Szene soziale Unterschiede gemacht werden.

Die soziale Bedeutung von Community-Events fängt Gabriella Coleman in einer ethnografischen Arbeit über „Hacker-Konferenzen“ (in dem Fall die *debconf* der Distribution *Debian*) gut ein (vgl. Coleman 2010). „[T]he conference is culturally significant because it allows hackers to collectively enact, make visible, and subsequently celebrate many elements of their quotidian technological lifeworld (...) all of which unfolds in an emotionally charged setting“ (ebd. 50; vgl. auch Hitzler et al. 2001: 217). Ein solches Community-Event verstärke die Gruppensolidarität; die Beteiligten fänden dort große Wertschätzung und Aufmerksamkeit für ihre Arbeit und schlössen Freundschaften. Das Event reiße sie für einige Tage aus ihrem Alltag — Coleman vergleicht eine Hacker-Konferenz mit einem Sommercamp ohne Regeln, mit einem Urlaub, der intensive und anstrengende Arbeit am Projekt und Schlafmangel mit sich bringe (vgl. ebd. 53). Gleichzeitig gebe es keine Gäste, sondern alle bei der *debconf* Anwesenden seien im Projekt Aktive, die wüssten, dass sie selbst Inhalt und Ergebnis der Konferenz auch spontan mitbestimmen können.

„Because very little beyond talks and a few planned events can be foreshadowed or predicted in advance, the social atmosphere is pregnant with possibility. Time takes on new qualities. Most especially, time in the ordinary (and often annoying) sense of having to keep it, is unimportant as are many other demands of day-to-day living. (...) Attention is given entirely to the present moment, so much so that the totality of the conference is usually recalled as startlingly unique“ (ebd. 53 f.).

Die Anwesenden stimmten implizit der Idee zu, dass virtuelle Kommunikation die Treffen von Angesicht zu Angesicht nicht voll ersetzen kann, da die zwischenmenschlichen Interaktionen der Kern der persönlichen Zusammentreffen seien (ebd. 54). Meine Beobachtungen bestätigen, dass diejenigen, die ein gemeinsames Projekt entwickeln, sich bei Linxtagen zur Weiterarbeit treffen, gerade auch in informellem Rahmen am Abend. Laut Yuwei Lin sind Community-Events auch wichtige Brücken zwischen Communities und Firmen, die im FLOSS-Bereich tätig sind. Gerade weil die Atmosphäre dort meist informeller ist als bei geschäftlichen Treffen, seien die Events so gute Plattformen, um Kontakte mit künftigen Kolleg_innen oder Kund_innen zu knüpfen (vgl. Lin 2004b: 195 ff.). Neben dem offiziellen Programm findet ein für das Community-Leben sehr wichtiges informelles Programm auf den Events statt. Ich werde im Laufe der Arbeit vor allem zwei wiederkehrende Programmpunkte von Community-Events im Hinblick auf soziale Differenzierung untersuchen, nämlich Keysigning-Parties und den Hacking-Contest.

2.2. Der historische Wissensbestand von FLOSS-Communities

Zur Geschichte freier Software gibt es in der Szene ein häufig wiederholtes Narrativ, das die Funktion eines Gründungsmythos hat. Diese Erzählung beginnt in der Regel mit der Entstehung des Internet, also in den späten 1960er Jahren. Im Zentrum des Mythos stehen das *MIT* und die damaligen Studierenden und Forschenden des Instituts. Bei der folgenden Zusammenfassung dieser Erzählung konzentriere ich mich auf communityartige Kooperationsformen, die sich durch diese Geschichte ziehen.

2.2.1. Die Entstehung des Internet

Die Anfänge des Internet-Vorläufers ARPANET gehen auf das US-Verteidigungsministerium zurück. Dort gab es die Forschungsabteilung *Advanced Research Projects Agency (ARPA)*, die maßgeblich daran beteiligt war, den Computer von einer Rechenmaschine zu einem Kommunikationsmedium umzudefinieren (vgl. Grassmuck 2004: 181). Dahinter stand das Ziel, ein dezentrales Kommunikationssystem zur Übermittlung von Informationen aufzubauen, „das gegenüber nuklearen Angriffen unempfindlich sein sollte“ (Castells 2003: 50). Das ARPA gab für die zu entwickelnden technischen Bestandteile dieser Vernetzung Ausschreibungen heraus. Die Zuschläge gingen an Universitäten und Forschungseinrichtungen, aber auch an die Rüstungsindustrie. Dabei spielten wissenschaftliche, institutionelle und persönliche Netzwerke zwischen dem Verteidigungsministerium, verschiedenen Universitäten und der Privatwirtschaft eine entscheidende Rolle (vgl. Castells 2003: 52).²² 1969 wurden die ersten vier Großrechner an verschiedenen US-Universitäten miteinander vernetzt (vgl. Grassmuck 2004: 181 ff.). Zunächst hatten nur Universitäten, die mit dem Verteidigungsministerium kooperierten, Zugang zum ARPANET, doch sie setzten die neue Kommunikationsmöglichkeit auch für ihre sonstige Arbeit ein und trieben damit die Vergrößerung des Netzwerkes für wissenschaftliche Einrichtungen voran.²³ Die Bezeichnung „Internet“ verbreitete sich in den frühen 1980er Jahren (vgl. Castells 2003: 50). Die technischen Standards für das Netz wurden ab 1969 in so genannten → *Request for Comments*-Dokumenten (RFC) diskutiert und entwickelt. Die RFCs wurden zum

²²So war etwa die Firma *Bolt, Beranek and Newman (BBN)* an der technologischen Entwicklung von Bestandteilen beteiligt. Der Leiter des ARPA hatte dort zuvor eine Forschungsgruppe im Computerbereich geleitet (vgl. Grassmuck 2004: 180 ff.).

²³Um die militärische Forschung davon abzuschotten, wurde 1983 das MILNET vom ARPANET abgekoppelt.

wichtigsten Kommunikationskanal der Beteiligten, oft Doktorand_innen, die um Kommentare zu ihren Ideen baten (vgl. Grassmuck 2004: 183). Die Diskussionen über die technischen Entwicklungen für das Netzwerk und die Datenübertragung zwischen Rechnern waren damit um 1970 online zugänglich, es waren jedoch nur 14 Rechner an das Netzwerk angeschlossen. Aus diesem Grund wurden die RFCs zu Beginn noch per Post ausgetauscht.²⁴ Auch die technische Leitung des Internet, die *Internet Engineering Task Force (IETF)*, war ein offenes Forum. Sie war eines der nicht-staatlichen Standardisierungsgremien, die Mitte der 1980er Jahre gegründet wurden. Die Kommunikation lief über rund 100 offene, aufgabenorientierte Mailinglisten, in die sich jede_r eintragen und mitdiskutieren konnte. Arbeitsergebnisse wurden in jedem Entwicklungsstadium veröffentlicht (vgl. Grassmuck 2004: 187). Volker Grassmuck beschreibt hier eine Praxis in der Entwicklung von Internettechnologien, die heute in der Entwicklung freier Software gängig ist. Er nennt die Zeit vom Ende der 1970er bis zum Ende der 1980er Jahre die „wilde Phase“ des Internet:

„eine Tauschökonomie für Software und Information, eine graswurzel-basierte Selbstorganisation, emergierende Communities und der Hacker-Geist, der jede Schließung, jede Beschränkung des Zugangs und des freien Informationsflusses zu umgehen weiß“ (ebd. 180).

Laut Grassmuck hat diese Phase geendet, als 1990 das ARPANET abgeschaltet und 1991 das Werbeverbot für das Internet aufgehoben wurde. Dies habe die Kommerzialisierung des Internet eingeleitet (vgl. ebd. 180, 195).

2.2.2. Die Entstehung des Softwaremarktes

Was Software angeht, so erfolgte die kommerzielle Schließung nicht erst um 1990, sondern bereits in den 1970er Jahren. Software war bis Ende der 1960er Jahre kein eigenständiges Produkt mit einem Markt. Sie wurde der Hardware beim Kauf immer mit → *Quellcode* beigelegt. Die Anwender_innen verfügten zu der Zeit selbstverständlich über Programmierkenntnisse, schrieben an der Software der Hersteller weiter und gaben ihre Ergebnisse an die „Nutzergemeinschaft“ (Grassmuck 2004: 202) zurück. Herstellerfirmen wie *International Business Machines Corporation (IBM)* und *Digital Equipment Corporation (DEC)* förderten die Selbstorganisation der Anwender_innen online in User Groups (vgl.

²⁴vgl. <http://www.heise.de/newsticker/40-Jahre-RFCs--/meldung/135391>, Abruf: 13.08.2009

ebd.) Das Prinzip freier Softwareentwicklung gehe bis in die 1950er Jahre zurück: 1952 habe die Arbeitsgruppe *SHARE* ihre Arbeit aufgenommen, die die Programmierarbeit für den hauptsächlich militärisch genutzten Großrechner *IBM 701* koordiniert habe (vgl. Glott et al. 2007).

1969 entkoppelte IBM unter dem Druck eines Kartellverfahrens die Software von der Hardware. Die anteilig aus der Community stammende Software blieb zwar frei, aber die Voraussetzung für eine eigenständige Softwareindustrie war damit geschaffen. Grassmuck nennt Bill Gates, den Gründer der Firma *Microsoft (MS)*, als einen Vorreiter, der ab Mitte der 1970er Jahre die proprietäre Schließung von Software betrieben habe. Auch am *Labor für Künstliche Intelligenz (KI)* des *MIT* stellten immer mehr Softwareentwickler_innen ihre Produkte unter ein Copyright. Stallmans Kolleg_innen gründeten Firmen und warben die Programmierer_innen von der Universität ab. „An die Stelle des *sharing spirit* war eine Konkurrenz um Marktanteile für dasselbe Produkt getreten“ (Grassmuck 2004: 220). An Universitäten geschriebene Software wurde bis in die 1970er Jahre hinein ebenfalls frei verbreitet, ohne dass von freier Software die Rede war (vgl. ebd. 202).²⁵ Richard Stallman erlebte diese Zeit mit: Er war ab 1971 am *Labor für Künstliche Intelligenz* tätig, das Grassmuck als „legendäres Hackerparadies“ bezeichnet (ebd. 218). Das KI-Labor des *MIT* steht im Zentrum des Gründungsmythos freier Software, weil es nicht so autoritär strukturiert war wie andere Informatikinstitute, was sich in einem unbeschränkten und unbürokratisch organisierten Zugang zu Ressourcen äußerte. Dort gab es in den 1970er Jahren eine Gemeinschaft von Softwareentwickler_innen, die ihr Wissen und ihre Software miteinander teilten und ihre Kreativität auslebten. Auch hier veränderte sich der Umgang damit ab den späten 70er Jahren radikal (vgl. Grassmuck 2004: 217 ff.), doch *dass* es diese Atmosphäre am *MIT* gab, ist ein starker Bezugspunkt für FLOSS-Communities. Sie wird in dem Buch *Hackers* von Steven Levy (2010) 1984 festgehalten — ein Buch, das einen ebenso starken Orientierungspunkt darstellt.

Mitte der 1970er Jahre entstanden auch informelle Szenezusammenhänge, die als Vorläufer von Linux User Groups und Community-Events bezeichnet werden können: Der *Homebrew Computer Club* entstand 1975. Mit dem ersten Heimcomputer, dem Altair 8800, der 1975 auf den Markt kam, sammelten sich begeisterte Anwender_innen im *Homebrew Computer Club* an der Universität Stanford. Das erste, fast zu einem Mythos gewordene

²⁵Zu der Zeit war es auch noch nicht so, dass man eine Software auf einem Rechner entwickeln und sie dann einfach auf einem anderen installieren und nutzen konnte. Die Weitergabe war stärker als heute durch die Hardware beschränkt.

Treffen habe in einer privaten Garage stattgefunden. Der Club habe weniger eine Kultur begründet, sondern vielmehr eine bereits existierende Kultur von kollektiver Innovation und dem Teilen von Wissen über Technologien im Silicon Valley aufgegriffen (vgl. Meyer 2003).

„These were people intensely interested in getting computers into their homes to study, to play with, to create with... and the fact that they would have to build the computers was no deterrent. (...) They discussed what they wanted in a club, and the words people used most were 'cooperation' and 'sharing'. (...) [T]he source of the intellectual heat was not a planner-like contemplation of the social changes possible by mass computing, but the white-hot hacker fascination with technology“ (Levy 2010: 202 f.).

Ähnlich vielen der heutigen Linux User Groups hatte der zehn Jahre existierende Club keine formale Mitgliedschaft, deckte Kosten über Spenden und verstand sich als offen für alle Interessierten. Auch die späteren *Apple*-Gründer Steve Jobs und Steve Wozniak gehörten ihm an. Selbst wenn manchmal kleine Firmen ihre Produkte hier vorgestellt hätten, seien dies informelle Treffen von Computerbegeisterten gewesen (vgl. Coleman 2010: 52). Linxutage als Community-Events haben ebenfalls Vorläufer, die bis in die 1970er Jahren zurückreichen: Bevor es konferenzähnliche Zusammenkünfte gab, waren professionelle Konferenzen und Workshops Treffpunkte. 1976 habe es erste Computermessen in den USA gegeben. Die erste aus der Szene heraus organisierte Hackerkonferenz habe 1984 nördlich von San Francisco stattgefunden und sei von Levys Buch *Hackers* beeinflusst gewesen. Hier und bei den etwa zeitgleich erstmals stattfindenden „SommerCons“ brauchte man eine persönliche Einladung. Die SommerCons wurden vom elektronischen Journal *Phrack* veranstaltet und waren bis 1995 nicht für die Öffentlichkeit offen. Die erste offene Konferenz war laut Coleman die *HoHoCon* 1990, veranstaltet von der texanischen Hackergruppe *Cult of the Dead Cow* und mit Vorbildcharakter für weitere Veranstaltungen. In Deutschland und den Niederlanden gibt es seit den späten 1980er Jahren Camps, die die weltweite Szene anziehen. Um das Jahr 2000 kamen Konferenzen auf, die sich speziell Linux oder freier Software widmeten (Coleman 2010: 52).

Bis in die 1980er Jahre wurden Computer durch sinkende Hardwarepreise zur „Massenware“ und auch die Software fand als eigenständiges Produkt schnell massenweise kommerzielle Verbreitung (vgl. Grassmuck 2004: 203 ff.). Zu Beginn der 1980er gab es fast nur noch proprietäre Software. Bis in die frühen 1990er Jahre gelang es *Microsoft*, andere bis dahin existierende Betriebssystem-Hersteller vom Markt zu verdrängen (vgl. Grassmuck

2004: 203 ff.). Des Weiteren verfolgte die Firma die Strategie, viele unabhängige Programme (z. B. Textverarbeitung und → *Webbrowser*) gleich in das Windows-Betriebssystem zu integrieren. Die Formate, in denen Dateien gespeichert wurden, waren nicht mit denen anderer Hersteller kompatibel (vgl. ebd. 207). Bis heute ist der Konzern immer wieder mit Kartellverfahren konfrontiert.

Neben *Microsofts MS-DOS*, später *Windows* und *Apples Mac OS X* gibt es Dutzende von anderen Betriebssystemen.²⁶ Ich greife hier die Geschichte von Unix kurz auf, weil sie für Linux eine Rolle spielt. Das Betriebssystem Unix wurde ab 1969 von einem Team um Ken Thompson geplant, der für die *Bell Laboratories* von *AT&T* arbeitete. In Unix flossen Erfahrungen aus dem Vorgängerprojekt *Multics* ein, an dem Thompson zuvor mitgearbeitet hatte. Er und einige Andere schrieben Unix ab 1971 neu, um geänderten technischen Anforderungen gerecht zu werden. Da *AT&T* in den USA ein staatliches Telefonmonopol hatte, durfte der Konzern nicht in anderen Bereichen wie der Softwareindustrie tätig werden. Unix wurde deshalb nicht vermarktet, sondern zum Selbstkostenpreis herausgegeben. „Da *AT&T* für ihr Unix keinerlei Support anbot, blieb den Nutzern gar nichts anderes übrig, als sich zu einer Community zusammenzufinden“ (Grassmuck 2004: 212). Um den Informationsaustausch über das ARPANET zu erleichtern, entwickelte der Doktorand Tom Truscott 1979 die Software für das *Usenet (Unix User Network)*, ein Netzwerk für Diskussionsforen.²⁷ Das *Usenet* wurde das Medium, „in dem die Mitglieder der Unix-Community sich gegenseitig unterstützten“ (Grassmuck 2004: 213).

An Universitäten war Unix ein sehr verbreitetes Betriebssystem. An der kalifornischen Universität Berkeley stellte der Doktorand Bill Joy 1977 aus Unix und einigen an der Universität entstandenen Erweiterungen die so genannte *Berkeley Software Distribution (BSD)* zusammen. Sie wurde frei verteilt und erschien in mehreren Versionen (vgl. Grassmuck 2004: 214 f.).

2.2.3. Kämpfe um die „Freiheit“ von Software

In den USA war Software seit 1981 patentierbar.²⁸ Zu Beginn der 1980er Jahre gab es dann bezüglich *BSD* eine rechtliche Einigung zwischen Berkeley und *AT&T*. Sie sah vor, dass für

²⁶vgl. zum Überblick http://de.wikipedia.org/wiki/Vergleich_von_Betriebssystemen und http://de.wikipedia.org/wiki/Liste_der_Betriebssysteme

²⁷Im *Usenet* gab es von Anfang an nicht nur Diskussionen über Unix, sondern über alle möglichen technischen und nichttechnischen Themen.

²⁸Bis dahin galt Software als nicht schützbares mathematische Formeln oder Algorithmen (vgl. ebd. 221).

den Quellcode-Anteil, der von dem Telefonmonopolisten stammte, eine Lizenzgebühr an AT&T entrichtet werden musste. Hier begann der Konzern mit der allmählichen Vermarktung von Unix. Als er 1984 aus kartellrechtlichen Gründen zerschlagen wurde, gründete er die Tochterfirma *Unix System Laboratories (USL)*, die Unix vermarktete und den Namen schützen ließ. Die für BSD zu entrichtenden Lizenzen wurden in den Folgejahren so teuer, dass ein Entwickler in Berkeley 1988 begann, das Betriebssystem von den Quellcode-Anteilen zu „befreien“, die auf AT&T zurückgingen. Es dauerte einige Jahre und Rechtsstreits, bis diese BSD-Version 1994 veröffentlicht wurde (vgl. ebd. 217).

1984, im Jahr der kommerziellen Schließung von Unix durch *USL*, nahm Richard Stallman am *MIT* die Arbeit am → *GNU*-Projekt auf. Sein Ziel war es, ein Unix ebenbürtiges Betriebssystem zu schreiben, das jedoch keinen geschützten Quellcode enthalten sollte. Stallman unterlegte seine Aktivität mit dem bereits erwähnten GNU-Manifesto: „Software sellers want to divide the users and conquer them, making each user agree not to share with others. I refuse to break solidarity with other users in this way“ (Stallman 1985). „Die Wahl fiel auf Unix [als Vorbild für GNU] (...), weil es sich bewährt hatte (...) und weil es bereits eine aktive weltweite Unix-Gemeinde gab, die durch seine Kompatibilität leicht zu GNU wechseln konnte“ (Grassmuck 2004: 222). Richard Stallman kündigte seine Stelle am *MIT*, damit sein Arbeitgeber keine Rechte an *GNU* geltend machen könne. Sein ehemaliger Vorgesetzter lud ihn jedoch ein, das Labor weiterhin zu benutzen. Die ersten Programme wurden sowohl kostenlos über das Internet verteilt als auch gegen Geld auf Bändern verschickt, da Stallman kein Einkommen mehr hatte. Ab 1985 verkaufte die *Free Software Foundation (FSF)* die Datenträger, verwaltete Spenden und brachte Bücher heraus, um von dem Ertrag einige Softwareentwickler_innen zu bezahlen (vgl. ebd. 223).²⁹ Stallman hat mit *GNU* und freier Software also nichts komplett Neues erfunden. Er hat sich vielmehr gegen die kommerzielle Entwicklung gestellt, die die Softwarekultur zerstört hatte, die bis etwa Mitte der 1970er Jahre verbreitet war. Grassmuck führt diese Kommerzialisierung auf Kartellverfahren des US-Justizministeriums und auf andere politische Entscheidungen wie die Patentierbarkeit von Software und die Werbeerlaubnis im Internet zurück. Er stellt implizit einen Zusammenhang her: Es habe in den USA lange Zeit Nischen gegeben, in denen Software communityartig entwickelt wurde, die z. B. durch die monopolistischen oder monopolähnlichen Marktpositionen von AT&T oder IBM Bestand hatten. Diese Nischen seien durch politische Regulierung abgeschafft worden, die

²⁹Die → *GNU General Public License* erlaubt es, für Dienstleistungen im Zusammenhang mit freier Software Geld zu nehmen. Darunter fallen etwa technischer Support, der Vertrieb von Installationsmedien und Handbüchern. Nur die Software selbst ist davon ausgenommen.

die Entwicklung von Software zu einem proprietären und kommerziellen Produkt vorantrieb. Die vormalige Selbstverständlichkeit, Software frei weiterzugeben, habe Stallman mit dem *GNU*-Projekt und der *GNU GPL* expliziert und zu einer ethischen und politischen Frage gemacht (vgl. Grassmuck 2004: 226).

1991 wurde der finnische Student Linus Torvalds im *Usenet* bekannt, als er von dem Betriebssystemkern berichtete, an dem er arbeitete. Um ihn herum bildete sich auf seine Einladung zur Mitarbeit hin schnell eine Community, die auf die *GNU*-Werkzeuge zurückgriff. Im Nachhinein wird Torvalds' Leistung oft als das fehlende Puzzleteil beschrieben, das die vielen *GNU*-Programme unter einen Hut brachte: Der Linux- \rightarrow *Kernel* stellte die bis dahin fehlende Grundlage des Betriebssystems dar (vgl. Moody 2002: 14). Ab 1998 spaltete die Szene sich, wie oben beschrieben, in die Verfechter_innen von freier und von Open Source-Software.

2.2.4. Kurze Reflexion

Der historische Wissensbestand der Szene enthält viele Bezüge auf Praktiken der Kooperation jenseits von kommerziellem Interesse, die älter sind als die Idee freier Software. Die Entstehung freier Software wird als Reaktion darauf gedeutet, dass Software zu einem kommerziellen und dadurch vereinnahmten Produkt gemacht wurde. In Abgrenzung von kostenloser Software und von Open Source-Software, die zwar den Quellcode offenlegt, aber nicht notwendig auch das Recht zur Veränderung und Weitergabe einräumt, ist mit „frei“ die Offenheit von Software im Sinne der *GNU GPL* gemeint: Transparenz und explizite Aufforderung, die Software selbst zu verbessern und weiterzugeben, ermöglichen den selbstbestimmten Umgang mit ihr.

Obwohl ich hier nur den Haupterzählstrang wiedergegeben habe, blendet dieser Wissensbestand auch bestimmte Fragen oder Akteur_innen komplett aus. Alternative Erzählungen liegen jedoch bereits vor (vgl. Hoffmann 1987; Hunger 2002; Wajcman 1994: 166 ff.), so dass ich hier lediglich ein paar Auslassungen erwähne: Dass die frühen Akteur_innen in der Erzählung der Szene alle über eine Anbindung an Universitäten oder Militär verfügen mussten, um überhaupt Zugang zu Computern zu haben, wird nicht als soziale Bedingung für Teilhabe thematisiert. Das gleiche gilt für Militär und Rüstungsindustrie als Voraussetzung für diese technologischen Entwicklungen. Diese Institutionen sind in der Erzählung einfach da, ihre politischen Ziele bleiben außen vor, ebenso wie die Zugangshürden, die durch sie entstanden. Statt diesem institutionellen Kontext stehen männliche,

technisch versierte, kreative und vernetzte Individuen im Mittelpunkt, die besonders im Fall der Kristallisationsfiguren in der Szeneliteratur als Helden vermittelt werden. Frauen kommen in der Erzählung der Szene weder als technikaffine Akteur_innen, noch als Autor_innen vor. Ute Hoffmann arbeitet ihre Anteile an der Geschichte der Computerentwicklung auf. Sie waren z. B. diejenigen, die im Zweiten Weltkrieg den ersten elektronischen Universalrechner *Electronic Numerical Integrator and Computer (ENIAC)* programmierten. Sie stellten ballistische Berechnungen im Auftrag der US-Armee an. Hoffmann sieht dieses Betätigungsfeld in der damaligen Zeit als berufliche Chance für mathematisch oder naturwissenschaftlich qualifizierte Frauen (vgl. Hoffmann 1987: 87 ff.).

Schließlich bleiben auch die Produktionsbedingungen weitestgehend außen vor, unter denen Computer hergestellt werden. In Levys Buch *Hackers* gibt es eine kurze diesbezügliche Passage: Einem der Gründungsmitglieder des *Homebrew Computer Club* sei bewusst gewesen, dass die Chips in Malaysia und anderen asiatischen Ländern unter harten Arbeitsbedingungen von Frauen hergestellt wurden. Er habe dies in der Gruppe jedoch nicht angesprochen, weil er nicht das Gefühl gehabt habe, der Club gehe auf solche Fragen ein (vgl. Levy 2010: 217 f.). Die Darstellungsweise ist jedoch höchst problematisch: „He heard how the Asian women were paid pitiful wages, worked in unsafe factories, and were unable to return to their villages, since they never had a chance to learn the traditional modes of cooking or raising a family“ (ebd. 217). Aus der Kritik wird hier somit keine Forderung nach besseren Arbeitsbedingungen oder nach einem Umdenken im globalen Norden abgeleitet. Die Lage der Frauen sei vielmehr ausweglos, weil sie ihre „angestammten“ Plätze in der Reproduktionsarbeit nicht ausfüllen könnten. Ob dies auf Levy oder Fred Moore, über den Levy schreibt, zurückgeht, ist nicht nachvollziehbar.

3. Theoretischer Zugang: Einführung in Bourdieus Praxistheorie

Diese Arbeit ist induktiv angelegt: Ich nähere mich meinem Forschungsgegenstand, den FLOSS-Communities, ohne fertiges Kategorienraster, sondern arbeite aus dem Material die Relevanzen des Forschungsfeldes heraus. Damit behaupte ich nicht, theoriefrei zu forschen, denn das ist eine Illusion: Bereits die Forschungsfragen wurden durch Theorien mitgeformt und ihre Bearbeitung bedarf einer theoretischen Rahmung. Ziel des Kapitels ist es, die Forschungsfragen theoretisch inspiriert zu reformulieren, um Anhaltspunkte für die Auswertung des Datenmaterials zu erhalten.

Mein theoretischer Hauptbezugspunkt sind die Konzepte des französischen Soziologen und Ethnologen Pierre Bourdieu. Er wird erstens oft in die soziale Ungleichheitsforschung eingeordnet (vgl. z. B. Hradil 1999: 138 ff.; Burzan 2007: 127-139), die die Verteilung von Teilhabe am gesellschaftlichen Leben und an relevanten Ressourcen erforscht. Moderne Ansätze innerhalb der sozialen Ungleichheitsforschung gehen davon aus, dass soziale Ungleichheit von der jeweiligen Gesellschaft selbst hervorgebracht wird. Eine Erforschung ihrer Ursachen, ihrer spezifischen Ausprägungen und ihrer Auswirkungen auf die Legitimation von Macht- und Herrschaftsverhältnissen soll damit auch die Veränderung von sozialer Ungleichheit ermöglichen (vgl. Burzan 2007). In Bezug auf soziale Chancengleichheit ist Bourdieu ein Skeptiker. Zweitens ist Pierre Bourdieu Praxistheoretiker (für einen Überblick über praxeologische Ansätze vgl. z. B. Reckwitz 2003). Praxistheoretische Ansätze nehmen in Bezug auf die Erforschung von sozialem Handeln eine bestimmte Perspektive ein: Sie teilen nicht nur die Annahme, dass das Handeln von Akteur_innen grundsätzlich durch kollektiv geteilte Wissensordnungen, Symbolsysteme und Sinnhorizonte strukturiert wird. Vielmehr richten sie ihr Forschungsinteresse auf „soziale Praktiken“ als den Ort des Sozialen. Kollektiv geteilte Wissensordnungen sind damit ein praktisches Wissen, das alltäglich eingesetzt und von den Beteiligten verstanden wird. Praxistheoretische Ansätze betonen, dass soziales Leben und Handeln aus vielen, einzeln benennbaren, aber miteinander verbundenen Praktiken besteht (vgl. Reckwitz 2003: 289). Meier argumen-

tiert, dass Bourdieus Theorie keine „Theorie sozialer Praktiken“ unter vielen anderen sei, weil er die Entstehungs- und Anwendungsbedingungen von Habitus, die Erwerbsmodalitäten von Kapital in einem Feld betrachte. Ihm gehe es nicht lediglich um die „Performativität des Handels“, also Praktiken als routinisierte körperliche Verhaltensmuster, sondern um die Rückkopplung von Praxis an soziale Strukturen (vgl. Meier 2004: 57 f.).

Ich verwende in dieser Arbeit Bourdieus Feldtheorie. Unter einem Feld versteht er einen relativ homogenen Teilbereich der Gesellschaft. Ein Feld wird dadurch konstituiert, dass die Akteur_innen aufgrund ähnlicher Existenzbedingungen eine gemeinsame Praxis, einen gemeinsamen Habitus herausbilden. Die Praxis und die historisch gewachsenen Strukturen des Feldes, etwa Regeln und Institutionen, bedingen einander gegenseitig (vgl. die detailliertere Erläuterung der Begrifflichkeiten in 3.1 und 3.2). Die Feldtheorie ermöglicht ethnografisches Arbeiten, weil sie jedem Feld seine Besonderheiten einräumt, hat Bourdieu sie doch aus seiner eigenen empirischen Forschung heraus entwickelt. Seine Oberbegriffe sind auf viele Felder übertragbar, sie sind jedoch jeweils zu hinterfragen und feldspezifisch zu füllen und zu ergänzen. Damit geben sie keine inhaltliche Strukturierung für die Beobachtung vor, sondern bieten ihr konzeptionelle Eckpunkte an. Diese Eckpunkte sind die Begriffe Feld, Habitus, Distinktion und Akkumulation verschiedener Kapitalformen. Mit ihnen sensibilisiert Bourdieu für subtile Formen der Hierarchiebildung in der Praxis. Mit diesen Eckpunkten kann ich mein Erkenntnisinteresse an der Herstellung von Differenz in FLOSS-Communities theoretisch einfassen. Zudem teile ich Bourdieus grundsätzliche Skepsis, was Chancengleichheit oder Gleichberechtigung angeht. Durch die Offenheit für die jeweilige feldspezifische Praxis lässt das Konzept Raum dafür, die Involviertheit von Technik in alltägliche soziale Praxis zu berücksichtigen, was mein Forschungsfeld erfordert, ohne dass sie vorgibt, wie diese auszudeuten ist. Bourdieu sieht schließlich die spezifischen Funktionsweisen gesellschaftlicher Teilbereiche in einem größeren sozialen, ökonomischen und politischen Kontext. Bourdieu ist kein Machttheoretiker im engeren Sinne, sondern Empiriker, seine Analyseperspektive und Begrifflichkeiten bieten jedoch explizit Raum für machtkritische Fragen. Er hantiert nicht mit einem klaren Machtbegriff, sondern liest Macht(unterschiede) etwas diffus als Homologie der Ressourcenverteilung in einem Feld.

Wie Bourdieu in seiner Praxistheorie stelle ich die soziologische Mikroebene in den Mittelpunkt der Arbeit. Um Wechselwirkungen zwischen beobachteter Praxis und gesellschaftlichen Strukturen zu explizieren, schlagen Nina Degele und Gabriele Winker eine kon-

krete Herangehensweise für empirische Arbeiten vor.¹ Auf dieser Ebene bestimmen die Beobachteten, in diesem Fall die Community-Mitglieder, ihr Verhältnis zu sich selbst, zu anderen Menschen und Community-Mitgliedern und zu Computern (vgl. Degele/Winker 2009: 63 ff.). Für die Analyse der Mikroebene empfehlen Degele/Winker, möglichst offen nach Differenzierungskategorien zu suchen, die das Forschungsmaterial enthält.

Von dieser Mikroebene der Praxis unterscheide ich zwei andere Ebenen, die nicht im Mittelpunkt meines Interesses stehen, nämlich erstens die Makroebene der gesellschaftlichen Strukturen und Institutionen und zweitens die Ebene der Repräsentationen, Diskurse und kulturellen Symbole (vgl. Harding 1991). Das Beobachtungsmaterial über Linux-Nutzer_innen enthält Verweise, die über die konkreten Situationen hinausweisen. Die Praxis steht immer auch in Wechselwirkung mit gesamtgesellschaftlichen *und* communityspezifischen Strukturen und Repräsentationen: Eine gesellschaftliche Norm, etwa der Stellenwert von Leistung, schafft beispielsweise auch in FLOSS-Communities normative Rahmen, z. B. in Form von sozialer Anerkennung für Leistung. Die Etablierung von Linux-Zertifikaten als Nachweis von Fähigkeiten steht in Zusammenhang mit der Relevanz institutionalisierter Bildungsabschlüsse in der Gesellschaft und auf dem Arbeitsmarkt (vgl. analog dazu Bourdieu 1983: 190).² Ebenso strukturiert die Geschichte der Computertechnologien die situative Praxis, z. B. wenn die Kenntnis von Konventionen oder textbasierten Programmen vorausgesetzt wird. Für eine adäquate Untersuchung der Wechselwirkungen zwischen diesen drei Ebenen reicht es nicht aus, sich ausschließlich auf empirisches Material zu beziehen. An den entsprechenden Stellen werden deshalb existierende Forschungsergebnisse hinzugezogen, die angrenzende Themen auf den anderen beiden Ebenen behandeln: Dies werden zum einen empirische Ergebnisse aus der FLOSS-Forschung sein. Zum anderen greife ich auf theoretische Arbeiten aus der Techniksoziologie, der Bildungsforschung, der feministischen (Technik-)Forschung und der Akteur-Netzwerktheorie zurück. Es gilt dann herauszufinden, „ob und wie die konkreten sozialen

¹Ihr Anliegen ist eine intersektionale Mehrebenenanalyse, mit der sie die Verschränkungen verschiedener „ungleichheitsgenerierender Dimensionen“ (Degele/Winker 2007) verstehen möchten. In intersektionalen Analysen wird meistens das Zusammenwirken von Diskriminierungsachsen entlang der Kategorien *race*, *class* und *gender* untersucht, teilweise jedoch auch weitere Kategorien (vgl. z. B. Davis 2008; Degele/Winker 2009: 11 ff.). Im Rahmen meiner Arbeit leiste ich keine intersektionale Analyse. Intersektionale Arbeiten hantieren in der Regel mit Kategorien, die Personen quasi „körperlich angeheftet“ werden, was in meinem Feld nicht unbedingt der Fall ist. Zuschreibungen erfolgen viel situativer anhand diverser Merkmale, so dass es mir nicht möglich erschien, klare und wiederkehrende Verknüpfungen aufzuzeigen.

²Zur Präzisierung möglicher Wechselwirkungen zwischen Praxis, Identitäten, Institutionen, kulturellen Symbolen und Diskursen vgl. Winker/Degele 2009: 74 ff..

Praxen der untersuchten Personen auf die [Differenzierungskategorien, S. M.] und damit verbundene Herrschaftsverhältnisse verweisen“ (Degele/Winker 2007: 13).

3.1. Eckpunkte für die Konzeptionalisierung des Forschungsfeldes

In diesem Abschnitt geht es um Bourdieus Perspektive auf gesellschaftliche Strukturen, ihre Wirkungen in Feldern und auf die sozialen Positionen von Akteur_innen. Die Praktiken der Akteur_innen auf der Mikroebene lassen sich nicht getrennt davon untersuchen. Schwingel spricht von einer komplementären Beziehung von Feld und Habitus: Die objektiven Strukturen sozialer Felder und die Habitusstrukturen der Akteur_innen seien die zwei Seiten des Sozialen (vgl. Schwingel 2000: 74.; vgl. auch Burzan 2007: 132).

3.1.1. Bourdieus Feldbegriff

Als Feld im Bourdieu'schen Sinne sind FLOSS-Communities ein Teilbereich der Gesellschaft, ein „Mikrokosmos im sozialen Makrokosmos“ (Bourdieu 2000: 22).³ Das Feld ist der strukturierte (und eben nicht neutrale) Raum, in dem die Praxisformen stattfinden (vgl. Schwingel 2000: 80). Jedes Feld hat seine spezifische Geschichte, die seine Funktionsweisen und Institutionen hervorgebracht hat (vgl. Fröhlich 1994: 41). Diese Historizität verleihe den Strukturen des Feldes ein Eigenleben: Strukturen existierten zwar erst durch die Praxis, sie erlegten den Akteur_innen aber Zwänge für ihr Handeln auf, z. B. in Form von Regeln. Regeln definierten, was im jeweiligen Feld überhaupt möglich und erlaubt sei. Sie regulierten jedoch nicht nur das Handeln im Feld, sondern seien konstitutiv für das Feld. Sie seien selten explizit, sondern würden von Akteur_innen in der Praxis einfach befolgt (vgl. Schwingel 2000: 80f.).⁴

³Ich bleibe hier beim Begriff Feld, obwohl Bourdieu stellenweise auch von „Subfeldern“ spricht. So könnten Linux-Communities als ein Subfeld eines größeren technologischen Feldes begriffen werden. Dieses technologische Feld umfasste dann die gesamte Technologieentwicklung und -nutzung, von Haushaltsgeräten bis hin zu Informations- und Kommunikationstechnologien, die entsprechenden Fachdisziplinen etc.

⁴Ein Beispiel aus der Technik: Auf allen Linuxrechnern finden Nutzer_innen sehr ähnliche Dateibäume vor. Sie sind eine Struktur von Verzeichnissen, in denen die Dateien des Betriebssystems und die der Nutzer_innen gespeichert sind. Die Namen und Funktionen der Verzeichnisse werden in der Szene und in Fachbüchern als „historisch gewachsen“ bezeichnet (vgl. z. B. Ganten/Alex 2004: 462). Der Dateibaum beruht allerdings auf einer *explizit* erarbeiteten Konvention, auf dem *Filesystem Hierarchy Standard (FHS)*. Er legt fest, in welchen Verzeichnissen welche Dateien des Betriebssystems abgelegt werden. Er macht z. B. Vorgaben dafür, wo Konfigurationsdateien und wo von Administrator_innen geschriebe-

Felder sind immer auch Räume, in denen um soziale Vorteile und Macht gerungen wird (vgl. Bourdieu 1985: 74; Burchardt 2004: 111). Durch diese Kämpfe „zwischen den Herrschenden und den Anwärtern auf die Herrschaft“ (Bourdieu 1993: 107) sind sie dynamisch. Die Ressourcen, um die gekämpft wird, sind die im jeweiligen Feld und Kontext relevanten Formen von Kapital (vgl. 3.1.2). Die in FLOSS-Communities relevanten Kapitalformen herauszuarbeiten, ist Teil dieser Arbeit. Die Akteur_innen sind in diese Kämpfe involviert, weil sie einen grundsätzlichen Glauben an die Relevanz dessen, was sie tun, teilen. Diesen Glauben nennt Bourdieu die *illusio*.

„Jedes soziale Feld (...) bringt es dahin, daß die Personen, die sich in es hineinbegeben, jenes Verhältnis zum Feld entwickeln, das ich *illusio* nenne“ (Bourdieu 1998b: 142). „*Illusio* bezeichnet die Tatsache, daß man vom Spiel erfaßt, vom Spiel gefangen ist, daß man glaubt, daß das Spiel den Einsatz wert ist oder (...) daß sich das Spielen lohnt“ (Bourdieu 1998b: 140f.).

Die *illusio* komme wiederum dadurch zustande, dass die objektiven Strukturen des Feldes und die mentalen Strukturen der Beteiligten (ihre Denk-, Wahrnehmungs- und Handlungsmuster) übereinstimmen, in diesem Fall also die Strukturen der Communities, die Habitus der Linux-Nutzer_innen und ihre mentale Involviertheit in diesen Zusammenhang. Die *illusio* sei aber nicht nur Produkt von Feldern, sondern ebenso ihre Voraussetzung. Sie wirke dahingehend, dass denen, die in die Logik und Praxis des Feldes involviert seien, das Geschehen selbstverständlich vorkomme und sie die Regeln nicht hinterfragten (vgl. Bourdieu 1998b: 141).⁵

ne lokale Skripte gespeichert werden sollten. Die Konvention muss nicht in jedem Fall berücksichtigt werden. Sie wird jedoch inkorporiert und reguliert das Handeln der Nutzer_innen: Sie erleichtert nicht nur die Orientierung auf einem Linuxrechner, sondern Konfigurationsdateien werden von Programmen auch nur dann eingelesen, wenn sie am konformen Ort gespeichert sind. Das Wissen um den FHS als technologische Struktur kann zum Kanon der Selbstverständlichkeiten gezählt werden. Neben anderen Strukturen, institutionellen z. B., gibt es in Linux-Communities zahlreiche solcher technologischen Strukturen, die das Feld mitkonstituieren.

⁵Ein Beispiel, an dem sich zeigen lässt, dass viele Community-Mitglieder eine *illusio* teilen, sind die in Kapitel 7 analysierten Keysigning-Parties. Nur wer den Sinn für das Spiel inkorporiert hat, kann die Relevanz des Rituals Keysigning-Party verstehen. Wer nicht versteht, warum Dutzende von Community-Mitgliedern sich mehrmals im Jahr treffen, sich ihre Ausweise vorzeigen und sich bestätigen, dass sie die realen Personen hinter ihren Online-Identitäten sind, teilt die *illusio* nicht, bleibt dem „Spiel“ gegenüber indifferent. Das sind in diesem Fall auch Teile der Communities, nicht nur Außenstehende.

3.1.2. Die Reproduktion von Feldern über die Kapitalverteilung

Über die sozialen Positionen der Akteur_innen eines Feldes bestimmt die Verteilung von Kapital im Feld (vgl. Fröhlich 1994: 41). Dieser Abschnitt gibt einen Überblick über Kapitalformen, die Bourdieu in *seinen* Analysen herausgearbeitet hat, um mit der Empirie die Frage zu beantworten, welche Kapitalformen das Feld FLOSS-Communities strukturieren und wie sie verteilt sind. Kapitalverteilung und Machtbeziehungen können nur zusammen gedacht werden: Für Bourdieu gründen Machtverhältnisse auf der ungleichen Verteilung von Kapital in einem sozialen Raum; Kapital werde gerade deshalb so bedeutsam, weil es ungleich verteilt sei (vgl. Bourdieu 1983: 188). Es könne ganz verschiedene Formen haben: Ökonomisches Kapital, etwa Geld oder ein Kapital, das sich direkt in Geld umwandeln lässt, werde aufgrund der „kapitalistischen ökonomischen Praxis“ als die dominierende Kapitalform wahrgenommen, obwohl andere Formen von Kapital und Profitmaximierung in Beziehungen sozialen Austausches ebenso relevant seien (Bourdieu 1983: 184). Welche Kapitalformen im untersuchten Feld wichtig werden, wie sie umkämpft werden und wie sie weitergegeben werden, hänge vom Feld ab.

„Eine allgemeine ökonomische Praxiswissenschaft muß sich deshalb bemühen, das Kapital und den Profit in allen ihren Erscheinungsformen zu erfassen und die Gesetze zu bestimmen, nach denen die verschiedenen Arten von Kapital (oder, was auf dasselbe herauskommt, die verschiedenen Arten von Macht) gegenseitig ineinander transformiert werden“ (ebd.).

Neben dem ökonomischen Kapital sind das soziale und das kulturelle Kapital die zentralen Kapitalformen bei Bourdieu. Soziales Kapital sind soziale Wirkungen, die sich nicht allein auf individuelle Eigenschaften Einzelner zurückführen lassen. Mit sozialem Kapital meint er nicht einfach Beziehungen, sondern *alle tatsächlich und möglicherweise mobilisierbaren Ressourcen*, die ein dauerhaftes Netz von Beziehungen und die Zugehörigkeit zu Gruppen bieten (vgl. Bourdieu 1983: 190f.). Er hat beobachtet, dass Individuen, die mit einem ähnlichen ökonomischen oder kulturellen Kapital ausgestattet sind, in einem sozialen Raum dennoch sehr Unterschiedliches erreichen können,

„und zwar je nach dem, inwieweit sie in der Lage sind, das Kapital einer mehr oder weniger institutionalisierten und kapitalkräftigen Gruppe (Familie, Ehemalige einer ‚Elite‘-Schule, vornehmer Club, Adel usw.) stellvertretend für sich zu mobilisieren“ (ebd. 191).

Soziales Kapital werde ständig produziert und gestärkt: Akteur_innen verfolgten mehr oder weniger bewusst Strategien zur Schaffung eines Beziehungsnetzes, das ihnen zu irgendeinem Zeitpunkt nützlich werden könnte. Solche Investitionen seien nötig, um dauerhafte und tatsächlich mobilisierbare Beziehungsnetze aufzubauen und erfolgten weniger berechnend als vielmehr aufgrund von inkorporierten sozialen und mentalen Dispositionen (vgl. Bourdieu 1998b: 168). Die Institutionen des Feldes lieferten dabei Unterstützung durch Anlässe, auf denen die Akteur_innen sich begegnen; dazu zählen in meinem Feld die untersuchten LUG-Treffen und Community-Events. Solche Institutionen und Anlässe ermöglichen „das Zusammentreffen von Individuen, die im Hinblick auf alle für das Leben und Überleben der Gruppe wichtigen Gesichtspunkte so homogen wie möglich sind“ (Bourdieu 1983: 193). Die Anerkennung der Akteur_innen und der Gegenstände des sozialen Austausches impliziere die Anerkennung der Mitgliedschaft in der entsprechenden Gruppe und verdeutliche auch immer wieder „die Grenzen, jenseits derer die für die Gruppe konstitutiven Austauschbeziehungen (...) nicht stattfinden können“ (ebd. 192).

Bei der Auswertung des Forschungsmaterials gilt es, die spezifischen Ausprägungen von sozialem Kapital in FLOSS-Communities zu rekonstruieren. Wie kann soziales Kapital als Ressource nutzbar gemacht werden? Welche Rituale und Institutionen fördern die Akkumulation von Sozialkapital?

Mit kulturellem Kapital meint Bourdieu Bildung. Kulturelles Kapital könne wiederum in inkorporierter, in objektivierter und in institutionalisierter Form vorkommen. *Inkorporiertes* Kulturkapital entstehe in einem Prozess der Verinnerlichung, des Erwerbs von Bildung über einen (meist längeren) Zeitraum und nur bei den Personen, die diese Zeit selbst investierten (vgl. Bourdieu 1983: 186). „Inkorporiertes Kapital ist ein Besitztum, das zu einem festen Bestandteil der ‚Person‘, zum Habitus geworden ist“ (ebd. 187). Die sozialen Bedingungen, unter denen inkorporiertes Kulturkapital akkumuliert und weitergegeben wird, hält Bourdieu für zentral, da sie die verfügbare Zeit maßgeblich beeinflussen, die Menschen für den Erwerb von kulturellem Kapital investieren können:

„[A]ndererseits ist aber auch bekannt, daß die Akkumulation kulturellen Kapitals von frühester Kindheit an — die Voraussetzung zur schnellen und mühelosen Aneignung jeglicher Art von nützlichen Fähigkeiten — ohne Verzögerung und Zeitverlust — nur in Familien stattfindet, die über ein so starkes Kulturkapital verfügen, daß die gesamte Zeit der Sozialisation zugleich eine Zeit der Akkumulation ist. Daraus folgt, daß die

Übertragung von Kulturkapital zweifellos die am besten verschleierte Form erblicher Übertragung von Kapital ist“ (Bourdieu 1983: 188).

Diese sozialen Voraussetzungen für die Aneignung von Linux werden in Kapitel 5 thematisiert. (Über sie wird die teilnehmende Beobachtung jedoch nur begrenzt Auskunft geben, so dass Sekundärliteratur herangezogen wird.)

Unter *objektiviertem* Kulturkapital fasst Bourdieu vergegenständlichte kulturelle Güter wie z. B. Bücher, Hardware oder Websites (vgl. ebd. 185). Objektiviertes Kulturkapital lasse sich in Form von materiellen Gütern weitergeben oder kaufen. Wer jedoch nicht über ein entsprechendes inkorporiertes Kulturkapital verfüge, könne sich kulturelle Güter nicht im eigentlichen Sinne *aneignen*: Eine Website haben und sie selbst programmieren, das sind verschiedene Dinge. Einen Computer anzuschaffen, bedeutet noch nicht, dass man ihn bedienen oder mit speziell ausgewählten Zusatzteilen ausstatten kann.⁶ Objektiviertes Kulturkapital werde von den Akteur_innen als „Waffe“ oder „Einsatz“ genutzt, um in bestimmten Feldern von ihrem inkorporierten Kulturkapital zu profitieren (vgl. ebd. 188 f.). Akteur_innen verdienen beispielsweise ihr Geld mit dem, was sie sich langfristig angeeignet haben, setzen ihr kulturelles in ökonomisches Kapital um. Diejenigen, die sie dafür bezahlen, verfügen dagegen über ökonomisches Kapital, jedoch nicht unbedingt über das benötigte inkorporierte kulturelle Kapital.

Als dritte Form von kulturellem Kapital nennt Bourdieu *institutionalisiertes* Kulturkapital, schulische bzw. akademische Titel, die dauerhafte Nachweise von inkorporiertem kulturellem Kapital darstellen. „Titel schaffen einen Unterschied zwischen dem kulturellen Kapital des Autodidakten, das ständig unter Beweiszwang steht, und dem kulturellen Kapital, das durch Titel schulisch sanktioniert und rechtlich garantiert ist“ (ebd. 189 f.). Institutionalisiertes Kulturkapital werde besonders offensichtlich eingesetzt, um Macht auf Menschen auszuüben, indem es sie dazu bringe, denen, die Titel innehaben, zu glauben und sie anzuerkennen (vgl. ebd. 190).

Welche Formen von kulturellem Kapital in Linux-Communities akkumuliert werden und wie dies vonstatten geht, ist mit einer zentralen Frage dieser Arbeit verbunden, nämlich mit der Frage nach dem postulierten offenen Umgang mit Wissen und Fähigkeiten. Kulturelles Kapital wird in Linux User Groups ständig angesammelt und weitergegeben; diese

⁶Eine extra nachgekaufte WLAN-Karte mit einem bestimmten Chipsatz als Ersatz für die standardmäßig eingebaute WLAN-Karte ist objektiviertes Kulturkapital. Jede_r kann sie sich kaufen, aber nur, wer das entsprechende kulturelle Kapital inkorporiert hat, weiß sie zu schätzen oder ist u.U. dazu in der Lage, sie mit den richtigen Treibern zum Laufen zu bringen.

Prozesse sind Hauptgegenstand meiner teilnehmenden Beobachtung. Kulturelles Kapital hat nicht nur einen sehr hohen Stellenwert, sondern — so meine Annahme — die spezifische Gewichtung verschiedener Formen kulturellen Kapitals ist konstitutiv für das Feld FLOSS-Communities.

Die Ebene der Symbole und Repräsentationen findet bei Bourdieu nicht dieselbe gesonderte Beachtung wie die Praktiken von Individuen (Habitus) und die gesellschaftlichen Strukturen (Kapitalformen im Feld). Er spricht von symbolischem Kapital.⁷ Die Akteur_innen, die aufgrund beliebiger Merkmale über viel symbolisches Kapital verfügen, können kulturelle Symbole und ihre (Be-)Deutungen *setzen*. Deuten und anwenden können sie nur diejenigen, die den communityspezifischen Habitus inkorporiert haben und damit über die nötigen Wahrnehmungs- und Deutungsmuster verfügen (vgl. Bourdieu 1998b: 173). Symbolisches Kapital bezeichnet Bourdieu auch als Prestige. Der Begriff bildet jegliche Form von sozialer Anerkennung und Wertschätzung ab und ist damit nicht scharf von den zuvor genannten Kapitalformen trennbar. Soziales Kapital habe immer auch die Funktion von symbolischem Kapital. Kulturkapital ver helfe vor allem in institutionalisierter Form zu symbolischem Kapital, da eine durch formale Abschlüsse legitimierte Bildung zu sozialer Anerkennung führe (vgl. Schwingel 2000: 90f.). Symbolisches Kapital übe auf *alle* im Feld Beteiligten die Wirkung aus, dass der Charakter von Beziehungen verklärt würde: Ungleichheiten, Herrschafts- und Unterwerfungsbeziehungen würden als affektive Beziehungen wahrgenommen, Macht als Charisma oder Charme (vgl. Bourdieu 1998b: 173). Damit diese Verklärung funktioniere, müsste im Feld eine „oft unsichtbare und jedenfalls vergessene, verdrängte Arbeit vorangegangen sein“ (ebd. 174), die bei den Menschen entsprechende mentale Dispositionen erzeugt hat, so dass sie ihre Unterwerfung und die ausgeübte symbolische Gewalt nicht hinterfragten. Ein Beispiel aus FLOSS-Communities könnte sein, dass Personen sich ihre Rechner von Helfer_innen wegnehmen lassen, weil ihre mentalen Strukturen im Einklang sind mit den Strukturen, die in die Aufforderung eingeflossen sind, den Rechner abzugeben (vgl. analog dazu ebd.). Symbolisches Kapital könne demnach nur wirken, wenn die Beteiligten die habituellen Voraussetzungen erfüllten, um es wahrzunehmen.

Bourdieu geht des Weiteren davon aus, dass die jeweilige Elite eines Feldes nicht nur am Erhalt ihres akkumulierten Kapitals interessiert ist, sondern auch an seiner Weitergabe,

⁷In weiteren Publikationen nennt er z. B. auch politisches Kapital oder technologisches Kapital, diese Formen hat er im Vergleich zu ökonomischem, sozialem, kulturellem und symbolischem Kapital jedoch nicht systematisch ausgearbeitet.

an der Reproduktion der Elite. Die Reproduktionsstrategien seien fortlaufend Gegenstand subversiver Kritik, z. B. Kritik an Erbfolgesetzen. Daraufhin entwickle die Elite ständig Strategien, die den Kapitaltransfer verschleiern sollen (vgl. Bourdieu 1983: 198). Die wichtigste Strategie sei die Konvertierung einer Kapitalform in andere Formen.

„Je mehr die offizielle Übertragung von ökonomischem Kapital verhindert oder gebremst wird, desto stärker bestimmt deshalb die geheime Zirkulation von Kapital in Gestalt der verschiedenen Formen des Kulturkapitals die Reproduktion der gesellschaftlichen Struktur“ (ebd.).

Ökonomisches Kapital liege zwar den anderen Formen zugrunde, sei aber bei der Weitergabe oder Vererbung am offensichtlichsten. Es könne durch besondere Arbeit und unter gewissem Schwund in soziales oder kulturelles Kapital umgewandelt werden. Diese Kapitalformen entfalten wiederum ihre spezifischen Wirkungen gerade deshalb, weil sie verschleiern, dass ihnen ökonomisches Kapital zugrundeliegt. Bourdieu betont dabei die Rolle des Unterrichtssystems mit seinen Bildungstiteln für die Reproduktion von Klassen (vgl. ebd. 196).⁸

Zur Untersuchung der Kapitalformen in FLOSS-Communities gehört die Analyse von Prozessen der Kapitalkonvertierung in diesem Feld. Wie wird Kapital reproduziert? Sind bestimmte Reproduktionsstrategien von Kapital, von Macht, besonders umkämpft? Gibt es eine „herrschende Klasse“ in den Communities und wodurch zeichnet sie sich aus? Welche Wege nimmt verschleiertes Kapital, wenn es weitergegeben wird? Wie relevant ist dabei der Wissenstransfer? Oder ist die Szene so offen, dass Szeneeliten ihre Reproduktion nicht verschleiern müssen?

3.2. Anhaltspunkte für die Untersuchung von Praxis

Die Ebene der Praxis ist nach Degele / Winker die Ebene, auf der Einzelne über sich und ihr Umfeld implizite und explizite Aussagen treffen, Klassifizierungen vornehmen und sich

⁸Als Beispiel für die Konvertierung von ökonomischem in soziales Kapital nennt Bourdieu selbstgemachte Geschenke, in denen Zeit, Geld und Gedanken stecken. Dies sei eine Arbeit an sozialen Beziehungen, von denen erst zu einem späteren Zeitpunkt profitiert werden könne (vgl. Bourdieu 1983: 196). Die Übertragung von kulturellem Kapital vor allem durch Erziehung und Sozialisation erfolge über sehr lange Zeiträume und sei schwer zu fassen. Um zu wirken, sei kulturelles Kapital zunehmend auf die Institutionalisierung in Form von formalen Bildungsabschlüssen angewiesen. Beide Beispiele zeigen auch das Risiko, den potenziellen Schwund, den die Kapitalkonvertierung mit sich bringt (vgl. ebd. 198).

damit ständig sozial positionieren (vgl. Degele/Winker 2009: 65). Mich interessieren die Dynamiken, über die in meinem Forschungsfeld Differenzen innerhalb von FLOSS-Communities geschaffen werden. Dies sind Prozesse, in denen Menschen sich (mehr oder weniger subtil) voneinander abheben. Für die Untersuchung dieser Differenzierungs- und Hierarchiebildungsprozesse greife ich auf Bourdieus Konzepte von Habitus und Praxis zurück.

3.2.1. Die Strukturierung der Praxis durch den Habitus

Bourdieu spricht von praktischem Sinn oder auch praktischer Logik, um zu beschreiben, was er mit Praxis meint. Er sieht die praktische Logik in Abgrenzung zur wissenschaftlichen Logik: Das Spezifikum der Praxis sei, dass sie sich keine Fragen über sich selbst stellt und stellen muss (vgl. Bourdieu 1999: 151 ff.). Die Mitglieder von FLOSS-Communities *handeln* einfach. Als Wissenschaftlerin trage ich meine eigene, dem Feld fremde Logik an sie heran. Die Praxis ergibt sich für Bourdieu aus einem *verinnerlichten Sinn* für die Praxis, dem Habitus. Der Habitus ermögliche Akteur_innen dauerhafte Orientierung im sozialen Raum, indem er ihnen Denk-, Wahrnehmungs- und Handlungsschemata zu Verfügung stelle (vgl. Schwingel 2000: 57 ff.; Bourdieu 1999: 107). Mit dem Habituskonzept untersucht Bourdieu Individuen als soziale Akteur_innen soziologisch.

„Die Subjekte sind (...) handelnde und erkennende Akteure, die über Praxissinn verfügen (...), über ein erworbenes Präferenzensystem, ein System von Wahrnehmungs- und Gliederungsprinzipien (das, was man gewöhnlich den Geschmack nennt), von dauerhaften kognitiven Strukturen (die im wesentlichen das Produkt der Inkorporierung der objektiven Strukturen sind) und Handlungsschemata, von denen sich die Wahrnehmung der Situation und die darauf abgestimmte Reaktion leiten lässt. Der Habitus ist jener Praxissinn, der einem sagt, was in einer bestimmten Situation zu tun ist - im Sport nennt man das ein Gespür für das Spiel, nämlich die Kunst, den zukünftigen Verlauf des Spiels, der sich im gegenwärtigen Stand des Spiels bereits abzeichnet, zu antizipieren“ (Bourdieu 1998b: 41 f.).

Die Mitglieder eines Feldes haben — so Bourdieu — einen gemeinsamen Habitus inkorporiert. Dieser Habitus erzeugt ihre sozialen Praxisformen nach Schemata, die dem Feld angepasst sind: Sie lassen die Akteur_innen wissen, was in feldspezifischen Situationen, z. B. bei der gegenseitigen Hilfestellung bei LUG-Treffen, zu tun ist. Sie lassen sie verstehen, wie Situationen zu deuten sind, z. B. der Rat, jetzt eine → *Konsole* zu öffnen. Die in-

korporierten Wahrnehmungsschemata des Habitus lassen sowohl Praxisformen als auch die Kapitaldistribution im Feld selbstverständlich erscheinen und machen Infragestellungen unwahrscheinlich (vgl. Bourdieu 1999: 113 f.). Bourdieu führt dies darauf zurück, dass die Wahrnehmungs-, Denk- und Handlungsschemata des Habitus nach den gleichen Logiken strukturiert sind wie die objektiven Strukturen des Feldes, also die Verteilung des Kapitals und die Institutionen. Ihr feldspezifischer Habitus sei nämlich durch die Internalisierung von materiellen und kulturellen Existenzbedingungen entstanden, also durch die Verinnerlichung der feldspezifischen Kapitalverteilungsstruktur. Laut Bourdieu wäre der feldspezifische Habitus sehr homogen und historisch gewachsen: Er werde über lange Zeiträume erlernt und sei maßgeblich durch die materiellen und kulturellen Existenzbedingungen geprägt (vgl. Schwingel 2000: 64). Im Verlauf der Sozialisation in der Community wandle er sich ebenfalls in Abhängigkeit von den verfügbaren Ressourcen. Die habituellen Schemata seien meist implizit und unreflektiert, weil ihre Entstehung und ihre Geschichte in Vergessenheit geraten sind. „Als einverleibte, zur Natur gewordene und damit als solche vergessene Geschichte ist der Habitus wirkende Präsenz der gesamten Vergangenheit, die ihn erzeugt hat“ (Bourdieu 1999: 105). In den habituellen Schemata sei diese Geschichte stets präsent; so stelle der Habitus den Akteur_innen historische Erfahrungen zur Verfügung und stelle eine Konstanz der möglichen Praxisformen im Zeitverlauf her (vgl. ebd. 101).

Bourdieu sieht die Praxis von Akteur_innen jedoch nicht im Detail durch den Habitus determiniert. Er untersucht auch nicht in erster Linie die einzelnen Praktiken von Individuen, sondern interessiert sich für übergeordnete gruppen- oder klassenspezifische Praxisformen, als deren Variationen die individuellen Praktiken betrachtet werden können. Der Habitus stecke vielmehr einen Spielraum ab, innerhalb dessen verschiedene individuelle Praxisformen, Denkweisen und Wahrnehmungen möglich seien (vgl. Schwingel 2000: 67 ff.; vgl. auch Fröhlich 1994: 38).

„Da er ein erworbenes System von Erzeugungsschemata ist, können mit dem Habitus alle Gedanken, Wahrnehmungen und Handlungen, und nur diese, frei hervorgebracht werden, die innerhalb der Grenzen der besonderen Bedingungen seiner eigenen Hervorbringung liegen. Über den Habitus regiert die Struktur, die ihn erzeugt hat, die Praxis, und zwar nicht in den Gleisen eines mechanischen Determinismus, sondern über die Einschränkungen und Grenzen, die seinen Erfindungen von vornherein gesetzt sind“ (Bourdieu 1999: 102 f.).

Der Habitus stelle sicher, dass es in der Praxis weder zu völligen Überraschungen noch zu mechanischer Reproduktion komme (vgl. ebd. 103). Akteur_innen hätten mit dem Habitus auch den Spielraum des Möglichen internalisiert, den sie individuell nutzten, wenn sie ihre jeweiligen Ressourcen zum Einsatz brächten (vgl. Schwingel 2000: 70). Mit diesem Konzept könne erklärt werden, wie die „stilistische Einheitlichkeit“ der Praktiken und materiellen Güter von Akteur_innen oder ganzen Klassen zustande kommt (vgl. Bourdieu 1998b: 21). Hier wird die zirkuläre Verknüpfung von Habitus, Praxis und Existenzbedingungen bei Bourdieu deutlich: Die historisch gewachsenen objektiven Strukturen eines Feldes strukturieren die habituellen Dispositionen von Akteur_innen, die wiederum eine bestimmte Praxis dieser Akteur_innen hervorbringen. Sowohl die objektiven Strukturen als auch der Habitus werden nur in der Praxis konstituiert und durch sie stabilisiert (vgl. Schwingel 2000: 67 ff.; vgl. auch Fröhlich 1994: 38).

Eine meiner Prämissen ist, dass die Mitglieder von FLOSS-Communities zumindest für ihre Zusammentreffen im Feld über einen geteilten Habitus verfügen. Er produziert Differenz und strukturiert damit das Feld nach innen. Die Analyse des Beobachtungsmaterials soll eine Beschreibung des feldspezifischen Habitus von FLOSS-Communities ermöglichen, um Aussagen darüber treffen zu können, welche Wahrnehmungs-, Denk- und Handlungsschemata die Praxis Linux strukturieren und welche objektiven Strukturen durch diese Praxis konstituiert und stabilisiert werden.

3.2.2. Distinktion: Die Herstellung von Unterschieden über Stil

Die habituellen Wahrnehmungsschemata von Community-Mitgliedern sind für die Betrachtung der sozialen Dynamiken in den Communities, der ständigen Kämpfe um die Kapitalverteilung bzw. um die Machtverhältnisse relevant: In der Praxis der Gruppen laufen ständig Klassifizierungs- und Distinktionsprozesse zwischen den Akteur_innen ab. Unterschiede zwischen Individuen denke ich nicht als bereits vorhandene Unterschiede, sondern als erst in der Praxis hergestellt werdende (vgl. Bourdieu 1998b: 26). Die dazu in der jeweiligen Community nötigen mentalen Dispositionen stellt der Habitus zur Verfügung.

„[E]in Unterschied, ein Unterscheidungsmerkmal, (...) wird nur dann zum sichtbaren, wahrnehmbaren, nicht indifferenten, sozial *relevanten* Unterschied, wenn es von jemandem wahrgenommen wird, der in der Lage ist, *einen Unterschied zu machen* - weil er selber in den betreffenden Raum gehört und daher nicht *indifferent* ist und

3. Theoretischer Zugang: Einführung in Bourdieus Praxistheorie

weil er über die Wahrnehmungskategorien verfügt, die Klassifizierungsschemata, den *Geschmack*, die es ihm erlauben, Unterschiede zu machen, Unterscheidungsvermögen an den Tag zu legen (...). Der Unterschied wird erst dann zum Zeichen und zum Zeichen des Distinguierten (oder Vulgären), wenn man ein Wahrnehmungs- und Gliederungsprinzip auf ihn anwendet“ (Bourdieu 1998b: 22 f.).

Entsprechende Wahrnehmungsschemata machen laut Bourdieu Unterschiede erst zu sozial relevanten. Bourdieu zeigt in *Die feinen Unterschiede* am Beispiel Geschmack, wie ästhetische Einstellungen, also Formen von kulturellem Kapital, sozial bedeutsam werden können, wenn sie auf einen Habitus treffen, der sie entsprechend verarbeitet: „Der Geschmack ist die Grundlage alles dessen, was man hat (...), wie dessen, was man für die anderen ist, dessen, womit man sich selbst einordnet und von den anderen eingeordnet wird“ (Bourdieu 1987: 104). Ästhetische Äußerungen (in Form von Kleidung, bevorzugten Technologien etc.) bilden ständige Abhebungen von anderen Geschmäckern und bringen Lebensstile hervor, die zu mächtigen „Klassenschranken“ (ebd. S. 106) werden. Sie zeigen allen im selben sozialen Raum, hier der FLOSS-Szene, wo jemand sich verortet und wovon er_sie sich damit abgrenzt. Daraus entstehe ein „Spiel sich gegenseitig ablehnender Ablehnungen“ (ebd. 107). In Bourdieus Vorstellung von einer stark stratifizierten Gesellschaft wird dieses Spiel von der „herrschenden Klasse“ bestimmt: Übertragen auf mein Feld wären technisch hochgradig versierte Linux-Nutzer_innen und -expert_innen in der Lage, „ihre Lebensform zu einer Kunstform zu erheben“ (ebd.). Sie könnten objektive soziale Unterschiede, nämlich die ungleiche Distribution von Kapital, so verschleiern, dass sie als auf bewusster ästhetischer Wahl beruhende Unterscheidungen erscheinen. Beispielsweise könnten sie ihre Verwendung textbasierter Programme als Geschmacksäußerungen inszenieren und damit verdecken, wie viel Zeit sie in das Erlernen seiner Bedienung gesteckt haben (vgl. Kap. 6). Die nicht „herrschenden Klassen“ hätten andere Rollen inne: Diejenigen, die sich noch im langwierigen Lernprozess befinden, hätten ständig Angst, sich durch ihre (Geschmacks- oder Kompetenz-)Äußerungen einer sozialen Klassifizierung auszusetzen, indem sie z. B. ihre Wissenslücken aufdecken. Diejenigen, die Linux zwar nutzen, sich aber nicht für das Betriebssystem selbst interessieren, und erst recht Windows-Nutzer_innen wären negative Bezugspunkte für alle anderen (vgl. Bourdieu 1987: 107). Ästhetische Entscheidungen dienen der Abgrenzung von „sozial weiter unten stehenden Gruppen“, oftmals genau von denen, zu denen eine unmittelbare Konkurrenz um soziale Positionen besteht (ebd. 111). Demnach arbeiten alle Akteur_innen in einem sozialen Raum ständig in solchen Distinktionsprozessen daran, ihre sozialen Po-

sitionen in Abgrenzung zu anderen sozialen Positionen zu manifestieren. Auch Hitzler et al. betrachten Stilisierung als Merkmal von Szenen: Unter Stilisierung verstehen sie „das nach (bestimmten, nämlich szenespezifischen) ästhetischen Kriterien selektierte Verwenden von Zeichen(-arrangements) mit der Absicht, einen (kulturell relativ) kompetenten *und zugleich* originellen Eindruck zu machen“ (Hitzler et al. 2001: 215 f., Herv. im Orig.). Der jeweilige Stil könne in seinem Detail nur von Eingeweihten gelesen und mit dem spezifischen sozialen Kontext in Verbindung gebracht werden. „Stilisieren ist für den Interaktionszusammenhang ‚Szene‘ konstitutiv, da durch dieses Handeln die Kultur der Szene am wirksamsten reproduziert bzw. weiterentwickelt wird“ (ebd. 216). Diesen Stil setzen zu können, ihn zu pflegen und zu lesen, ist Teil des Dazugehörens zur Szene und ihrer Distinktionsprozesse (vgl. auch Bourdieu 1987: 104 ff.).

Wie die Distinktion über Stil in FLOSS-Communities praktiziert wird und welche Unterschiede dabei produziert werden, ist eine Kernfrage dieser Arbeit, da mich stilistische Distinktion, und besonders Distinktion *über* den Einsatz von Technologien, interessiert. Was Bourdieu über das Unterscheidungsvermögen schreibt, bedeutet für mich, dass ich FLOSS-Communities nur erforschen kann, wenn ich ihre Codes kenne bzw. kennen lerne — ganz Außenstehenden wäre eine Erforschung demnach nicht möglich.

3.3. Theoretisch informierte Reformulierung der Forschungsfragen

Um vor dem Einstieg ins Datenmaterial diese theoretischen Betrachtungen nutzbar zu machen, reformuliere ich die Forschungsfragen für eine Habitusuntersuchung in FLOSS-Communities. Neben Bourdieus Konzepten fließt hier ebenso meine in der Einleitung ausgeführte Forschungsperspektive ein, die einen feministischen, machtkritischen und skeptischen Blick auf soziale Ungleichheitsverhältnisse wirft. Zudem leitet mich das Anliegen, das Feld von innen heraus verständlich zu machen und die Selbstverständlichkeiten offen zu legen.

Zur Bearbeitung meiner ersten Forschungsfrage nach den konstitutiven Praktiken von FLOSS-Communities stelle ich quasi naiv die von Clifford Geertz entlehnte Frage „What the hell is going on here?“ (Amann/Hirschauer 1997: 20). Anders als Christopher Kelty befrage ich das Feld nicht über explizit erklärte Praktiken.⁹ Ich versuche vielmehr, das

⁹Kelty macht in seiner Ethnografie ebenfalls vier Praxiskomplexe aus. Sie entsprechen den erklärten gemeinsamen Aktivitäten von FLOSS-Communities: Quellcode teilen, offene Systeme konzipieren, Lizen-

zu benennen, was diesen *erklärten* Praktiken zugrunde liegt. Die folgenden vier Kapitel sind nach vier zentralen Praxiskomplexen gegliedert, die ich aus dem Material herausgearbeitet habe: eine gemeinsame Sprache (1), ein spezifischer Umgang mit Wissen und eine szeninterne Organisation von Lernverhältnissen (2), eine gemeinsame Techniknutzung und -aneignung (3) und Praktiken des Formalisierens von technischen Artefakten und sozialen Ritualen (4).

Meine Fragen nach dem praktischen Wissen, das Linux-Nutzer_innen mobilisieren, und nach den Kategorien, entlang derer in diesen vier Praxiskomplexen Differenzen hergestellt werden, fallen — mit Bourdieu gedacht — zusammen: Sie zielen auf eine Rekonstruktion des Habitus und der relevanten Kapitalformen ab. Welcher praktischen Logik folgt das, was sie im Bezug aufeinander und auf technische Artefakte tun? An welche Regeln und Strukturen glauben sie? Wie lässt sich die feldspezifische *illusio* beschreiben? Welche Formen von Kapital häufen die Community-Mitglieder an und welche feldspezifischen Ausprägungen haben diese Kapitalformen? Wie setzen die Beteiligten sie ein? Welche Wahrnehmungsmuster sind nötig, um im Feld Unterschiede machen zu können? Nicht zuletzt stellt sich bezüglich Linux User Groups als Lernumfeld die Frage, ob überhaupt von einem homogenen Habitus, wie Bourdieu ihn konzipiert gesprochen werden kann.

Schließlich interessiert mich der Stellenwert der technischen Artefakte in den habituellen Praktiken: Wie erforderlich ist es, einen szenespezifischen Habitus inkorporiert zu haben, um Linux einzusetzen und es sich anzueignen? Bezüglich der Artefakte nutze ich besonders den Begriff der Distinktion: Wie heben sich also Community-Mitglieder über geschmackliche Äußerungen in Bezug auf freie Software voneinander ab? Anhand von welchen Kriterien stilisieren sie ihren Umgang mit Linux? Welche Codes verwenden sie dafür?

zen schreiben und Zusammenarbeit koordinieren (Kelty 2008b). Daneben untersucht er die Metapraxis der Kommunikation über diese vier Komplexe.

4. Eine Sprache sprechen

Die Sprache, die in FLOSS-Communities gesprochen wird, wird als erster Punkt in dieser Arbeit behandelt, strukturiert die spezifische Sprache doch maßgeblich die Szene, das gesamte Forschungsmaterial und diese Arbeit selbst. Eine gemeinsame Sprache zu sprechen, ist eine konstitutive Praxis der Gruppen. Dies ist mitnichten ein Phänomen, das für diese Szene spezifisch ist, Szenesprachen wurden vor allem im Bezug auf Jugendstadien schon ausführlich untersucht (vgl. etwa Janke/Niehues 1995; Klein/Friedrich 2003; Baacke 2007). Szenesprachen gelten als ein- und ausgrenzend, sie dienen häufig der kulturellen Abgrenzung und stärken das Zugehörigkeitsgefühl zur jeweiligen Szene (vgl. Klein/Friedrich 2003: 37). In FLOSS-Communities wird technische Fachsprache gesprochen. Das liegt zum einen an den Inhalten der Gespräche. Freie Software und Hardware sind die häufigsten Gesprächsthemen der Community-Treffen. Auch berufliche Fragen technisch-inhaltlicher Natur und berufliche Pläne mit freier Software werden häufig besprochen.¹ Die Fachsprache hat sich aber auch ihren Weg in die Umgangssprache der Communities gebahnt. Sie ist von technischen Begriffen durchzogen, die auf besondere Weise in den Humor eingeflochten werden. Dieser Humor, der aus der Technikaffinität schöpft, zeichnet die Communities aus. Auch dies steht in einem größeren Kontext: „Computer-orientierter Humor“ (Friedman/Friedman 2003) ist kein auf FLOSS-Communities beschränktes Charakteristikum. Linda Weiser Friedman und Hershey H. Friedman sprechen vielmehr von einem besonderen Humor im gesamten Computerbereich, der „intelligent, somewhat superior, and slyly subversive“ sei (ebd. 1). Andere technische Artefakte, die die Welt ähnlich tiefgreifend verändert haben, hätten *bei Weitem* nicht so viel Humor hervorgebracht wie Computer. Sprache, Wortwitz und Humor in den Communities sind in besonderem Maße von der Beschäftigung mit *linuxspezifischen* Technologien geprägt.

¹ „A1, A44 und ich gehen mit. Auf dem Weg hinüber reden die drei nur über Hardware und zwar gespickt mit technischen Details, Zahlen, was sie haben, was sie sich wünschen. (...) Zwischendurch geht es wieder zu anderen Themen: Kunden, zu ihrer Hardware, was auf ihren Servern läuft“ (LUG 2008j).

In diesem Kapitel geht es um diese linuxspezifische Fachsprache als gemeinsame Praxis und um die Wirkungen, die aus ihr entstehen. Zunächst gehe ich auf die technisierte Sprache und den darin eingeflochtenen Humor ein (4.1). Politische Anspielungen interpretiere ich dann als inhaltlich stark verkürztes Sprechen über Politik (4.2). Daraus ergibt sich die Frage, ob die Beteiligten sich selbst und Linux als politisches Projekt verorten. Sie wird in einem Exkurs und unter Rückgriff auf Literatur diskutiert (4.3). Vergeschlechtlichten Stereotypen und sexistischen Witzen widme ich schließlich einen separaten Abschnitt (4.4).

4.1. Mitreden und mitlachen können

Die vielleicht bekannteste sprachliche Eigenheit von FLOSS-Communities sind rekursive Akronyme. Dies sind Abkürzungen, die sich, wenn man sie ausschreibt, auf sich selbst beziehen, weil die Abkürzung in der Langversion erhalten bleibt. Das erste freie Betriebssystem heißt GNU, was für „GNU is not Unix“ steht. Der Erfinder von *GNU*, Richard Stallman hatte jedoch genau das Ziel, ein Betriebssystem zu schreiben, das Unix vergleichbare Funktionen hat, nur ohne auf den von *AT&T* geschützten → *Quellcode* zurückzugreifen (vgl. Grassmuck 2004: 222). „This kind of recursion is often used as a programming technique, and applying it to words is highly popular amongst hackers“ (Moody 2002: 20). Rekursion in der Programmierung ist z. B. eine Funktion, die sich selbst immer wieder aufruft.² Eine ganze Reihe von Programmen haben rekursive Akronyme zum Namen, z. B. *YaST* (*Yet another Setup Tool*)³ mancher oder *LAME* (*Lame Ain't An MP3 Encoder*). Oft wirken die Bezeichnungen dadurch absurd, dass sie genau das verneinen, was sie bezeichnen: *Lame* ist beispielsweise inzwischen ein Mp3-Kodierer, war es aber zum Zeitpunkt der Benennung noch nicht. Die Geschichte des Programmes bildet sich hier im Namen ab. An das rekursive Akronym *GNU* knüpfte wiederum ein als → *Bugreport* formulierter Witz an, der besagt, dass die englische Sprache an rekursiven Akronymen abstürze, dass sie also im Gegensatz zu Programmiersprachen der Rekursion nicht gewachsen sei.⁴ Es gibt eine sehr große Anzahl solcher humorvoller Bezüge, die in irgendeiner Form mit technischen

²Diese Programmieretechnik wird beispielsweise in Sortieralgorithmen verwendet: Die Funktion sucht zwei Ausdrücke, vergleicht sie, stellt sie in die richtige Reihenfolge und beginnt so oft wieder von vorn, bis alle Ausdrücke in der richtigen Reihenfolge sortiert sind.

³*YaST* ist eine Software für die Konfiguration des Betriebssystems, die in der → *Distribution* OpenSUSE zum Einsatz kommt.

⁴<http://www.gnu.org/fun/jokes/gnu-overflow.html>

Konventionen hantieren. Ein Witz, der auf ähnliche Weise auf mathematisches Denken zurückgreift, steht öfter auf T-Shirts oder Websites: „There are only 10 types of people in the world — those who understand binary, and those who don't.“ Die Zahl 10 im Dualsystem entspricht der Zahl 2 im Dezimalsystem. Der Satz verweist diejenigen, die ihn nicht verstehen, direkt auf ihr eigenes Unwissen zurück. In queeren technikaffinen Kontexten wird dasselbe Zitat wiederum gerne aufgegriffen, weil es auch als Infragestellung des binären Geschlechtersystems gelesen werden kann, etwa in diesem Sinne: „Wir verstehen das Binärsystem nicht — es gibt doch nicht nur „Mann“ und „Frau“, sondern vielleicht „10 types of people“. Das Nicht-Verstehen(wollen) des Binärsystems wird hier positiv gewendet.⁵

An vielen Stellen fließen technische Ausdrücke in die Umgangssprache ein, die dadurch auch nur bestimmte, dieser Sprache mächtige Personen adressiert.

A1⁶ruft nach hinten, ob A48 noch da ist. Jemand aus dem anderen Raum antwortet: „Nein!“ A1 ruft: „Dann rm minus rf!“ A54 fragt, was das heißen soll. A1: „Restloses Löschen.“ Sie lacht (LUG 2009h).

„Dann vergiß es“, ist der Sinn, in dem A1 den Befehl für die → *Kommandozeile*, *rm -rf*, verwendet, einen Befehl, der in einem Verzeichnis alle Dateien und Unterverzeichnisse löschen würde. A54 versteht die Interaktion nur durch Nachfragen. Aus der vorhergehenden Situation heraus, habe ich die Situation so interpretiert, dass A1 auch wusste, dass seine Sitznachbarin seine Äußerung nicht verstehen würde. Auch in der beobachteten IT-Abteilung gab es eine ähnliche Szene:

A27 hat in der Mittagspause von einem Community-Event Folgendes erzählt: A28 habe A29 und seine Tochter beim Event begrüßt mit „Hallo A29, hi Fork!“ Die IT-Kolleg_innen am Tisch, die das verstehen, lachen, die anderen fragen nach (Fir 2008b).

Ein *Fork* ist eine Abspaltung von einem Softwareprojekt, in der der → *Quellcode* kopiert und unabhängig vom ursprünglichen Projekt weiterentwickelt wird. In der beschriebenen Szene wird ein Kind als *Fork* bezeichnet, gewissermaßen als genetische Abspaltung des Vaters. Beide Beobachtungen setzen auf die Voraussetzungen der Hörer_innen und in

⁵Diese Lesart kam in einer Diskussion mit der AG Queer Studies des Asta der Universität Hamburg zur Sprache.

⁶Zur Anonymisierung der Namen und Orte habe ich folgende Bezeichnungen gewählt: Personen beginnen mit dem Buchstaben A und sind dann in der Reihenfolge durchnummeriert, in der ich das Material ausgewertet habe. Orte beginnen mit Z.

beide Situationen waren Menschen involviert, von denen bekannt war, dass sie nicht der Szene angehören. Der Esprit ist hier und in sehr vielen Situationen nur erfassbar, wenn die Anwesenden ein gemeinsames Wissen oder einen Erfahrungsschatz teilen und dadurch eine „zusammenschweißende“ Wirkung erfahren. Friedman / Friedman bezeichnen das Besondere an diesem verbindenden computer-orientierten Humor mit dem „I-get-it“-Effekt: „In fact, for both parties in the I-get-it transaction, the greatest satisfaction is derived when no explanation is required, requested, or offered. Between the narrator and the listener, there is an implicit understanding, the secret handshake“ (Friedman/Friedman 2003: 13). Sie unterscheiden diese I-get-it-Momente von einem Humor, der auf die Demonstration von Überlegenheit abzielt. Es gehe eben nicht darum, jemanden zu verspotten, sondern das gemeinsame Verständnis unter Insider_innen zu genießen. „Those who would not understand the joke — the presumed outsiders — are not relevant to this transaction because they are not there“ (ebd.). Die Autor_innen gehen von computer-orientiertem Humor innerhalb von wissensmäßig homogenen Gruppen aus. Meines Erachtens lässt sich diese Trennung von „secret handshake“ und Überlegenheitsdemonstration in Linux User Groups nicht aufrechterhalten, eben weil das vorausgesetzte Wissen so ungleich verteilt ist, weil die Gruppen heterogen sind. Die Freude am I-get-it-Effekt unter einigen wirkt in sehr vielen Situationen gleichzeitig auf andere Anwesende eben doch als Zurschaustellung von Überlegenheit über andere (vgl. ebd. 8) — „I get it, but you don't.“

Es gibt zahlreiche andere Beispiele für die Technisierung der Umgangssprache in FLOSS-Communities: Im Chat wird aus „Arbeit“ „arbyte“⁷, die große gemeinsame Pizza-Bestellung bei einem alljährlichen Community-Event heißt „Pizza-Proxy“⁸ und diejenigen, die sich für die Ausweiskontrolle bei der Keysigning-Party (vgl. 7.2) aufstellen, werden scherzhaft aufgefordert, sich einen „Sortieralgorithmus“ auszudenken, um sich in die richtige Reihenfolge zu bringen. Teilweise wird in den Interaktionen auf Logiken aus Programmiersprachen zurückgegriffen. Ich diskutierte zum Beispiel mit jemandem per Chat über das von mir verwendete Binnen-I. Mein Kommunikationspartner verwendete die Schreibweise „\${word}Innen“⁹, um ein beliebiges Wort mit der Endung „Innen“ zu beschreiben. Andere Begriffe, die als Platzhalter für Beliebiges eingesetzt werden, sind „foo“, „bar“ oder „foobar“. „Foo“ ist sowohl in schnell hingeschriebenen Beispielskripten als Va-

⁷Byte ist eine Maßeinheit für eine digitale Datenmenge.

⁸Ein Proxyserver leitet alle Anfragen aus einem Computernetzwerk nach außen weiter.

⁹Die Schreibweise $\{\dots\}$ bezieht sich auf die Nutzung von Variablen beim Programmieren. $\{\text{word}\}$ wäre eine beliebig definierbare Variable, an die dann die Endung „Innen“ gehängt wird.

riable anzutreffen als auch in der gesprochenen Sprache als Synonym für „Zeug“ oder „Chaos“ oder in zahlreichen anderen Verwendungen. Auch die beiden großen Vortragsräume bei einem Community-Event trugen die Namen „Foo“ und „Bar“. Die Etymologie und Bedeutung dieser Begriffe ist nicht eindeutig geklärt, so dass sie — mitten unter den offiziellen → *Request for Comments*-Dokumenten — in dem humorvollen RFC 3092 einfach definiert wurde. Zu „Foo“ ist dort zu lesen:

„Used very generally as a sample name for absolutely anything, esp. programs and files (...). First on the standard list of metasyntactic variables¹⁰ used in syntax examples“¹¹

Als mögliche Herkunft wird ein Akronym aus dem Slang der US-Armee im Zweiten Weltkrieg genannt: Fubar könne „Fucked up beyond all Recognition/Repair“ bedeutet haben. Das Wort „Fubar“ könne vom deutschen Wort „furchtbar“ beeinflusst sein (vgl. ebd.) Es entspricht dem Humor der Communities, neben technischen Spezifikationen auch solche humorvollen Konventionen in offiziellen RFCs niederzuschreiben.

Kreatives Spielen mit Elementen aus dem Programmieren ist gängig: Ein Community-Mitglied verschickte eine Geburtstags Einladung in der Syntax der Programmiersprache *Python* (vgl. Abb. 4.1). In der Einladung steht, dass die Freund_innen von A104 aus einer Datei importiert werden sollen. Dann werden seine Kontaktdaten und Zeit und Ort des Ereignisses definiert. Schließlich soll jede Person in der Datei A104 bestätigen, ob er oder sie zu dem „Event“ kommt, wenn möglich kommen und seinen dreißigsten Geburtstag mitfeiern. Bei der Feier stellte sich heraus, dass viele Eingeladene die E-Mail in der Syntax anderer Programmiersprachen beantwortet hatten. Sie haben damit die Kreativität gewissermaßen „gewürdigt“ und gezeigt, dass sie einerseits Programmiersprachen beherrschen und andererseits den Humor teilen.

Anspielungen auf einen geteilten kulturellen Kanon sind sehr häufig. Zu diesem Kanon gehört sowohl linuxspezifisches Wissen als auch bestimmte popkulturelle Werke. Darunter sind viele Science-Fiction-Bücher und -Filme, z. B. Werke von William Gibson (dem Erfinder der Idee des „Cyberspace“) oder die Serie *Star Trek*. Besonders beliebt sind Bezüge auf das Buch *Per Anhalter durch die Galaxis* von Douglas Adams: Die Teilnehmer_innen von

¹⁰Metasyntaktische Variablen haben keine eigene Bedeutung. Sie werden lediglich zur Benennung beliebiger Einheiten verwendet, und das auch meist nur in unfertigen Programmen, bevor sie dann durch sinnvolle Namen ersetzt werden.

¹¹<http://tools.ietf.org/html/rfc3092>, Zugriff: 17.11.2010

4. Eine Sprache sprechen

```
Hallo Silke,  
  
#/usr/bin/env/python  
  
>from A104.friends import *  
  
contact = {  
    "Name" : "A104",  
    "EMail" : "A104@blabla.de",  
    "Telefon": "0111/111 11 11"  
}  
  
event = {  
    "Name" : "Cafe Blabla",  
    "Strasse": "Blastr. 33",  
    "Ort" : "11111 Stadt",  
    "Datum" : "Sonntag, den dd.mm.2007",  
    "Beginn" : "10:30 Uhr",  
    "Essen" : "Brunch"  
}  
  
for person in friends:  
    confirm (contact, event)  
    if(available):  
        come_to (event)  
        celebrate_birthday (name="a104",alter="30")  
  
Beste Gruesse,  
  
A104
```

Abb. 4.1.: Geburtstagseinladung in *Python*-Syntax

Keysigning-Parties werden durchnummeriert. „Ein junger Mann sagt: ‚43 bin ich. Ich wollte 42 sein‘ und lacht“ (KSP 2008a). Auf Community-Events werden u.a. Anstecker verkauft, auf denen die Zahl 42 abgebildet ist. Damit wird immer wieder auf eine Szene in dem Buch angespielt: Dort wird ein Computer damit beauftragt, die Antwort auf die „Letzte aller Fragen nach dem Leben, dem Universum und allem“ zu berechnen. Nach siebeneinhalb Millionen Jahren liefert er als Antwort „42“. Er kommentiert das Ergebnis damit, dass die Menschen nicht genau wüssten, wie die Frage eigentlich lautete. Wenn sie dies herausgefunden hätten, dann verstünden sie auch die Antwort 42. Er werde den Computer konstruieren, der diese wichtigste aller Fragen berechnen würde. Dieser Computer ist die Erde, die jedoch zerstört wird, fünf Minuten, bevor sie die Antwort hat (vgl. Adams 1999: 163 ff.).

Andere Anspielung beziehen sich auf den Diskordianismus, eine Religionsparodie, die durch Robert Anton Wilsons Buch *Illuminatus!* bekannt wurde. Im Diskordianismus geht es um das Hinterfragen von Hierarchie, Ordnung und objektiver Wahrheit. Elemente daraus finden sich an diversen Stellen wieder: So kann man sich in vielen Linux-Distributionen das Datum nach dem Diskordianischen Kalender ausgeben lassen, eine freie Konferenzplanungssoftware trägt den Namen *Pentabarf*, wie die fünf Gebote im Diskordia-

nismus. Auch die Anspielungen auf Kühe in freier Software gehen (wahrscheinlich) auf das *Sacred Chao* zurück, das heilige Symbol des Diskordianismus. *Chao* sei der Singular von Chaos. Dass damit gleichzeitig *cow* gemeint ist, kommt hier zum Ausdruck: „In einer Geschichte in der Principia Discordia antwortet die grasende Heilige Chao auf die Frage ‚Erzähl mir, du stummes Vieh, warum tust du nicht etwas Sinnvolles? Was ist dein Ziel im Leben, auf irgendeine Weise?‘ mit der Antwort ‚Mu‘.“¹² „Mu“ steht hier gleichzeitig für ein „Nichts“, das auf den Zen-Buddhismus anspielt¹³ und für das Muhen einer Kuh, das auch in verschiedene freie Programme eingebaut ist. Ein Beispiel: Es gibt verschiedene Programme zur Installation von Software bzw. zur Verwaltung der installierten Software, darunter *apt-get* und *aptitude*. Ersteres verfügt über „Super-Kuh-Kräfte“, letzteres nicht. Diese Information geht aus den Logmeldungen hervor, die das Programm ausgibt. Damit ist gemeint, dass beide Programme unterschiedliche Antworten beim Aufrufen von *moo* geben (4.2). Während das Programm *apt-get* eine Kuh ausgibt, behauptet *aptitude*, es gebe keine versteckten Witze — so genannte ‚Easter Eggs‘ — in diesem Programm, um bei einem Beharren der Nutzer_innen doch immer mehr davon zu bieten.¹⁴

```

root@lovelace:~# apt-get moo
      (oo)
     /-----\
    /         \
   /           \
  /             \
 /               \
* /             \
  /             \
 /               \
/                 \
... "Have you mooed today?"...
root@lovelace:~# aptitude moo
In diesem Programm gibt es keine Easter Eggs.
root@lovelace:~# aptitude -v moo
In diesem Programm gibt es wirklich keine Easter Eggs.
root@lovelace:~# aptitude -vv moo
Habe ich nicht bereits erklärt, dass es in diesem Programm keine Easter Eggs gibt?
root@lovelace:~# aptitude -vvv moo
Hör auf!
root@lovelace:~# aptitude -vvvv moo
Okay, wenn ich Dir ein Easter Egg gebe, wirst Du dann aufhören?
root@lovelace:~# aptitude -vvvvv moo
Gut, Du hast gewonnen.

      /-----\
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
...
root@lovelace:~# aptitude -vvvvvv moo
Was das ist? Natürlich ein Elefant, der von einer Schlange gefressen wurde.

```

Abb. 4.2.: Verstecker Humor in Software

Viele Linux-Fanartikel spiegeln die Sprache der Communities wider und werden für humorvolle Bezüge genutzt. „A2 trägt ein schwarzes T-Shirt, auf dem vorne *rftm* steht. Hinten steht „read the fucking magazine“ und auf dem Ärmel ist vermerkt, dass das T-Shirt

¹²<http://de.wikipedia.org/wiki/Diskordianismus>, Abruf: 08.02.2011

¹³[http://de.wikipedia.org/wiki/Mu_\(Philosophie\)](http://de.wikipedia.org/wiki/Mu_(Philosophie))

¹⁴Ein anderes Beispiel für die Verwendung von Kühen ist das Programm *cowsay*, das in Abschnitt 6.2 unter einem anderen Aspekt behandelt wird.

von der Computerzeitschrift *c't* ist“ (LUG 2009b). *rtfm* ist eine der bekanntesten Abkürzungen in der Sprache von FLOSS-Communities und bedeutet „Read the fucking manual“ — „Lies die verdammte Dokumentation.“ *rtfm* ist im Internet sehr oft als Antwort auf Fragen zu lesen, die die fragende Person sich durch Lektüre der Handbücher selbst hätte beantworten können. *rtfm* ist ein sehr kurzer und nicht freundlich gemeinter Hinweis darauf, dass man keine Fragen stellt, bevor man nicht die (Online-)Handbücher gelesen hat, die offen zur Verfügung stehen. Auf dem Werbehemd der *c't* taucht der Spruch in abgewandelter Form auf und zieht vor allem das Interesse derer auf sich, die mit der Abkürzung etwas anfangen können. Andere T-Shirts tragen Aufschriften wie „There is no place like 127.0.0.1“¹⁵, „Ich bin root, ich darf das“¹⁶ oder „Sie können Ihren Computer jetzt wegschmeißen“. Dies ist eine Anspielung auf frühere Microsoft Windows-Computer, die beim Herunterfahren auf dem Bildschirm anzeigten „Sie können Ihren Computer jetzt ausschalten“. Die Abgrenzung von Microsoft durch Sprüche, Programme und Fanartikel ist fester Bestandteil der Konstitution von FLOSS-Communities.

Der Humor der Communities zeugt von einem Sinn für Selbstironie und Absurdes. Er arbeitet mit Missverhältnissen (vgl. Friedman/Friedman 2003: 7): Ewig rechnende Computer kommen nicht zu aussagekräftigen Ergebnissen, Menschen denken und formulieren sehr unpräzise, ihre Sprache wird komplexen Anforderungen nicht gerecht. Die ganze Erde ist ein konstruierter Computer zur Beantwortung philosophischer Fragen, der diese nicht liefert. Diese Selbstironie zeigt auch Richard Stallman, wenn er Vorträge zu freier Software mit der Verballhornung einer religiösen Messe der „Holy Church of Emacs“ enden lässt. Die Ironie bezieht sich einerseits auf das Versprechen, Computer könnten alle Probleme lösen: Dieser Glaube, die Verabsolutierung von Computern als Allheilmittel, wird von ihnen ironisch gewendet. Andererseits machen sie sich über ihr eigenes Verhalten lustig, indem sie den Glauben an freie Software als „das Gute“ durch Religionsvergleiche verballhornen. Dem Humor nach zu urteilen, rücken sich die Communities eher in die Nähe einer Glaubensgemeinschaft als in die Nähe einer politischen Bewegung.

¹⁵127.0.0.1 ist immer die → *IP-Adresse* des eigenen Rechners. Das T-Shirt spielt auf das Sprichwort „There is no place like home“ an.

¹⁶Der → *root*-Account auf einem Linuxrechner ist der *Administrator_innen*-Account. Er ist mit allen Rechten ausgestattet.

4.2. „Das ‚s‘ steht für Schäuble“ — Humoristisches Sprechen über Politik

Politische Anspielungen nehmen einen wesentlich kleineren Raum ein als technische. An einer Stelle wird durch eine Anspielung auf den damaligen Innenminister Wolfgang Schäuble ein kleiner Fehler in einem Vortrag überspielt:

A41 will das iso-Image auf den Stick kopieren. Er gibt ein `scp -r ./ * /dev/sda1`. A1 fragt nach dem Pfad und „wieso scp“? A41 merkt, dass er sich an beiden Stellen vertan hat. Er ändert die Zeile in `cp -r ./ /media/disk`. Jemand sagt: „Das ‚s‘ steht für Schäuble.“ Gelächter (LUG 2008a).

Während das Programm *cp* (copy) eine Datei an einen anderen Ort kopiert, ist das Programm *scp* (secure copy) dazu da, eine Datei verschlüsselt von einem Rechner auf einen anderen zu kopieren. Das ‚s‘ steht gewissermaßen für Verschlüsselung. Der Zuhörer, der den Witz macht, betont damit die Relevanz von verschlüsselter Datenübertragung.¹⁷ Damit knüpft er implizit an Schäubles politische Pläne für die Vorratsdatenspeicherung von Telekommunikationsdaten und die Onlinedurchsuchung (den so genannten Bundestrojaner) an. Diese Pläne wurden und werden nicht nur in den Communities scharf kritisiert, sondern Zusammenschlüsse wie der *Chaos Computer Club* zeigen auch regelmäßige Grenzen und Sicherheitslücken in den konkretisierten Umsetzungsvorschlägen auf. In Linux User Groups habe ich zwar etliche solcher politischer Anspielungen beobachtet, jedoch keine inhaltlichen Diskussionen über Tagespolitik oder Vorstellungen zur Gestaltung des Internet mitbekommen. Einige LUG-Mitglieder waren gemeinsam auf einer „Freiheit statt Angst“-Demonstration gegen die geplante Vorratsdatenspeicherung, was beim LUG-Treffen in einem Dialog deutlich wurde.¹⁸

Einer kommt mit einer CD mit Demo-Fotos (wahrscheinlich von der großen Demo gegen Vorratsdatenspeicherung vor ein paar Wochen) und will sie auf den Server der Gruppe1 laden (LUG 2007a).

Das einzige Thema, um das es in den LUGs sonst manchmal ging, war die Unterscheidung zwischen freier Software und Open-Source-Software. Im gesamten Materialkorpus

¹⁷Dass mit dem Einwurf aus dem Publikum ein Irrtum des Referenten überspielt wird, ist ein Beispiel dafür, dass Humor in bestimmten sozialen Situationen eine potenzielle Spannung auflöst (vgl. Friedman/Friedman 2003: 9).

¹⁸Da dies das erste von mir angefertigte Beobachtungsprotokoll ist, habe ich keine Erhebungen zu eventuellen Diskussionen im Vorfeld der Demonstration.

zeichnet sich lediglich der beobachtete Vortrag von Richard Stallman dadurch aus, dass er die Philosophie von freier Software ethisch, politisch und ökonomisch einordnet (vgl. Sta 2008).

Politische Anspielungen finden sich auch in Namen von Programmen: Eine Software zur → *Versionskontrolle* trägt z. B. den Namen *subversion*, ein Programm zum Abspielen von Videos hieß eine Zeit lang *Democracy Player*. Auf Community-Events werden vom → *Debian*-Projekt T-Shirts verkauft, auf denen außer dem *Debian*-Logo der Befehl *apt-get install anarchism* („Installiere Anarchie!“) aufgedruckt ist. Friedman / Friedman würden dies als „antiestablishment“-Humor einordnen (vgl. Friedman/Friedman 2003: 4). Der Befehl lässt sich tatsächlich unter Debian-Linux ausführen und installiert eine Dokumentation der anarchistischen Theorie auf dem Rechner, und zwar — wie es die Konvention erfordert — im Verzeichnis der (sonst technischen) Dokumentationen */usr/share/doc/anarchism*. Die jeweils aktuelle Version des Textes ist auch auf einer Website verfügbar und umfasst weit über 1200 Seiten.¹⁹ Wortspiele mit der Semantik von Ordner- oder Dateinamen sind verbreitet.²⁰ Beides bringt z. B. ein Blogbetreiber zusammen, dessen Blog den Titel *apt-get install anarchism* trägt.²¹ Er hat den Untertitel *echo -n 3 > /sys/power/state* hinzugefügt. Dies bedeutet in etwa, dass er die Texte zu Anarchie in die Datei „Staat“ schreiben möchte. „Staat“ ist im Ordner „Macht“ gespeichert, der wiederum ein Unterordner vom „System“ ist. Der Pfad */sys/power/state* existiert auf Linuxrechnern tatsächlich, hat jedoch etwas mit der Energieversorgung des Rechners zu tun.²² Diese Benennung von Programmen spielt mit der Vorstellung, ein anderes politisches System einfach installieren zu können und somit politische Probleme technisch lösen zu können — eine augenzwinkernde Verkürzung, die in meinem Beobachtungsmaterial unaufgelöst stehen bleibt.

4.3. Exkurs: Linux als politisches Projekt?

In Anbetracht dessen, dass im Datenmaterial Politik abgesehen von humoristischen Bezügen eine marginale Rolle spielt, drängt sich die Frage auf, ob FLOSS-Communities und

¹⁹<http://www.infoshop.org/AnAnarchistFAQ>, Abruf: 21.10.2013. Die Seitenzahl bezieht sich auf ausdruckbare DIN A4-Seiten. Die letzte Aktualisierung ist auf Januar 2010 datiert. Siehe auch https://en.wikipedia.org/wiki/An_Anarchist_FAQ.

²⁰Ein solches Wortspiel ist auch in dem weiter unten angeführten T-Shirt-Text enthalten, wo „nach Hause gehen“ durch *cd ~* ausgedrückt wird — wechsele in das Verzeichnis mit den persönlichen Dateien (S. 89).

²¹<https://aptgetinstallanarchism.wordpress.com/>

²²Eine theoretische Einordnung von den hierarchischen Klassifikationsmethoden der Informatik (z. B. Baumstrukturen) erfolgt in Abschnitt 7.1.

speziell Linux User Groups sich als politische Bewegung begreifen, oder ob ihnen das eher von „außen“ zugeschrieben wird. Folgen sie einem verinnerlichten praktischen Sinn, dann stellen sie sich dabei laut Bourdieu eben keine Fragen über die Einordnung ihres Handelns als politisches Handeln. Derartige Reflexionen würden höchstens von anderen, z. B. von Wissenschaftler_innen, an sie oder ihr Feld herangetragen. In der Tat beschäftigen sich etliche Autor_innen mit der Frage nach politischem Anspruch und Selbstverortung dieser Softwarebewegung; einige von ihnen ordne ich als Szenegänger_innen ein (etwa Glyn Moody, Christian Imhorst und Steven Levy, auf den Imhorst sich bezieht), ihre Auseinandersetzung mit dem Thema als eher populärwissenschaftlich — wenn sich diese Grenze auch nicht immer eindeutig ziehen lässt. Wie bei vielen Themen bezieht sich auch die Literatur zu dieser Frage entweder auf Richard Stallman oder allgemeiner auf Softwareentwickler_innen. Die Autor_innen deuten an vielen Stellen einen unter Entwickler_innen freier Software verbreiteten Liberalismus an, wobei dies jedoch kein homogenes Bild ist.

Die Beobachtung, dass vor allem der Gründer der → *GNU*-Projektes Richard Stallman die moralische, politische und ökonomische Kontextualisierung von freier Software stark macht, teilen verschiedene Autor_innen (vgl. Grassmuck 2004: 226; Moody 2002: 29²³; Yeats 2006: 57 ff.): Stallman argumentiert für freie Software, indem er kritisiert, dass Softwarepatente die Nutzungsmöglichkeiten von Programmen einschränkten. Daraus resultiere, dass sich Programmierer_innen einander gegenüber unethisch verhalten müssten:

„I consider that the golden rule requires that if I like a program I must share it with other people who like it. Software sellers want to divide the users and conquer them, making each user agree not to share with others. I refuse to break solidarity with other users in this way. (...) The fundamental act of friendship among programmers is the sharing of programs; marketing arrangements now typically used essentially forbid programmers to treat others as friends. The purchaser of software must choose between friendship and obeying the law“ (Stallman 1985).

In seinem Vortrag nennt er den Teil, in dem er die vier Freiheiten von freier Software erläutert, die „ethical perspective“. In der daran anschließenden „perspective of business“ grenzt er sich vom ihm häufig zugeschriebenen Kommunismus (vgl. z. B. Yeats 2006) ab

²³Glyn Moody stellt in seinem Buchtitel *Rebel Code. The Inside Story of Linux and the Open Source Revolution* durch die Wortwahl von „Rebel“ und „Revolution“ eine politische Lesart des Titels her, die sich in seinem Buch nicht unbedingt wiederfindet.

und bekennt sich zu freiem Markt und Privateigentum. Freie Software bringe einen „free market for all sorts of services“. Statt um neue Features bitten zu müssen, könnten sich interessierte Firmen in Organisationen zusammentun und Geld zusammenlegen, um Programmierer zu bezahlen, die ihnen ihre Wünsche umsetzen (Sta 2008). Christian Imhorst bezeichnet die → *GNU General Public License* als „politischen Ausdruck des *free spirit* in der freien Software-Bewegung“ (Imhorst 2005: 284). Geistiges Eigentum solle durch die Lizenz auf eine bestimmte Weise geschützt werden, nämlich so, dass Produkte wie freie Software Eigentum aller, nicht Einzelner seien (vgl. ebd.).

Stallmans Aussagen werden politisch sehr unterschiedlich eingeordnet: Dave Yeats nennt Stallmans Gedanken kommunistisch, weil ihm das uneingeschränkte Teilen von Code wichtiger sei, als für die Softwareentwicklung bezahlt zu werden, und er dies auch von anderen Entwickler_innen behauptete (vgl. Yeats 2006: 57 f.).²⁴ Imhorst nennt Stallman einen Vertreter eines „genossenschaftlichen Anarchismus“ (Imhorst 2005: 291). Stallmans Forderung nach einer „neuen Ordnung des geistigen Eigentums im Sinne der Hackerethik“ (ebd. 284) ordnet er in einen spezifischen US-amerikanischen Anarchismus ein.²⁵ Den amerikanischen Anarchisten sei es nie um die Abschaffung aller Autoritäten gegangen, sondern um die Nichteinmischung des Staates in ihre Angelegenheiten. Imhorst spricht von einer Spaltung in linke Anarchist_innen, die genossenschaftliche Organisationsformen favorisierten, und rechte „Anarchokapitalist_innen“, die schlichtweg ohne Eingriffe von Autoritäten über ihr Geld verfügen wollten. Stallman stehe für den „genossenschaftlichen Anarchismus“, während die *Open Source Initiative* um Eric Raymond dem „marktwirtschaftlichen Anarchismus“ zuzuordnen sei (Imhorst 2005: 286 ff.; ausführlicher Imhorst 2004).

Diese Interpretation ist in der Szene höchst umstritten, wie ein Blick in die oben erwähnten *Anarchist FAQ* zeigt. Die Verfasser_innen nennen als ihren Anstoß, die *FAQ* aufzuschreiben nämlich genau die Vereinnahmung des Anarchismusbegriffes für eine Interpretation, die Anarchismus und Kapitalismus vereinbar machen solle.²⁶ „Anarcho’-

²⁴In seiner Analyse des *GNU Manifesto* kommt er dann jedoch zu dem Ergebnis, dass Stallmans Manifest sich an eine technisch versierte Elite richtet: „Given Stallman’s apparent attraction to Marxist thoughts, it’s surprising that he would so easily include certain groups while excluding others. Essentially, Stallman contends that sharing is fundamental, but only for developers“ (ebd. 63).

²⁵Imhorst trennt diesen Anarchismus-Begriff nicht klar von seinem ebenfalls verwendeten Begriff des „amerikanischen Radikalismus“. Auch die begrifflichen oder ideellen Unterschiede zum Liberalismus werden nicht deutlich.

²⁶<http://www.infoshop.org/page/AnarchistFAQIntro>, Abruf: 03.02.2011

capitalists claim to be anarchists because they say that they oppose government“²⁷. Für die Verfasser_innen der *FAQ* bedeutet Anarchismus jedoch viel mehr als das, nämlich auch eine deutliche Opposition gegen hierarchische Ausbeutungs- und Unterdrückungsverhältnisse und eine klare Kapitalismuskritik.²⁸ Sie verweigern allen Befürworter_innen des Kapitalismus die Bezeichnung „Anarchist_in“, weil sie die politischen und ökonomischen Vorstellungen des Anarchismus nicht teilten und alle Elemente und Vertreter_innen des Anarchismus, die ihnen nicht passten, ausschlossen. Anstatt gegen Autorität, Hierarchie oder den Staat zu kämpfen, wollten sie die entsprechenden Logiken für ihre Zwecke privatisieren.²⁹ In den *FAQ* wird keine Verbindung zu freier Software als anarchistischem Projekt hergestellt. In dem *Debian*-Paket wird lediglich darauf hingewiesen, dass diese Dokumentation selbst freie Software sei, zu der man beitragen könne.³⁰

Gabriella Coleman hat in ihrer Untersuchung von Politikverständnissen der „Free and Open Source Software“-Bewegung einen Wandel beobachtet. In einem Artikel von 2004 arbeitet sie eine klare Abgrenzung der Communities von politischen Anliegen heraus und erklärt genau mit dieser Abstinenz die erfolgreiche Verbreitung von freier Software. Die Software sei für so verschiedene gesellschaftliche, politische und wirtschaftliche Interessengruppen anschlussfähig, *weil* sie nicht mit expliziten politischen Anliegen verbreitet würde, sondern weil Lizenzen wie die → *GNU General Public License* politisch neutral gehalten seien. Die Bewegung wende sich gleichzeitig mit diesen Lizenzen dagegen, aus allem Eigentum zu machen, und habe durch den Umgang mit dem eigenen → *Quellcode* letztlich doch eine politische Reichweite, ein implizites politisches Programm: Sie habe den etablierten Umgang mit geistigen Eigentumsrechten nachhaltig irritiert (vgl. Coleman 2004: 508 f.).

An der *GPL* setzt einerseits Kritik an, andererseits ist sie ein Grund, aus dem politische Zusammenhänge freie Software nutzen. Sabine Nuss äußert kritisch, dass die ausdrückli-

²⁷<http://www.infoshop.org/page/AnarchistFAQSectionF1>, Abruf: 03.02.2011

²⁸„Thus, opposition to government is a necessary but not sufficient condition for being an anarchist – you also need to be opposed to exploitation and capitalist private property. As ‘anarcho’-capitalists do not consider interest, rent and profits (i.e. capitalism) to be exploitative nor oppose capitalist property rights, they are not anarchists.“ <http://www.infoshop.org/page/AnarchistFAQSectionF1>, Abruf: 03.02.2011

²⁹<http://www.infoshop.org/page/AnarchistFAQAppendix11#app5>, <http://www.infoshop.org/page/AnarchistFAQSectionF1>, Abruf: 03.02.2011

³⁰Auf meine E-Mail-Anfragen an den *Debian*-Paketmaintainer und die Verfasser_innen der *FAQ*, wie es dazu kam, dass die Dokumentation ein eigenes *Debian*-Paket wurde, habe ich leider keine Antworten erhalten. Stefan Meretz, einer der Autor_innen des Blogs *keimform.de* mutmaßt dazu: „Warum aber gibt’s das überhaupt, und dann noch als offizelles *Debian*-Paket? Weil es die Bibel gibt, auch bei *Debian*, also zum Ausgleich. Die beiden neutralisieren sich gewissermaßen: Jede Löschaufforderung des einen, wird mit der Existenz des anderen abgelehnt“ (<http://www.keimform.de/2007/apt-get-install-anarchism/>).

che politische Neutralität, gepaart mit entsprechenden Lizenzen, es eben auch politisch eventuell unerwünschten Gruppen oder Konzernen ermögliche, freie Software für ihre Zwecke zu nutzen: „Schließlich ist es dem abstrakten ‚frei‘ nicht anzusehen, welche Rolle der jeweilige Code im gesellschaftlichen Kontext spielt. Es ist meines Erachtens nicht völlig beliebig, ob Freie Software/Open Source eingesetzt wird in einem Atomkraftwerk oder in einer Anti-Atom-Initiative (...)“ (Nuss 2006: 234). Andererseits macht die Lizenz freie Software für politische Gruppen oder Projekte ideell attraktiv, denen die Produktionsbedingungen der Produkte, die sie nutzen, wichtig sind. So schreibt Patricia Jung: „Der Open-Source Gedanke als solcher steht anderen emanzipatorischen Projekten durchaus nahe und verschafft Open-Source-Software z. B. in Entwicklungshilfe- aber auch in Frauenprojekten einen Sympathiebonus gegenüber *closed source*“ (Jung 2006: 237). Coleman / Golub konstatieren umgekehrt auch, dass die linke Globalisierungskritik der letzten Jahre Sympathien in der Hacker_innenszene geweckt hat (vgl. Coleman/Golub 2008: 271). Freie Software ist also für verschiedene Diskurse *anschlussfähig*, unter anderem für linkspolitische oder emanzipatorische Zusammenhänge.

In einem Artikel von 2009 erkennt Coleman eine veränderte Situation: Die Programmierenden verstünden ihren Code, das ungehinderte Verfassen und Weitergeben von Code vermehrt als freie Meinungsäußerung (vgl. Coleman 2009). *Wie* es dazu kam, dass Quellcode und freie Meinungsäußerung zusammen diskutiert wurden — eine Assoziation, die es einige Jahre zuvor noch nicht gab —, zeigt sie anhand der Proteste auf, die mit der Verhaftung des Programmierers Jon Johansen 1999 einhergingen (vgl. Coleman 2009: 433 ff.).³¹ Die Schwierigkeit der Grenzziehung zwischen Code einerseits und Text als freie Meinungsäußerung andererseits wurde dadurch thematisiert, dass Menschen den rechtlich umstrittenen Code von Johansen, die dort verwendeten Algorithmen, in Poesie übersetzten. Diese Poesie war für Programmierer_innen rückübersetzbar (vgl. ebd. 439 ff.). Durch das breite Engagement im Protest gegen die Verhaftung hätten die Entwickler_innen zentrale Grundsätze der liberalen Tradition neu „konfiguriert“. Über die konkrete Verteidigung der eigenen produktiven Autonomie hinaus sei es dabei auch um grundlegende demokratische Bürgerrechte gegangen (vgl. ebd. 449). Inzwischen bescheinigt Coleman den Entwickler_innen freier Software eine extreme Versiertheit in rechtlichen Fragen und ein politisches Bewusstsein, das klar der freien Meinungsäußerung verpflichtet ist (vgl. ebd. 421). Auch wenn Coleman nicht von einem Politisierungsprozess

³¹Der damals 16-jährige Johansen hatte die Software *DeCSS* publiziert, mit der es möglich wurde, DVDs unter Linux anzusehen, die durch → *Digital Rights Management* geschützt waren und zuvor nur unter *Microsoft Windows* oder *Mac OS X* liefen (vgl. ebd. 435).

spricht, ordnen die von ihr untersuchten Programmierenden ihre Tätigkeit offenbar in jüngerer Zeit anders in den sie umgebenden politischen und sozialen Kontext ein. In einem dritten Aufsatz wiederum bezeichnen Coleman und ihr Co-Autor Alex Golub liberale Ideale als festen Bestandteil der Alltagspraktiken von Hacker_innen, ohne dabei eine homogene Hackerethik zu konstruieren: Vielmehr gebe es wiederkehrende liberale Elemente, die stets neu verhandelt würden, z. B. Verständnisse von Eigentum oder Privatsphäre. Liberalismus sei damit eher eine „kulturelle Sensibilität mit verschiedenen und teilweise konfligierenden Strängen“ (Coleman/Golub 2008: 256). In dieser „interconnected heterogeneity“ (ebd. 271) sei die Auseinandersetzung mit der Bedeutung von Freiheit ein konstitutives Moment für „moralische“ Diskurse in Hackerszenen (vgl. ebd. 257). Darüber hinaus könne aber weder von einheitlichen politischen Anliegen innerhalb von Projekten oder Bewegungen gesprochen werden, noch von einheitlichen Entscheidungen bezüglich der zu wählenden (z. B. rechtlichen) Mittel oder Wege.³²

Christopher Kelty kritisiert den Drang, die Szene einer politischen Strömung zuzuordnen zu wollen: Unter den Szenegänger_innen, die Kelty als → *Geeks* bezeichnet, gebe es lediglich den Konsens, sich selbst regieren zu wollen. Für sie stünden nicht Ideologien, sondern Praktiken im Mittelpunkt, die als Hinweise auf politische Aktivitäten ausgedeutet werden müssten. Die gemeinsamen Praktiken in Bezug auf Software begleiteten in der Szene stets die etablierten Formen politischen Ausdrucks (vgl. Kelty 2008b: 7 f.). Die Szenegänger_innen bezögen sich auf eine für Außenstehende befremdliche Mischung von Versatzstücken und Anknüpfungspunkten aus (Neo-)Liberalismus, Libertarismus, Anarchismus, Idealismus und Pragmatismus. Einordnungsversuche mit konventionellen politischen Philosophien würden der Szene jedoch nicht gerecht, weil sie die starke Rolle der Technologien nicht einbeziehen (vgl. ebd. 77). In seinen Augen ist gerade das Interessante, dass in FLOSS-Communities Neues entwickelt wird, was auch eine Irritation der althergebrachten Kategorien impliziere. Politik sei für die Szenegänger_innen eng mit den technischen Details des Internet verknüpft, ebenso wie mit den Akteur_innen in diesem Bereich, mit Organisationen, Gesetzen und Personen. Die Szene bestehe damit aus Be-

³²Coleman und Golub beziehen sich beispielhaft auf die Cypherpunk-Bewegung, deren Anliegen die Verbreitung von Verschlüsselungstechnologien war (vgl. 7.2.1). „Cypherpunks’ pessimism regarding the intrusive nature of government and corporations is neither politically Left nor Right — its suspicion of the industrial-military complex falls as easily within the libertarian Right as it does a certain anti-military Left-pacifism. And while some of the most adept practitioners are often libertarian loyalists who hold a faith in free-market capitalism, the loose association of Cypherpunks professes no outward political affiliation. As a result, crypto-freedom practices, groups and events include people with divergent political viewpoints“ (Coleman/Golub 2008: 260).

triebssystemen *und* sozialen Systemen. Die Beteiligten erschlossen sich *über* die historisch spezifische Situation des Internet allgemeine Probleme von Wissen, Macht, Freiheit, Aufklärung, Fortschritt und Intervention (vgl. ebd. 93 f.)

Ein anderer Diskussionsstrang behandelt die politische Frage, welche Bedeutung freie Software für die Veränderung der Produktion habe. Hier geht es um Praktiken des Produzierens in einem kapitalistischen System, die den etablierten Logiken zu widersprechen scheinen: Menschen stellen freiwillig, selbst organisiert und oft unbezahlt gute Software her und stellen sie der Allgemeinheit zur Verfügung. Für diese Produktionsweise hat sich der Begriff *peer production* etabliert, den Yochai Benkler 2002 geprägt hat (vgl. Benkler 2002).³³ Die Möglichkeiten und Grenzen der *Peer production* werden seit Jahren in der *Oekonux*-Mailingliste und dem → *Weblog keimform.de* diskutiert. Aus diesem Diskussionszusammenhang kommt auch Christian Siefkes, der in seinem Buch *From Exchange to Contributions. Generalizing Peer Production into the Physical World* durchspielt, ob und wie sich diese Produktionsweise jenseits von immateriellen Gütern verwirklichen lässt, als Produktionsweise für eine gesamte Gesellschaft (vgl. Siefkes 2007). Auch Dmytri Kleiner von der Berliner Gruppe *Telekommunisten* bezieht sich auf Benkler. Die *Telekommunisten* entwickeln Software, Kleiner hat dies jedoch inhaltlich mit einer marxistisch inspirierten Kritik an der zentralisierten Netzwerktopologie des Internet verknüpft. Das heutige kommerzialisierte Internet gehorche vom ganzen Aufbau her der kapitalistischen Logik, da es sehr stark nach dem → *Client*- → *Server*-Prinzip aufgebaut sei: Im Gegensatz zu früher kommunizierten Netzteilnehmer_innen nicht mehr direkt miteinander, sondern (fast) die gesamte Kommunikation laufe über die zentralisierten Server von Diensteanbietern.³⁴ Die Nutzer_innen schufen mit ihren Inhalten den Mehrwert dieser Plattformen.³⁵ Kleiners Vorstellung ist die einer klassenlosen Gesellschaft von ebenbürtigen Produzierenden, die er in seinem Konzept von „venture communism“ verwirklicht sieht. „Venture-Kommunen“ sind bei Kleiner firmenähnliche Organisationen, die einen Bestand an Pro-

³³Benkler beleuchtet dort er aus einer *rational choice*-Perspektive das Phänomen, dass Tausende von Softwareentwickler_innen freiwillig und unbezahlt an gemeinsamen (und erfolgreichen) Projekten arbeiten.

³⁴Zur Veranschaulichung: Informationen können zwischen zwei vernetzten Computern theoretisch direkt übermittelt werden. Die Kommunikationsteilnehmer_innen können direkt miteinander chatten oder Dateien übermitteln. Die großen Anbieter von Onlinediensten sind jedoch in der heutigen Kommunikationspraxis meistens dazwischengeschaltet: Chat läuft beispielsweise über die Server von Facebook, ebenso wie das „Teilen“ anderer Inhalte.

³⁵„Web 2.0 is a business model of private capture of community-created value. (...) The real value of YouTube is not created by the developers of the site; rather it is created by the people who upload videos to the site“ (Kleiner 2010: 17).

duktionsmitteln verwalten und die Nutzung dieser Ressourcen durch die Mitglieder organisieren (vgl. ebd. 23 ff.).

Eine sehr kritische, linke Perspektive wirft Sabine Nuss auf das Thema. Sie fragt nach dem subversiven Potenzial von freier Software und kommt zu dem Schluss, dass die Entwickler_innen nicht das Anliegen hätten, den Kapitalismus zu überwinden, oder sich sogar explizit davon distanzieren (vgl. Nuss 2006: 231 f.). Stallman bringe seine Kritik an proprietärer Software nicht mit kapitalistischen Verhältnissen in Verbindung, sondern er kritisiere es lediglich als unmoralisch, wenn Privateigentum dazu genutzt würde, die Zusammenarbeit zu behindern (vgl. ebd.). Die Bewegung betone zwar ihren Freiheitsbegriff. Sie meine damit jedoch nicht die Emanzipation des Menschen vom kapitalistischen Arbeits- oder Akkumulationszwang, sondern sie meine „jene Freiheit, die (...) ganz der bürgerlichen Produktionsweise entspringt: die Freiheit, mit ‚seinem Eigentum‘ [zu] verfahren (...) wie man möchte“ (Nuss 2006: 232). Dies spiegele sich in freien Lizenzen wie der *GNU GPL* wider, die nicht auf Konzepten von Gemeineigentum aufbauten, sondern „auf der im Urheberrecht verankerten ausschließlichen Verfügungsgewalt des Rechteinhabers“ (ebd. 235), der sich individuell dazu entscheide, sein_ihr Recht in einer anderen Weise zu nutzen. Für die Institution des bürgerlichen Eigentums selbst hätten die meisten Programmierer_innen kein kritisches Bewusstsein. Trotz der Lizenzen, die schon eine besondere Eigentumsform darstellten, sei freie Software voll in kapitalistische Verwertungsprozesse integriert worden: Die künstliche Verknappung eines Angebots sei hier nur vom → *Quellcode* auf die Dienstleistungen rund um die Software verlagert worden, die *das* Geschäftsmodell in diesem Bereich seien. Dass freie Software die kapitalistischen Produktionsverhältnisse und Verwertungsprozesse nicht angreife, sondern ihnen entgegenkomme, zeige auch die verbreitete Argumentation, dass sie durch die Verhinderung von Wissensmonopolen den Wettbewerb stärke und dadurch ein großes Innovationspotenzial habe (vgl. ebd. 232 ff.). Zuletzt thematisiert Nuss die Arbeitsorganisation von Entwickler_innen freier Software, die extrem prekär sei:

„Zugleich wird die bei Freier Software/Open Source intensive Kooperation als kostenlose Produktivkraftentwicklung genutzt, indem projektbezogene, über [nationale, S. M.] Grenzen gehende, arbeitsintensive, vertragslose und freiwillige Arbeit teilweise sogar unbezahlt auf sich gezogen wird. Loyalitätsversprechen, Community-Geist und die Werbung mit sozialem Kapital (Reputation) sind neben dem Interesse an der Lösung des Programmierproblems selbst die Mittel, diese Arbeitskräfte für ein Projekt kostengünstig zu begeistern“ (Nuss 2006: 237 f.).

Die Software als Produkt und die Arbeitsbeziehungen sieht Nuss demnach eher als Wegweiser für eine neoliberale Umgestaltung des Arbeitsmarktes im informationellen Kapitalismus. Was sie der freien Software zubilligt, ist, dass sie aufzeige, dass andere Formen der Produktion möglich sind und dass damit prinzipiell auch Eigentumskonzepte hinterfragbar werden — allerdings nur, wenn die Praxis auch in einen entsprechenden Diskurs eingebettet sei (vgl. ebd. 239 ff.).

Dieser Exkurs hat gezeigt, dass die Szene durchaus mit politischen oder politökonomischen Anliegen in Verbindung gebracht und daraufhin untersucht wird. Um an Sabine Nuss anzuschließen: Meine Erhebungen haben *nicht* ergeben, dass potenzielle politische Anliegen oder Praktiken von LUGs in einordnende oder unterstützende Diskurse eingebettet wären. Die untersuchten Linux User Groups sehen sich zumindest nicht in einem Maße politischen Zielen verpflichtet, dass sie gemeinsam darüber diskutieren würden.³⁶ Es *gibt* sowohl Software-Entwickler_innen, die ihre Arbeit in einen politischen Kontext oder in Relation zu gesellschaftlichen oder ökonomischen Bedingungen sehen, als auch solche, die derlei Zusammenhänge *nicht* herstellen. In jedem Fall spielt das bei der gemeinsamen Aneignung von Linux-Nutzungspraktiken in LUGs keine Rolle. Die Verfasser_innen von Literatur über die Szene widmen sich solchen Kontextualisierungen freier Software viel stärker als diese Communities selbst es in der Praxis tun. Somit kann auch festgehalten werden, dass Praktiken in Verbindung mit freier Software dadurch politisch werden, dass sie als politisch interpretiert werden. Mit Bourdieu gesprochen ist dabei aber zu betonen, dass solche herangetragenen Interpretationen von Praxis unabhängig von der Logik der Praxis selbst sind (vgl. Bourdieu 1999: 157 ff.).

Meine Beobachtungen der Praxis selbst führen mich zu folgender Interpretation: Es gibt bei vielen, aber nicht allen Beteiligten in LUGs ein Bewusstsein für den Unterschied zwischen freier und Open-Source-Software, über den gelegentlich diskutiert wird. Die sympathisierenden Anspielungen auf grundsätzlich andere politische Systeme, die der Software in Form von Programmnamen eingeschrieben werden, verbleiben jedoch auf der Ebene von Humor; sie finden in Praktiken von FLOSS-Communities keinen sichtbaren Niederschlag und sind auch kein Gesprächsthema. Coleman liest aus dem Kampf von Softwareentwickler_innen für die Anerkennung von Code als freier Meinungsäußerung

³⁶Damit will ich auf keinen Fall Einzelperson ihr politisches Interesse absprechen. Im Gruppenzusammenhang der LUGs wird es jedoch nicht thematisiert.

einen Kampf für Bürgerrechte heraus³⁷ — eine Deutung, die ich im Anschluss an Sabine Nuss (2006: 232) hinterfragen möchte: Meine These ist, dass die Entwickler_innen freier Software — wie Stallman — in erster Linie uneingeschränkt mit dem Quellcode hantieren können möchten. Stoßen sie dabei auf rechtliche Hürden, dann kann die Argumentation mit Redefreiheit ein Mittel sein, um ihre persönliche Freiheit durchzusetzen. Das macht sie in meinen Augen noch nicht zu Kämpfer_innen für Grund- und Menschenrechte mit einem Gesellschaftsentwurf, in dem kollektive Elemente stark wären. Hinzu kommt, dass sie sich mit einer Forderung nach freier Meinungsäußerung auf eine demokratische Grundlage zurückziehen, über die kein Konsens ausgehandelt werden muss, weil sie als Teil des Grundgesetzes zumindest formal rechtlich gesichert ist. Coleman / Golub stellen richtig fest, dass die sehr heterogenen Zusammenhänge, in denen freie Software entwickelt (und genutzt, S.M.) wird, mit Freiheitsdiskursen ein konstitutives Moment teilen (vgl. Coleman/Golub 2008: 257). Sie damit schon als Bürgerrechtler_innen zu bezeichnen, erlaubt es jedoch auch, die Produktions- und Besitzverhältnisse, die Verteilung von Wissen und die Sozialisation in einen technikaffinen Habitus auszublenken, die in der Praxis sozial relevant sind. Verschiedene politische Ansichten, aber auch soziale Ungleichheiten und Machtverhältnisse bleiben dadurch unproblematisiert. Meine Kritik bezieht sich in erster Linie auf die Forschung, die sich mit den politischen, sozialen und ökonomischen Implikationen von freier Software auseinandersetzt, da die untersuchten Gruppen sich selbst nicht als politisch verstehen. Sie äußern nirgendwo einen Anspruch auf politische Interventionen. Ihr Anliegen ist vielmehr die Verbreitung von freier Software durch kostenlose Bildungsangebote und Hilfestellung, die Förderung von Wissenschaft, Forschung und Bildung im Bereich der freien Software, wie es in der Vereinssatzung einer LUG ausgedrückt wird.

An dieser Stelle möchte ich ein Teilergebnis von Kapitel 7 vorwegnehmen, das im Widerspruch zu diesem Humor zu stehen scheint. Es geht um die alltäglichen Arbeitspraktiken von Linux-Systemadministrator_innen: Sie arbeiten zur Absicherung von Rechnern nämlich laufend mit klaren Klassifizierungen, etwa durch Zugriffsrechte oder Firewalls. Dadurch implementieren sie viele technischen Restriktionen und hantieren mit der eindeutigen Zuordnung von Menschen zu Accounts und Dateien, die Nutzer_innen nicht einfach umgehen können, wenn ihnen diese Regulierungen nicht gefallen. Diese Praktiken des Formalisierens und Hierarchisierens erscheinen als unvereinbar mit den humorvollen

³⁷Dieser Kampf findet durchaus an anderen Orten statt, wie z.B. auf den *Freiheit statt Angst*-Demonstrationen, nicht jedoch in Linux User Groups.

Anspielungen auf Subversion, Anarchie oder auch Basisdemokratie: Die Ablehnung von Autoritäten und ihre Reproduktion in der Systemadministration prallen aufeinander.³⁸ Der subversiv angehauchte „anti-establishment“-Humor (Friedman/Friedman 2003) von FLOSS-Communities muss deshalb und aufgrund des zuvor Gesagten meiner Interpretation nach als *symbolische* politische Auflehnung gelesen werden, die nicht unbedingt impliziert, dass diejenigen, die sie äußern, derart grundlegende Veränderungen des politischen Systems auch wollen und praktizieren oder sich aktiv für sie einsetzen oder sich aktiv für sie einsetzen würden.

Zum Schluss noch einige Gedanken zu möglichen Gründen, sich in der LUG nicht in politischen Diskussionen zu ergehen. Bei den Treffen steht die Beschäftigung mit der Technik im Mittelpunkt. Die Beteiligten tauschen sich in erster Linie über Neuerungen und Probleme im Zusammenhang mit Linux aus. Darüber hinaus kannten sich die LUG-Mitglieder (selbst die formalen Vereinsmitglieder) teilweise nicht oder nur schlecht³⁹, was ein Grund dafür sein kann, nicht miteinander über konfliktträchtige politische Themen zu diskutieren. Eine Einzelperson äußerte in der Raucherpause durchaus, dass sie Linux aus einer Kritik am Neoliberalismus nutze. Insgesamt ist die LUG aber kein Ort für derartige Auseinandersetzungen. Es ist auch möglich, dass ein vermuteter Konsens herrscht, oder zumindest eine angenommene Zustimmung zu grob liberalen politischen Einstellungen oder Ansichten, die aber nicht ausdiskutiert werden.

4.4. Vergeschlechtlichte Stereotype und sexistische Witze

Zurück zu Sprache und Humor in LUGs: Um „dazuzugehören“ reicht es nicht aus, die technisierte Sprache zu sprechen und den kulturellen Kanon zu kennen. Wie feine Unterschiede gemacht werden, zeigt sich etwa in einem Vortrag in der LUG, wo die Referentin A82 ein *Apple*-Notebook nutzt. Über den Beamer ist der Name des Rechners zu sehen, *zweiundvierzigsbaby*, eine Bezugnahme auf die „42“. Der Organisator des Abends stellt die Referentin als „Fachfrau/Fachmann“ vor. Er lässt „Fachfrau“ nicht unergänzt stehen.

³⁸Vgl. auch die ähnlich widersprüchlichen Beobachtungen von Kehily/Nayak zum subversiv anmutenden Humor unter (männlich sozialisierten) Schülern. In ihrem Humor lehnten sie sich gegen Autoritäten auf. Wenn es jedoch um Geschlechtsidentitäten und Zuordnungen in der heterosexuellen Matrix gehe, reproduzierten sie strikt klassische Normen (vgl. Kehily/Nayak 1997: 70).

³⁹Als es beispielsweise um die Aufgabenverteilung für ein Community-Event ging, stellte sich heraus, dass sie sich nur teilweise mit Namen kannten.

A82 sorgt damit gleich zu Beginn ihres Vortrages für eine Irritation; sie stellt eine Abweichung vom „Fachmann“ dar und wird sprachlich zur „anderen“ gemacht. Sie antizipiert darüber hinaus, dass der *Mac*⁴⁰ nicht unkommentiert bleiben wird und nimmt dem Publikum die Kritik vorweg: Sie habe den *Mac*, weil darauf alle drei Betriebssysteme liefen, was sie brauche, um den Anforderungen ihrer verschiedenen „Kunden“ nachkommen zu können. Andere, nicht sie selbst, sind der Grund, aus dem sie kein reines Linux-Notebook betreibt (LUG 2008f).⁴¹ Meiner Einschätzung nach hat sie die *illusio* des Feldes insoweit verinnerlicht, dass sie die Kommentare kommen sieht und weiß, dass sie sich durch den Rechner unterscheidet. Sie teilt jedoch nicht den Habitus vieler Community-Mitglieder, nach dem sie *auf jeden Fall* Linux als Hauptbetriebssystem auf ihrem Rechner installiert hätte. Welche Hardware und welches Betriebssystem jemand nutzt, wird hier zu einem sozial relevanten Unterschied gemacht: *Apple*-Nutzer_innen sind in der Regel keine Linux-Nutzer_innen (auch wenn es möglich ist, auf einem *Mac* Linux zu installieren). In dieser Sequenz wird deutlich, dass A82 in ihrem LUG-Vortrag mit den Kategorien Gender und Technologie (Hardware und Betriebssystem) konfrontiert wird, die Differenz schaffen. Sie entspinnt sich an der Bezeichnung als „Fachfrau/Fachmann“ durch den Organisatoren und wird von A82 selbst in Bezug auf ihr Gerät fortgeführt. Sie kommt um diesen Prozess nicht herum, obwohl sie den Kanon der Community kennt, die Sprache spricht.⁴² Sie rekurriert später in dem Vortrag sogar selbst auf ein vergeschlechtlichtes Stereotyp einer technisch nicht versierten Frau: „Tante Tillie“ ist ein Bezug zu einem Text von Eric Raymond. Der Mitbegründer freier Software beschreibt dort, wie benutzerunfreundlich die Druckereinrichtung unter Linux ist. Der Artikel ist teilweise polemisch, Raymond hat ihn geschrieben, weil er selbst nicht mit der Konfiguration von Netzwerkdruckern zurecht kam. Dennoch bezieht er immer wieder eine imaginäre „Tante Tillie“ ein um zu belegen, dass die Software für technisch wenig versierte Menschen, zu denen er sich selbst nicht zählt, kaum bedienbar ist:

„If I were Aunt Tillie the archetypal nontechnical user, I am at this point thinking ‚What in the holy fleeping frack does that mean?‘ (...) The help is... unhelpful. Aunt Tillie, at this point, is either resigning herself to another session of being tortured by

⁴⁰Die Notebooks der Firma *Apple* werden mit dem proprietären Betriebssystem *Mac OS X* ausgeliefert, so dass die Besitzer_innen in der Regel keine Linux-Nutzer_innen sind. (Es ist möglich, darauf Linux zu installieren, die meisten haben den Rechner jedoch wegen des Betriebssystems *Mac OS X*.)

⁴¹In einem Gespräch mit A9, wenige Tage nach diesem Vortrag, formulierte er die gleiche Einschätzung, dass A82 sich mit ihrem *Mac* gleich in einer Verteidigungsposition befunden habe (A9- 2008).

⁴²A82 ist seit Jahren in einer bestimmten Community aktiv und für ihre Expertise auf dem Gebiet recht bekannt.

4. Eine Sprache sprechen

the poor UI⁴³ choices of well-meaning idiots or deciding to chuck this whole Linux thing and go back to the old Windows box. It blue-screened a lot, but at least it allowed her the luxury of ignorance — she didn't have to know, or care, about what a JetDirect or a CUPS might be.

I am not ignorant, but I have my own equivalent of Aunt Tillie's problem. I know I want one of the top two methods, but I don't know which one. And I don't want to know or care about the difference either; I have better things to do with my brain than clutter it with sysadminning details. If the tool can detect that both methods are available on the local net (...) it should put '(recommended)' next to one so I can click and keep going" (Raymond 2006).

Raymond schreibt, dass er sich von einem Assistenten zur Druckereinrichtung Empfehlungen wünscht, falls er eine Entscheidung aus Unwissenheit nicht selbst treffen kann. Er möchte etwas anklicken und weitermachen. ‚Tante Tillie‘ schreibt er das gleiche zu: Sie wünsche sich den Luxus der Unwissenheit. Der Unterschied ist für ihn, dass *sie* an dem Punkt aufhöre, Linux zu benutzen und zu Windows zurückkehre, das oft abstürze, aber den Nutzer_innen diesen Luxus der Unwissenheit gönne. *Er* hingegen löse das Problem irgendwann und trage durch seinen Text Tipps und Leitfragen für Softwareentwickler_innen in der Community bei.

A82 spricht die Community-Sprache bis zu dem Punkt, dass sie selbst das durch Raymonds Text reproduzierte und breit bekannte Geschlechterstereotyp aufgreift, in dem eine als weiblich vorgestellte und eventuell auch ältere Person („Tante“) die technisch nicht versierte Computernutzerin repräsentiert (vgl. Ratto 2003). In Anbetracht dessen, dass sie selbst von der Differenzierung über die Kategorie Geschlecht betroffen ist, erscheint dies widersprüchlich. Anita Thaler fasst dieses Verhalten als „coping strategy for women“ auf: „[N]ot only do women laugh about difficult situations and also about jokes at their expenses in order to prove their sense of humour. Women are also making gender-related jokes themselves“ (Thaler 2008). Thaler findet einen Erklärungsansatz dafür bei Helga Kotthoff, die darauf verweist, dass ein Austausch ironischer Kommentare und Witze und das gemeinsame Lachen unter Umständen trotz der damit markierten Geschlechtergrenzen ein Gruppengefühl herstellen kann (ebd.).⁴⁴ Eine ähnliche verbreitete Repräsentation von

⁴³UI steht für „User Interface“, Benutzeroberfläche.

⁴⁴Anita Thaler zitiert aus Kotthoff, Helga (2005): Gender and joking: On the complexities of women's image policies in humorous narratives, in: Journal of Pragmatics, 32, S.55-80. Thaler identifiziert zwei weitere Zwecke, die gender-bezogene Witze situativ erfüllen können: Erstens können Frauen sowohl Auslöserinnen sexistischer Witze sein als auch „zivilisierende Faktoren“, wenn diejenigen, die die Witze machen,

unterschiedlichen Anwender_innen oder Anwendungsbereichen, die auch in dem Vortrag (und anderen beobachteten Szenen) vorkommt, ist die von technisch inkompetenten weiblichen Sekretärinnen⁴⁵:

Als ich gerade mal nicht so genau aufpasse, fließt das Bild von der Sekretärin ein, das mir im Zusammenhang mit Software schon x mal begegnet ist. Die Sekretärin wolle ihre Benutzeroberfläche einfach und immer gleich aussehend haben, eben keine Power Userin (LUG 2008f).

Sie reden dann über Windows im Anwenderbereich und A1 plädiert dagegen, Leute von Windows abbringen zu wollen. Er malt das Bild, ‚die Tippse‘ wolle, dass alles haargenau so aussieht wie zu Hause auf ihrem Rechner, und ‚der Chef‘ wolle schnell über Outlook Termine verteilen (LUG 2008j).

Mary Jane Kehily and Anoop Nayak identifizieren in einer Arbeit über die Produktion von heterosexuellen Männlichkeiten und Hierarchien in Schulen das kollektive Erzählen und Wiederholen von Geschichten vor Publikum als eine Praxis zur Produktion und Konsolidierung heterosexueller Männlichkeiten (vgl. Kehily/Nayak 1997: 76). Die an den LUG-Treffen Beteiligten erzählen hier zwar keine gemeinsamen Erlebnisse nach, sie rekurrieren jedoch mit einer ähnlichen Wirkung für die Ingroup auf Stereotype: Sie konstruieren eine scheinbar homogene männliche Expertengruppe. Das Geschlechterstereotyp, das Frauen Technikkompetenz abspricht, gehört fest zum Wissensbestand der Communities: Frauen könnten linuxspezifische Probleme nicht lösen, wollten unwissend bleiben und sich nicht mit technischen Details auseinandersetzen, sie reagierten unflexibel auf anders aussehende Benutzeroberflächen, hätten im privaten Gebrauch Windows und wollten deshalb auf der Arbeit nichts anderes nutzen. Das Stereotyp wird in zahlreichen Situationen bemüht, obwohl die Technikkompetenz unter *allen* Beteiligten meist sehr unterschiedlich verteilt ist.

Besonders im Zusammenhang mit Witzen wird Sprache oft auf Geschlecht, und speziell auf weibliche Geschlechterstereotype bezogen. An dieser Stelle kommt meine persönliche

bei ihrem Auftauchen verstummen. Eine solche Szene habe ich wie folgt dokumentiert: „A41 sagte beim Rauchen irgendetwas Vulgäres (ich weiß es schon nicht mehr). Er sieht, dass ich hinter ihm komme und fügt hinzu, er hätte sowas nicht gesagt, wenn er gesehen hätte, dass eine Frau dabei ist“ (LUG 2008i). In beiden Fällen werde durch solche Situationen eine Trennung zwischen hegemonialen Männlichkeiten und weiblichem Publikum produziert. Zweitens seien sexistische Witze auch Strategien für Männer, mit einer Verunsicherung darüber umzugehen, was sie in Zeiten der *political correctness* korrekterweise über Gender sagen „dürfen“ (vgl. Thaler 2008).

⁴⁵Zur Wirkung dieses sehr verbreiteten Stereotyps bei der Entwicklung von Textverarbeitungssoftware vgl. Hofmann (1997), Hofmann (1999).

4. Eine Sprache sprechen

Position ins Spiel, denn meine Rollen als Beobachterin, als Teilnehmerin, als Frau, als Kritikerin vereinfachter binärer Zuschreibungen und als Erforscherin von Differenzierungsprozessen kollidieren. Als Ethnografin möchte ich das Feld verstehen, doch die Prozesse, die ich dort untersuche, treffen mich in bestimmten Momenten selbst, auch wenn sie sich nicht auf mich persönlich beziehen. In der folgenden Situation waren ca. sechs männlich sozialisierte Personen und ich im Raum.

A38 und A41 liefern sich einen Wortwechsel, in etwa so: Es geht um die Tastatur und wofür bestimmte Lämpchen an verschiedenen Thinkpads da sind. A38: „Bei meiner Frau kann man die Knöpfe ausmachen.“ A41: „Ich mach meine Frau an den Knöpfen an.“ A38: „Die → *Softkeys* von Frauen sind interessanter.“ (LUG 2008g)

A38 spricht zunächst vom Computer seiner Frau, A41 stellt rhetorisch eine Verbindung von Frauen und Knöpfen zu Brustwarzen und Sex her. A38 greift das auf und nennt weibliche Brustwarzen „*Softkeys*“, also Tasten, die kontextabhängig Verschiedenes bewirken. Während dies ein situativer Ausspruch war, ist ein anderes Beispiel sehr weit verbreitet, in dem auch ein Steuerelement des Rechners als weibliches Geschlechtsorgan bezeichnet wird: Thinkpads (die verbreitetsten Notebooks in FLOSS-Communities) und einige andere Notebook-Modelle verfügen über einen so genannten Trackpoint in der Tastatur, der Funktionen der Maus zum Navigieren übernimmt.

A44 steht jetzt hinter A1, sieht auf die Tastatur von A54s Notebook mit dem Loch zwischen den Tasten G und H. Er fragt: „War da mal ’ne Klit drin, die sich verpisst hat?“ A1 und A54 fragen nach und A44 stellt die Frage nochmal. A54 sagt laut: „Das war ’ne Zigarette“ (LUG 2009h).

Anderswo ist mir für den Trackpoint auch die Bezeichnung „Cyberclit“ begegnet. In dem in der Szene sehr beliebten Comicblog *xkcd* wird dies karikiert: Unter dem Titel „Appropriate Term“ ist eine Zeichnung des Trackpoints in der Tastatur. Sie ist überschrieben mit „How to refer to the pointer thing on laptop keyboards“. Darunter ist eine Skala für den Umgangston zwischen „formell“ und „informell“ aufgemalt. Von „Trackpoint“ verschiebt sich die Bezeichnung mit wachsender Informalität des Gesprächs von „Noppe“ über „Brustwarze“ zu „Klitoris“.⁴⁶ Hinter beiden Beispielen steht der Vergleich von Knöpfen an Computern mit weiblichen Geschlechtsorganen. Diese Knöpfe werden von (heterosexuellen) Männern bedient: Es wird eine Assoziation hergestellt zwischen Männern,

⁴⁶<http://xkcd.com/243/>

die Computer über bestimmte Hardware bedienen und Männern, die Frauen sexuell erregen — und indirekt wird den Computern und Frauen eine passive Rolle, ein Gesteuertwerden, zugeschrieben. Unter den bereits erwähnten Fanartikeln ist ein T-Shirt, das mir besonders aufgefallen ist, weil es dieses Steuern in Form von Wortspielen mit Befehlen darstellt. Auf dem T-Shirt wird mit den Namen von → *Kommandozeilen*-Programmen gespielt, um Sex zu beschreiben. Es trägt folgenden Aufdruck:

```
Linux is sexy!  
guru@linux:~> who | grep -i blonde | talk;  
cd ~; wine; talk; touch; unzip; touch;  
strip; gasp; finger; mount; fsck; more;  
yes; gasp; umount; make clean; sleep;
```

Tafel 4.1: T-Shirt-Aufdruck „Linux is sexy!“

Eine technische Lesart derselben Aufschrift würde den Zweck der Programme einbeziehen. Ihre groben Funktionen sind in Tabelle 4.1 aufgelistet. Der Nutzer „Guru“ gibt eine Folge von Befehlen in seine → *Shell* ein. Die Programme `mount`; `fsck` (als Anspielung auf das Wort „fuck“) und später `umount` beschreiben den Akt der Penetration: Binde ein → *Dateisystem* ein, überprüfe es auf seine Integrität hin, repariere es, wenn es beschädigt ist und hänge es danach wieder aus!⁴⁷

Ein Beispiel dafür, dass Community-Mitglieder Geschlecht als Diskriminierungskategorie reflektieren, sind die oben erwähnten Frequently Asked Questions (FAQ) zum Anarchismus. Die Verfasser_innen gehen explizit auf ihre Sprachverwendung ein.

„Language has changed a lot over the years (...). The use of the term 'man' to refer to humanity is one such change. Needless to say, in today's world such usage is inappropriate as it effectively ignores half the human race. For this reason the FAQ has tried to be gender neutral. However, this awareness is relatively recent (...). When we are quoting past comrades who use 'man' in this way, it obviously means humanity as a whole rather than the male sex. Where possible, we add 'woman', 'women', 'her' and so on but if this would result in making the quote unreadable, we have left it as it stands. We hope this makes our position clear.“⁴⁸

⁴⁷Das Aushängen von Dateisystemen mit `umount` ist mit dem „sicheren Entfernen“ von Datenträgern unter Windows vergleichbar.

⁴⁸<http://www.infoshop.org/page/AnarchistFAQIntro>, Abruf: 03.02.2011

4. Eine Sprache sprechen

Begriff	Technische Bedeutung
guru	der Loginname der Person, die diese Shell bedient
linux	hier: der Name des Rechners, auf dem diese Shell läuft
who	Programm, das anzeigt, wer gerade eingeloggt ist
grep (-i)	Programm, das nach einem Suchmuster (hier: „blonde“) sucht, (ohne auf Groß- oder Kleinschreibung zu achten).
cd ~	Wechsel ins Homeverzeichnis (in den Ordner mit persönlichen Dateien)
wine	Programm, mit dem manche Windows-Anwendungen unter Linux gestartet werden können
touch	Programm, das den Zeitstempel einer Datei ändert
unzip	Programm zum Extrahieren von zip-Archiven
strip	Programm, das bestimmte Symbole aus Dateien streicht
finger	Programm, das Informationen über Nutzer_innen-Accounts auf einem Rechner ausgibt
mount	Programm zum Einbinden von → <i>Dateisystemen</i> (z. B. Festplatten, USB-Sticks)
fsck	Programm zur Überprüfung und Reparatur von Dateisystemen
more	Programm, mit dem Dateien seitenweise angezeigt werden können
yes	Programm, das eine Zeichenkette wiederholt auf dem Bildschirm ausgibt
gasp	Programm, das Assembler-Code, eine für Menschen lesbarere Form von Maschinensprache, in binäre Maschinensprache übersetzt
umount	Programm zum Aushängen („sicheren Entfernen“) von Dateisystemen
make clean	entfernt nach dem Kompilieren eines Programms die unnötigen Dateien der vorhergegangenen Arbeitsschritte
sleep	Programm, das die Ausführung von Prozessen verzögert

Tabelle 4.1.: Erläuterungen zum T-Shirt „Linux is sexy!“

Des Weiteren enthalten die FAQ auch einen Abschnitt zu Rassismus, Sexismus und Homophobie als gesellschaftlichen Herrschaftsverhältnissen, die dort als breit institutionalisierte Dominanzverhältnisse bezeichnet werden. „The primary cause of these three evil attitudes is the need for ideologies that justify domination and exploitation, which are inherent in hierarchy – in other words, ‘theories’ that ‘justify’ and ‘explain’ oppression and injustice.“⁴⁹

Aus meiner oben umrissenen Position heraus sehe ich in den Witzen vor allem das Handeln mit einem bestimmten Frauenbild. Frauen haben „Knöpfe“, die Männer betätigen können, Frauen werden wie Festplatten „gemountet“ und „gecheckt“, sie sind passiv und sie werden mit den Computern verglichen, über die der „guru@linux“ befiehlt. Queere oder schwule Witze gibt es in der LUG nicht, der sexualisierte Humor bewegt sich ausschließlich in der heterosexuellen Norm. (Auch wenn das Suchschema *grep -i blonde* dies nicht eindeutig besagt, so assoziiere ich implizit eine blonde Frau damit, anknüpfend an das Klischee von einfältigen „Blondinen“ mit Sexappeal.) Als Beobachterin und Teilnehmerin konnte ich zwar nach und nach mehr von der technischen Sprache verstehen, sie selbst sprechen oder mit ihr spielen. Bezugnahmen auf solche Stereotype wirkten auf mich jedoch stets als „Zurückfallen“ hinter alle bisherige Kommunikation. Witze, die nicht auf mich persönlich bezogen waren, sondern für manche zum Repertoire selbstverständlicher Sprüche zu gehören scheinen, *machten* mich immer wieder zur „Frau“. Die Hürde der Fachsprache sinkt mit der Zeit — die Festschreibung innerhalb einer zweigeschlechtlichen, heteronormativen Matrix bleibt (und mit ihr auch die Frage, ob ich mit bestimmten Gegenübern dann überhaupt über technische Fragen sprechen möchte, wenn ich offenbar immer in erste Linie „Frau“ bleibe). Als ich diese Wahrnehmung thematisierte, bekam ich von einem LUG-Mitglied die Antwort: „Freu Dich doch, dass Du so viel Aufmerksamkeit bekommst.“ Eine Äußerung, die genau mein Problem mit der Zuschreibung reproduziert, indem sie mich nochmal als „Frau“ anruft. Diejenigen, die sexistische oder andere diskriminierende Äußerungen machen, heben sich darüber auch vom Rest der Gruppe ab: Es sind Einzelne, die sich derart äußern, es sind deutlich mehr, die über sexistische Witze lachen, hingegen habe ich in der LUG kein Einschreiten dagegen beobachtet. Wenige machen solche Witze, aber viele lassen solche Witze im Raum stehen; sie sagen nichts dagegen.

⁴⁹<http://www.infoshop.org/page/AnarchistFAQSectionB1#secb14>

4.5. Empirische Zwischenergebnisse

Die spezifischen sprachlichen Eigenheiten von FLOSS-Communities sind konstitutive Elemente der Praxis der Gruppen. Technische Diskussionen finden in der Regel auf hohem Niveau statt. Wenn es auch möglich ist, sich zu erkundigen, worum es geht, so sind das Faszinierende oft genau die Details und die Gestaltungsmöglichkeiten, die die Software erlaubt. Der Humor ist stark auf Insider zugeschnitten, die die technischen Bezüge verstehen, z. B. das Umfunktionieren etablierter Formen wie RFCs, Bugreports oder Programmiersprachen für Witze und Anspielungen. Community-Mitglieder haben meist die sozialen und kulturellen Voraussetzungen bereits erworben, die nötig sind, um die Wortspiele als kreative Anstrengungen wahrzunehmen und am „I-get-it“-Effekt von Witzen teilzuhaben. Die Wirkung von Fachsprache und technisierter Umgangssprache ist, dass sich darüber technische Expert_innen konstituieren. In ihrem Sprechen entstehen wiederum andere, die sich nicht zutrauen, über Linux zu sprechen, mitzureden. Wie im folgenden Kapitel gezeigt werden wird, ist es ein Bestandteil des Feldes, dass sehr viele Linux-Nutzer_innen sich selbst über Jahre als technisch inkompetente „Newbies“ bezeichnen. Communityintern werden so über Sprache mehr und weniger Wissende hervorgebracht.⁵⁰ Viele Beobachtungen aus dem Bereich Sprache zeigen, wie über Sprache Geschlechtergrenzen und damit das heteronormative Zweigeschlechtersystem ständig (wieder-)hergestellt werden. Vergeschlechtlichte Stereotype, die Frauen als passive Sexualobjekte oder als desinteressierte Techniknutzer_innen konstruieren, sind fester Bestandteil der sprachlichen Bezüge und des Humors.

In Bezug auf die Kategorien, entlang derer Differenzen geschaffen werden, lässt sich festhalten: Die Sprache und das technische Wissen der Communities sind erlernbar. In Sprache und Humor kommt jedoch die Relevanz anderer Kategorien und Zuschreibungen zum Ausdruck. Wer die Sprache spricht, kann aufgrund von „falscher“ Hard- oder Software oder aufgrund von Geschlecht von subtiler Differenzierung, stilistischer Distinktion oder — im härteren Fall — von offener Diskriminierung betroffen sein. Kategorien wie Hardware / Software und Geschlecht können, wie am Beispiel der Referentin gesehen, zusammenwirken, gleichzeitig sozial relevant werden. Es ist festzuhalten, dass es sich bei diesen Kategorien nicht ausschließlich um breiter erforschte Diskriminierungskategori-

⁵⁰Ob überhaupt „objektive“ Maßstäben für Expert_innenwissen in FLOSS-Communities existieren, wird in Kapitel 5 untersucht. Diese Maßstäbe sind kontextabhängig, hängen vom inkorporierten Wissen der jeweils Anwesenden ab und sind eng mit der Spezialisierung auf bestimmte Themen verbunden. Wer sich viel Wissen über *ein* Thema angeeignet hat, kann auf einem anderen Gebiet Neuling sein.

en handelt. Vielmehr scheinen sich gesamtgesellschaftliche Differenzkategorien (wie Geschlecht) feldspezifisch mit anderen Kategorien (wie Software) zu verbinden. Dieses Zwischenergebnis wird im Verlauf der Arbeit aufgegriffen und überprüft.

4.6. Theoretische Reflexion: Fachsprache und Humor als Voraussetzung für Teilhabe

In diesem Kapitel wurden mehrere Dimensionen von Sprache in FLOSS-Communities beleuchtet: Erstens wurde aufgezeigt, wie die Beteiligten Expertise über Fachsprache und Humor konstituieren. Die Interaktion mit Noviz_innen im Feld wird dadurch teilweise stark gestört. Die Verwendung von Fachsprache ist ein grundsätzliches, kein linuxbezogenes Problem (vgl. etwa zur Kommunikation zwischen Ärzt_innen und Patient_innen Oksaar 1988: 171 ff.; siehe zu Jugendsprachen die eingangs angeführte Literatur). Die Sprachwissenschaftlerin Els Oksaar nennt die Benutzung von Fachausdrücken ein verbreitetes Stilmittel, das mit dem Wissen eingesetzt werde, dass der Gegenüber sie nicht verstehe (vgl. ebd.). Diese Fixierung auf die Fachsprache zieht sich auch durch den Humor von FLOSS-Communities wie Abschnitt 4.1 gezeigt hat. Laut Thomas Störel erfüllten Fachsprache und Witze je nach Szene oder Fachkultur sehr kontextspezifische Zwecke (vgl. Störel 2004: 423 f.). Wilfried Schütte hat dies für Ironie unter Orchestermusiker_innen sehr detailgenau gezeigt — das Orchester stellt einen Raum mit begrenzten expliziten Äußerungsmöglichkeiten dar, in dem Konflikte nicht direkt ausgetragen werden können, so dass in dem Kontext der Konfliktvermeidung und Kooperationssicherung eine wichtige Rolle zukommt (vgl. Schütte 1991: 333 ff.). Zudem werden innerhalb des Orchesters über Scherze Ingroups konstituiert (vgl. ebd. 340). Für Linux User Groups habe ich oben festgehalten, dass durch humorvolle Äußerungen einerseits die Komplizenschaft des „I-get-it“-Effekts nach (Friedman/Friedman 2003: 13) wirkt. Andererseits kommt es dadurch oft zu einer gleichzeitigen Demonstration von Überlegenheit gegenüber denen, die nicht an dem gemeinsamen Verstehen teilhaben.

„Es entsteht aber ein interessanter Nebeneffekt. Der Grad an Fachlichkeit wird hörbar; Der Fachmann lacht, der Laie schweigt.(...) Lachen als Indikator für Fachkompetenz! (...) Dabei hat das Lachen zugleich eine vereinende und eine abgrenzende Funktion“ (Störel 2004: 430).

4. Eine Sprache sprechen

Fachlicher Humor ist damit *auch* eine Inszenierung von Fachkompetenz, die Verständnisprobleme Dritter in Kauf nimmt, ein hermetischer, wissensbasierter Humor (vgl. Friedman/Friedman 2003: 13).

Zweitens wurden politische Diskussionen als Auslassung in den Gesprächen von LUGs identifiziert. Politik wird dort lediglich in humorvollen, verkürzenden Äußerungen kommentiert, die sich oft um die Vorstellung drehen, politische Probleme technisch lösen zu können.

Drittens ging es um die Reproduktion von Geschlechtergrenzen in der Community-Sprache, die auf verschiedenen Ebenen geschieht. Erstens werden weiblich sozialisierte Personen in Gruppen, in denen männlich sozialisierte Personen in der Mehrzahl sind, ständig als „Frauen“ angerufen. „Als geschlechtstypisch geltende Eigenschaften und Verhaltensweisen prallen aufeinander, plötzlich spielt nicht mehr das Sachthema, dem sich die Gruppe widmet, die Hauptrolle, sondern das Geschlecht steht im Rampenlicht“ (Jung 2006: 238). Zweitens wird mit Geschlechterstereotypen hantiert, die Frauen Technikkompetenz absprechen (vgl. Bath et al. 2010). Drittens werden technische Artefakte „scherzhaft“ nach weiblichen Körperteilen bzw. Geschlechtsorganen benannt. So werden Frauen implizit mit passiven Aspekten technischer Geräte verglichen. Gleichzeitig werden damit Sexualität, soziale Beziehungen und Politik in der Sprache von FLOSS-Communities einer technischen Logik unterworfen.

Die Sprache von FLOSS-Communities habe ich als Praxis untersucht. Dabei ging es mir um die gewollten oder ungewollten sozialen Effekte, die durch den Sprachgebrauch entstehen. Sie ist ein Teil des Habitus, den sich Linux-Nutzer_innen in ihren Lernprozessen aneignen, und ist damit Teil der Praktiken, die Differenzen schaffen. Dies ist, wie gesehen, nicht nur in FLOSS-Communities so, dort werden die gezeigten Wirkungen jedoch mit linux- oder computerspezifischer Fachsprache erzeugt. Analog zu anderen Äußerungsformen von Habitus wirkt auch die „Macht der Worte (...) nur auf diejenigen, die disponiert sind, sie zu verstehen und auf sie zu hören“ (Bourdieu 1992: 83). Diejenigen, die die Sprache (noch) nicht (so gut) beherrschen, bleiben sehr oft außen vor.

5. Informelles Lernen in Linux User Groups

Linux User Groups haben das Ziel, Linux zu verbreiten und das zu seiner Anwendung nötige technische Wissen zu vermitteln. In welchen Kontexten dies geschieht, in welche Praktiken dieser Wissenserwerb eingebettet ist, ist Thema dieses Kapitels. Vorab ist es wichtig, die Kontexte in den Blick zu nehmen, in denen Linux-Nutzer_innen lernen, mit dem Betriebssystem umzugehen. Freie Software ist in der Regel kein Gegenstand formeller Lernkontexte: In Schule, Ausbildung oder Studium spielt Linux eine marginale Rolle. Die Entscheidung, unter Linux zu arbeiten, erfolgt sehr oft individuell, so dass die Nutzer_innen sich auch den Wissenserwerb selbst organisieren müssen. Die autodidaktische Aneignung von Wissen ist im Zusammenhang mit Linux grundlegend.¹ Hier kommen LUGs ins Spiel, die eine zentrale Rolle dabei spielen, Neulinge zu unterstützen. Über das erste Erlernen hinaus, ist Software auch ein Bereich, in dem Wissen aufgrund der technologischen Entwicklung schnell veraltet; den Communities muss demnach auch für die ständige Aktualisierung von Wissen eine große Bedeutung beigemessen werden (vgl. Glott et al. 2007: 2). Viele Linux-Nutzer_innen sind über Freund_innen oder Bekannte dazu gekommen umzusteigen. Von ihnen haben sie Unterstützung bekommen oder sie haben sich mit ihnen gemeinsam das Betriebssystem angeeignet (vgl. z. B. Int 2007a). Christina Haralanova hat in ihrer nicht repräsentativen Arbeit über Frauen in Open-Source-Projekten erfahren, dass es nur einer Minderheit der Befragten so ergangen ist. Der Einstieg in freie Software ist bei den meisten anderen im beruflichen Umfeld erfolgt, wo Kolleg_innen diese unterstützende Rolle einnahmen (vgl. Haralanova 2010: 84). Die von Haralanova Befragten erinnern sich an diesen Prozess, an die Integration von freier Software in die alltägliche Arbeitspraxis als einen langwierigen Prozess, der von oft nur graduellen Fortschritten und viel Autodidaktik geprägt war (vgl. ebd., siehe auch Int 2007b). Die Beobachtungen, die in diesem Kapitel analysiert werden, stehen damit vor dem Hintergrund, dass Linux-Nutzer_innen weder einen formalisierten Lernprozess durchmachen, noch einen gemeinsamen Kanon

¹Anders als in Bourdieus Analysen stehen in FLOSS-Communities die Autodidakt_innen nicht unter Beweiszwang gegenüber denen, deren Wissen in irgendeiner Form zertifiziert wurde (vgl. Bourdieu 1983: 189 f.).

an Wissen anhäufen. Zudem ist die Aneignung von technischem Wissen für alle ein nie endender Prozess: Jede_r startet mit etwas anderem (z. B. mit den Standard-Programmen der für den Einstieg gewählten → *Distribution*) und verfolgt nach Bedarf, Interesse, Ratschlägen oder Zufall bestimmte Themen; der Weg ist damit meistens individuell und keineswegs linear (vgl. z. B. Haralanova 2010: 83 ff.).

Linux User Groups sind Institutionen, die sich die Unterstützung dieses Lernprozesses zum Ziel gesetzt haben. Die wenigen Vorarbeiten, die speziell zu LUGs vorliegen, wurden in Abschnitt 2.1.6 bereits vorgestellt (Bagozzi/Dholakia 2006; Lin 2004a; Huysman/-Lin 2005). Ihnen geht es aber primär um die Potenziale des gemeinsamen Lernens und um LUGs als soziale Zusammenhänge. Sie nehmen keine kritischen Perspektiven ein und es geht ihnen nicht im Detail um die Bedeutung der Praktiken.

In LUGs gibt es verschiedene Formen von Wissenstransfer, die von der Größe der Gruppen, vom Organisationsgrad und den räumlichen Möglichkeiten abhängen. LUGs, die als Stammtische in Kneipen stattfinden, haben z. B. weniger Möglichkeiten, Vorträge mit Beamer anzubieten, sehr kleine Gruppen haben nicht genug Zulauf von interessiertem Publikum oder zu wenig personelle Ressourcen für die Organisation eines Programmes. In der LUG, in der ich hauptsächlich Daten erhoben habe, finden gelegentlich Vorträge statt, regelmäßig aber die Treffen, bei denen technischer Support angeboten wird. Diese Abende nutzen die Anwesenden auch, um sich selbst in Stillarbeit Neues beizubringen² oder gemeinsam mit anderen Software auszuprobieren. An gesonderten Terminen trafen sich auch ein Linux-Einsteiger_innenkurs und eine Lerngruppe für Prüfungskandidat_innen, die sich gemeinsam auf die Prüfung des → *Linux Professional Institute (LPI)* vorbereiteten. In der LUG werden sehr viele verschiedene technische Probleme gemeinsam behandelt. Inhaltlich ging es dabei oft um Netzwerkthemen (WLAN-Konfiguration, die Einrichtung von Firewalls, das Verständnis verschiedener Ebenen des Netzwerkverkehrs und die Zuordnung der jeweiligen Protokolle), um den Installationsprozess an sich, um die Installation und Konfiguration bestimmter Programme. Auch die → *Partitionierung* von Festplatten (oft mit mehreren parallel installierten Betriebssystemen oder → *Distributionen*) war ein häufiges Thema.

²„Ich setze mich neben A14, der bisher extrem still ist. Ich frage mich, ob er schon etwas gesagt hat und was er eigentlich die ganze Zeit macht“ (LUG 2009l). „Jetzt sind noch fünf Personen (Männer) am Tisch. Minutenlang spricht niemand. Einer guckt in die Runde, die anderen sind jeder mit seinem Notebook beschäftigt. A50 scheint zu chatten, er hat einen textbasierten Client, vermutlich *irssi*, vor sich. Die Schrift ist winzig. Als eine Nachricht eingeht, macht der Rechner, ein Medion, ‚plönk‘. A47 blickt kurz rüber“ (LUG 2009i).

In diesem Kapitel gebe ich zunächst eine Beschreibung von einem Abendverlauf in einer LUG (5.1), um dann konkrete Sequenzen genauer zu untersuchen, in denen es um den Wissenstransfer in sich spontan konstituierenden Kleingruppen geht.

5.1. Annäherung an LUG-Treffen

Die User Groups, die ich beobachtet habe, treffen sich jeweils regelmäßig an festen Orten. Die größte und älteste der Gruppen hat Vereinsräume, die sie sich mit einem anderen im Internetbereich aktiven Verein teilt. An einem Abend in der Woche gehören die Räume nur der LUG. Es sind Ladenräume in einem Altbau, in einer Straße mit Wohnhäusern. Die beiden Schaufenster sind mit Gegenständen und Plakaten der beiden Vereine dekoriert. Dort liegen Linux-CDs, Stoffpinguine und Hardwareteile, z. B. eine auseinandergenommene Festplatte.

Ich komme schon gegen 18:15 Uhr an, sehe beim Anschließen meines Rades, dass noch tote Hose ist. Gehe noch einen Tee gegenüber im Café trinken. (...) Ca. 18:45 Uhr, ich gehe rüber und wundere mich, dass im hinteren Zimmer ein Haufen Leute sind. War es vorher wirklich leer? Vorne ist es immer noch eher schummerig (anders als sonst, wo auch vorne Licht ist und im Seitenzimmer die Jalousie hochgezogen wird). Im Seitenzimmer ist niemand, aber im Eingangszimmer steht einer, der sagt, es sei ja total voll da hinten. Er scheint zu zögern hineinzugehen. Ich drängele mich ins Getümmel (LUG 2007b).³

Es gibt vier Räume: Der Eingangsbereich ist der größte Raum, in dem auch Vorträge der LUG stattfinden. Unter der Decke hängt ein Beamer, aufgestapelte Stühle stehen an der einen Wand, ein Tisch ist immer wie ein Empfangstisch oder Rednerpult neben der Tür aufgestellt. Rechts davon liegt ein kleiner Raum mit einem ovalen Tisch, an dem etwa acht bis zehn Personen Platz haben. An den Wänden hängen Plakate von verschiedenen vergangenen Community-Events. Unter dem Tisch sind etliche Mehrfachsteckdosen und → *Switche* mit Kabelbindern an Querverstrebungen der Tischbeine befestigt. Einige

³In diesem Protokollauschnitt wird eine Hürde angedeutet: In dem von der Straße aus sichtbaren Raum brennt kein Licht, im Nebenzimmer ist die Jalousie heruntergelassen, so dass von außen nicht sichtbar ist, ob drin überhaupt ein Treffen stattfindet. Die erste Person, der ich begegne, hat offenbar eine Hemmung, sich in das volle Hinterzimmer zu begeben. Die Situation wirkt zunächst so, dass sich eine Ingroup im nicht einsehbaren Teil der Vereinsräume versammelt hat und es für Neulinge u.U. Überwindung kostet hineinzugehen. Gleichzeitig fällt mir diese Situation als ungewöhnlich auf, weil die „gefühlte Zugänglichkeit“ sonst durch die Beleuchtung und hochgezogenen Jalousien anders gestaltet ist.

Netzwerkkabel hängen in den Switchen. Vom Eingangsbereich kommt man durch einen kleinen Flur, in dem auch Getränkevorräte und ein paar Schränke stehen, in das hintere Zimmer, dessen einziges Fenster in einer verwinkelten Ecke auf den Innenhof zeigt. Der Raum ist 25–30 Quadratmeter groß, lang und eher schmal, an beiden Wänden stehen Tische und Stühle. Neben der Tür hängt ein ausgedienter Switch oben an der Wand, in dem Netzwerkkabel hängen. Man kann sie sich hier leihen. Die Dekoration besteht in diesem Raum aus vielen bunten Schachteln von verschiedenen Linux-*Distributionen*, in denen einmal CDs oder Bücher verpackt waren. Sie sind entlang der Wände auf dem Sockel aufgereiht, in dem Strom und Netzwerkkabel verlegt sind.

Beim Hereintreten in den hinteren Raum ist sofort sichtbar, dass der Raum sich verändert hat. Er wirkt entrümpelt, leerer, aufgeräumter und da stehen neue Flachbildschirme auf den Tischen. Ich höre von irgendwem, vielleicht A4, dass die LUG eine Spende erhalten habe und sie davon angeschafft habe (LUG 2008c).

In einem abschließbaren Metallschrank wird Hardware aufbewahrt.⁴ Hinter dem Raum schließt sich noch ein kleiner Raum an, in dem eine Sofaecke, ein Couchtisch und ein Regal mit Linux-Büchern und -Zeitschriften stehen. Hier finden manchmal während der Abende vereinsinterne Treffen statt, manchmal wird der Raum auch als *Raucher_innenzimmer* benutzt, in der Regel gehen Raucher_innen jedoch auf die Straße hinaus.

Begrüßung und Abschied erfolgen unter *bestimmten* Anwesenden, die sich nämlich kennen, mit Handschlag.⁵ Mir fiel nach über einem halben Jahr in der LUG auf, dass ich plötzlich mit einem Handschlag verabschiedet wurde.

Zwischendurch kommt A43, der A1 und A4 mit Handschlag begrüßt, mich nicht, obwohl ich quasi zwischen den beiden sitze (LUG 2008g).

Irgendwann kommt einer rein und begrüßt die ca. vier Männer auf der anderen Seite des Tisches (A3, A4, ...) mit Handschlag. A52: „Klopp doch auf den Tisch, dann musst Du nicht rumgehen.“ Mir fällt auf, dass sie nicht unter denen ist, die so begrüßt werden (ich sowieso nicht, ich kenne den Hereinkommenden nicht) (LUG 2008k).

⁴Bei Gesprächen über die LPI-Prüfungen erfuhr ich, dass die LUG für Lernzwecke hier auch alte Hardware vorhält, z. B. Isa-Netzwerkkarten, die längst nicht mehr verwendet werden, aber beim LPI noch Prüfungsthema sein können.

⁵Die Begrüßung per Handschlag ist mir zunächst sehr aufgefallen, weil sie mir in einem Freizeitrahmen ungewöhnlich „formal“ erschien. Zu unterschiedlichen (sub-)kulturellen (Be-)Deutungen eines Begrüßungsritual vgl. jedoch Wagner (2006).

Was mir sehr aufgefallen ist, ist dass mich erstmalig mehrere Gruppe1-Leute mit Handschlag verabschiedet haben (LUG 2008l).

A52 ist eine Frau, die mindestens 65 Jahre alt ist und als langjährige Linuxnutzerin unregelmäßig zu den Treffen geht. Für ihre Bemerkung gibt es verschiedene mögliche Interpretationen: Möglicherweise findet sie es umständlich, dass der Neuankömmling sich in dem engen Raum hinter den Stühlen entlangdrängelt, um bestimmten Personen die Hand zu geben. Eine andere Lesart ist, dass sie registriert, dass der Hereinkommende ihr *nicht* die Hand gibt. Ihre Äußerung kann somit auch als Vorschlag verstanden werden, zum Gruß aller Anwesenden auf den Tisch zu klopfen. Sie würde ihm damit eine deutlich männlich konnotierte Geste vorschlagen, die unter den Anwesenden jedoch nicht diesen Unterschied zwischen Begrüßten und nicht Begrüßten produzieren würde.

Beim Ankommen schaute ich immer zuerst, wer alles da ist und wo ich mich dazusetzen könnte. Die Abende richten sich an alle, die Fragen zu Linux haben. Die Fragen oder Probleme werden dann in sich spontan organisierenden Kleingruppen bearbeitet. Diese Kleingruppenarbeit endet gegen 22:00 Uhr. Oft begibt sich ein Teil der Anwesenden im Anschluss noch in ein Restaurant in derselben Straße. Vorher wird herumgefragt, wer noch mitgehen möchte.

5.2. Durch technischen Support „der Community etwas zurückgeben“

5.2.1. Hilfesuche als Herausforderung im informellen Kontext

Eine Schlüsselsituation in der LUG in Bezug auf Wissenstransfer ist die Suche nach Hilfe. Jemand hat eine Frage oder ein Problem und versucht, eine Person zu finden, die helfen kann und will. Diese Suche verläuft informell, in dem Sinne, dass es am Eingang kein „Helpdesk“ gibt, wie z. B. im Rechenzentrum der Universität, wo die Probleme besprochen werden und die Fragenden dann an eine_n Helfer_in weiterverwiesen werden. Hier muss sich jede_r selbst jemanden suchen.⁶

Bei der Suche nach Hilfe oder Reaktionen auf Hilfesuche gibt es unterschiedliche Praktiken: Auf manche unbekanntes Gesichter gehen LUG-Mitglieder zu und fragen, ob sie

⁶Dieser Vergleich von formellen und informellen Kontexten des Computersupports entstammt einer Diskussion im Colloquium.

Hilfe suchten. Mir selbst ging es anders: Bei meinen ersten Besuchen wurde ich nicht beachtet. Später wurde ich dadurch, dass ich mich in die Kleingruppen hineinbegab, in manchen Situationen selbst als potenzielle Helferin in technische Fragen involviert, vor allem, wenn ich nicht offensichtlich gerade Feldnotizen machte.⁷ An anderen Abenden hatte ich wiederum selbst meinen Rechner mitgebracht und hatte technische Fragen.

Das Helfen in FLOSS-Communities ist bereits in einigen wenigen wissenschaftlichen Arbeiten untersucht worden, die jedoch hauptsächlich die Softwareentwickler_innen oder die anderweitig *direkt* an Softwareprojekten Beteiligten in den Blick nehmen. Karim Lakhani und Eric von Hippel haben die Motivationen derer untersucht, die kostenlosen Online-Support für den *Apache* → *Webserver* gewährleisten. Sie kommen zu dem Ergebnis, dass die Helfer_innen den größten Teil der Zeit im entsprechenden Forum damit verbringen, die Fragen und Antworten anderer zu *lesen*. Sie antworteten in der Regel dann, wenn sie die Antwort auf eine Frage schon wussten, recherchierten also eher selten zusätzlich und brauchten deshalb für den Support nur wenige Minuten.

„Information providers report that their motive for reading questions is primarily to learn about problems that other Apache users are experiencing. This learning helps them to manage and update their own Apache websites and software code. In other words, the major cost in providing help (...) is carried out by providers because they directly receive a reward for this activity“ (Lakhani/von Hippel 2003: 924).

Die Studie von Lakhani/Hippel folgt einem rational choice-Ansatz, der die Kosten-Nutzen-Erwägungen der Helfer_innen diskutiert. Ein „Nutzen“, der auf ihrer Seite oft erwähnt wird, ist die Erwartung von Gegenseitigkeit: Wer hilft, erwartet, in einer anderen Situation selbst Hilfe zu bekommen, und zwar nicht von derselben Person, der geholfen wird, sondern als verallgemeinerter Austausch innerhalb einer Community (vgl. ebd. 927). Auch die Hilfe aus Überzeugung vom Projekt oder von freier Software allgemein ist ein häufig genannter Grund, ebenso wie Spaß oder Pause von einer anderen Aktivität. Dagegen wird das Erarbeiten eines guten Rufes für eine berufliche Karriere sehr selten genannt.⁸

⁷„Ich gehe weiter und will A5, A6 und A4 ein bisschen zuschauen. A76 fängt mich ab und fragt, ob ich mich mit WLAN auskenne. Ich frage, was das Problem sei (LUG 2008b).

⁸Die Autoren weisen an dieser Stelle darauf hin, dass die Angaben mit Vorsicht zu genießen sind, weil die Befragten möglicherweise die „sozial korrekten“ Motivationen besonders hervorgehoben hätten (vgl. ebd. 938).

Meine Beobachtungen in der LUG widersprechen dem zitierten Ergebnis teilweise. Das voneinander Lernen mag auch hier im Vordergrund stehen, und ebenso das Verständnis von Wissensaustausch in einer Community, der man „etwas zurückgeben“ möchte, im Austausch für all das, was man selbst an Software oder Hilfe aus der Community bekommen hat. In der LUG wird jedoch nicht nur dann geholfen, wenn die Lösung schon bereit liegt, sondern sie wird oft *über Stunden* gemeinsam erarbeitet. Dieser Unterschied lässt sich m.E. genau dadurch erklären, dass die LUG kein Online-Support ist, wo Fragen eingeschickt werden und Leute sie erst lesen, dann aber beiseite legen können. Hilfesuchende und potenzielle Helfer_innen stehen sich hier von Angesicht zu Angesicht gegenüber; die LUG ist ein expliziter Ort für das gemeinsame Bearbeiten technischer Probleme. Dennoch kommt es nicht bei jeder Anfrage auch zu einem Austausch oder Wissenstransfer:

Ich gucke nur kurz rum und gehe dann zu A1, der gerade fertig wird, jemandem etwas zu erklären. Ein mir unbekannter junger Mann (A44) spricht ihn darauf an, ob der bei → *wakeonlan* helfen könnte. (...) Während wir warten, dass A1 frei wird, fragt er mich, ob ich Ahnung von *wakeonlan* hätte. Ich sage, ich hätte mich ein bisschen damit beschäftigt (...). A1 überprüft mit dem Programm *ethtool*, ob die Netzwerkkarte bereit ist, es sieht so aus, trotzdem passiert nichts. Nach einer Raucherpause auch noch mit A45 (der mal ein *Powerbook* hatte, jetzt einen nagelneuen *Thinkpad* mit ein paar Problemen hat), beschäftigt sich A1 mit A45. Ich finde das nicht so richtig nett, denn das Problem des jungen Typen war ja ungelöst, und wir hatten auch nicht lange versucht. Ich probiere, mich von A44s Laptop mit seinem → *Server* zu verbinden (...). Wir konfigurieren beide Netzwerkschnittstellen passend und versuchen uns anzu→ *pingen*. Es geht nicht. Ich überprüfe die Kabel (...). An einer Stelle sagt er: „Ein Server ohne *wakeonlan* ist wie ein Mann ohne...“ Ich glaube, er sieht, dass ich damit nichts anfangen kann und fügt schnell hinzu: „wie eine Frau ohne... , wie ein Auto ohne...“ (...) Zwischendurch fragen wir A1 einmal, ob er noch etwas anderes getan hat als seinem Rechner die → *IP-Adresse* zu geben. Er kommt kurz gucken, wundert sich, setzt sich aber wieder zu A45. (...) A1 und A45 wollen eine rauchen gehen, da fragen wir ihn, warum wir uns nicht einloggen können. (...) In der Raucherpause erläutert A1 uns, wozu die Datei gut ist und wie er mit dem *vi* eine bestimmte Zeile löscht bzw. wie er sie überhaupt findet, wenn es sich um umbrechende Zeilen handelt. Da erzählt A44, er hätte es unter *Windows XP* hinbekommen mit dem *wakeonlan*. A1 wird hellhörig und hat Ideen mit *modinfo*. Er ist nun überzeugt davon, dass es gehen müsste und setzt sich nach der Pause wieder zu uns rüber (LUG 2008j).

Zur Interpretation: In dieser Sequenz versuchen A44 und ich (als hinzugezogene, aber

erfolglose Helferin), die Aufmerksamkeit eines Helfers, A1, auf uns und A44s Problem zu ziehen. Dabei entsteht eine implizite Konkurrenz um ihn, denn sein Bekannter A45 kommt mit seinem neuen Rechner. Es erfordert insgesamt vier Kontaktaufnahmen, bis A1 sich „ernsthaft“ A44s Problem widmet. Dabei markieren die Raucherpausen vor der Tür Momente, in denen A1 von der vorherigen Kleingruppe „abgeworben“ werden kann, wenn es gelingt, ihn für das eigene Problem zu interessieren. Ich verlasse die Beobachter_innenrolle, als ich unzufrieden damit bin, wie schnell A1 die Szene wieder verlässt, und fange selbst an, mich mit dem Problem zu beschäftigen. Es kommt darüber hinaus zu einem Moment der Irritation, als A44 der Situation eine vergeschlechtlichte Bedeutung gibt: Er führt den Satz zwar nicht zuende, doch ich interpretiere seine Äußerung so, dass er die *wakeonlan*-Funktionalität für seinen Server so wichtig findet wie seine Geschlechtsorgane für seine Männlichkeit. Ich reagiere nicht zustimmend, so dass er den Satz auf mich als Frau anpasst, dann stattdessen einen Vergleich mit einer anderen Maschine einbaut. Die Szene ist damit gleichzeitig ein Beispiel dafür, wie die Kategorie Geschlecht in Szenen eingeführt wird, in denen es eigentlich um eine technische Frage geht.

In einer anderen Szene sucht ebenfalls jemand die Hilfe von A1, der mit etwas anderem beschäftigt ist. Der Hilfesuchende versucht länger erfolglos, Hilfe zu bekommen. Auch diese Szene beginnt beim Rauchen draußen.

A53 hat eine Frage, die er (glaub ich) in die Runde stellt, A1 antwortet. Es geht darum, dass A53 unter irgendeinem Windowmanager das Programm *nm-applet* wohl nur als root starten kann, das aber gerne immer ohne Passwort automatisch starten würde. A1 erläutert, wie er das lösen würde. (...) A1 nennt A53 einen Weg, das über Gruppenzugehörigkeiten⁹ zu lösen, und dann gehen alle wieder rein. Ich folge A53 und frage, ob ich ihn beobachten dürfe. Ich setze mich links neben ihn. (...) Zwischen dem Laptop und sich hat A53 ein aufgeschlagenes Buch liegen, in dem Befehle erklärt sind. Er gibt versuchsweise ‚group‘ ein, doch der Befehl existiert nicht. Er sieht sich suchend um, scheint A1 zu suchen, der nicht zu sehen ist. (...) A53 bewegt den Cursor mit den Pfeiltasten durch die Datei. Er sieht wohl die gesuchte Gruppe nicht, denn er fragt mich danach. Ich kann mich auch nicht erinnern, was A1 gesagt hatte und schlage vor, nach einer Gruppe mit ‚net‘ zu suchen. Er scrollt mit der angeschlossenen PS/2-

⁹Unter Linux gibt es ein Rechtekonzept, das regelt, wer was tun darf, welche Hardware nutzen darf etc. Dazu gibt es Gruppen, denen ein_e User_in angehören muss, um z. B. Zugriff auf Netzwerkressourcen oder das CDROM-Laufwerk zu haben.

Maus¹⁰ durch die Datei. Er murmelt: „Oh, das ist ’n bisschen kompliziert hier.“ Er beginnt, mit einer Hand eine Zeile am Ende der Datei hinzuzufügen. Die andere Hand hat er auf dem Buch: Er tippt: `nm-applet:x`, hält inne und blättert im Buch. (...) Er wendet sich wieder seinem Rechner zu und sagt: „Ich weiß nicht, ob ich da noch Bock hab.“ Er fragt einen anderen Mann, der schräg hinter ihm steht und schildert das Problem nochmal: Er möchte das *nm-applet* ohne root-Rechte starten können. Der Mann gibt ihm den Tipp im Manual des Programms nachzulesen, dort müsse das dokumentiert sein. (Beim Protokollieren denke ich: `rtfm`.¹¹) (...) Er spricht A6 an, der rechts neben ihm sitzt und fragt den. Der, der hinter ihm steht, blickt auf die begonnene Zeile `nm-applet:x` und sagt, A53 solle für diese Gruppe eine neue Nummer vergeben, die nächste freie, also die 111. A53 schreibt `nm-applet:x:111:A53`. A1 taucht auf und fragt A53 nach einer ‚Mini-Linux‘-Distribution, von der sie offenbar vorher gesprochen hatten. A53 gibt ihm Auskunft, wie die heißt. (...) Bevor A1 wieder geht, hakt A53 nochmal nach, hier sei die Datei, was er jetzt tun müsse. A1 rät ihm, bei Google zu suchen. A53 sagt, dazu sei sein Rechner zu langsam. A1 bietet an, sie könnten ja schnell bei ihm gucken, holt seinen Thinkpad von einem der Nachbartische und stellt sich hinter A53 und mich. Ich rücke zur Seite, so dass er den Rechner zwischen uns auf dem Tisch abstellen kann. A1 bedankt sich und gibt ein *grep A1 /etc/group*. Er schlägt vor, es könne vlt um die Gruppe `plugdev` gehen. (...) In dem Moment kommen ein Mann und eine Frau in den hinteren Raum, jemand der bereits Anwesenden (ich weiß nicht mehr, wer) hat die beiden nach hinten begleitet. Der Mann sagt laut: „Wir suchen einen Linux-Experten!“ (...) Er sagt zu A53: „Gib mir ne Shell, ich mach’s schnell.“ Da bekommt er schon mit, dass die beiden anderen drüben schon sitzen. Er bricht ab und diktiert A53: „`visudo`“. Dann sagt er sehr schnell, was A53 machen soll. A53 fährt sich mit den Händen durch die Haare und atmet laut aus. A1 sagt, das sei zu schnell gewesen, geht dann aber aus dem Raum. (...) Er stöhnt und ändert die Datei wieder in ihren Originalzustand. Dann spielt er mit den Seiten des Buches herum, nimmt ein dickes Bündel Seiten und wippt sie von einer Hand in die andere und guckt rum. Er beteiligt sich kurz an einer Diskussion, die rechts hinter ihm zwischen zwei Stehenden geführt wird, über den Unterschied zwischen gebrannten und gepressten CDs. Dann sagt er, er komme nicht weiter und es reiche für heute (LUG 2009g).

A53 bekommt beim Rauchen einen Hinweis von A1, der sich jedoch danach nicht zu ihm setzt. Er kommt allein bei der Umsetzung nicht zurecht, fragt nacheinander mich, einen

¹⁰A53 hat einen auffällig alten Laptop dabei, der noch über einen PS/2-Anschluss verfügt und kaum Rechenkapazitäten für Internetrecherchen hat. „An einer Stelle sagt er auch, dass er Spaß dran habe, diesen alten Laptop am Laufen zu halten“ (LUG 2009g).

¹¹ „Read the fucking manual!“

Umstehenden, seinen anderen Sitznachbarn und wieder A1 um Hilfe. Zweimal wird er darauf verwiesen, dass er selbst recherchieren solle. Als der Helfer A1 sich dem Problem annimmt, treffen neue Personen auf Hilfesuche ein (der Ausschnitt wird im Folgenden genauer beleuchtet). A1 bietet A53 dann zunächst kurz an, das Problem schnell selbst zu lösen, diktiert ihm dann jedoch viel zu schnell, wie es geht, merkt, dass er A53 überfordert und geht. A53 kann sein Problem nicht lösen, er äußert verbal und gestenhaft seine Frustration darüber. Er fordert die Hilfe aber nicht explizit ein, in dem Sinne, dass er A1 auffordern würde zu bleiben.¹² In einer anderen Szene brauchte ich selbst Hilfe und setzte mich neben jemanden, der gerade einer dritten Person half. „Es war dann mehr so, dass ich zwischendurch ein bisschen seiner Aufmerksamkeit abbekommen konnte, wenn der andere rauchen war“ (LUG 2008i). In der Situation habe ich die Vermutung notiert, dass meine Frage für den Helfer nicht so ein interessantes Thema berührte. Andererseits kann es genau so gut daran gelegen haben, dass ich zurückhaltend aufgetreten bin, um die beiden nicht stören. Die Informalität des Hilfeverhältnisses bewirkt, dass jeweils neu ausgelotet werden muss, wie sehr ein Helfer beansprucht werden kann. Um manche Personen scharen sich viele auf der Suche nach Hilfe, ihre Zeit und Aufmerksamkeit sind gefragt. Sie erbringen die Hilfeleistungen jedoch auf einer informellen und freiwilligen Basis, was es sehr schwer macht, eine bestimmte Qualität oder Kontinuität zu verlangen. Verschiedene unberechenbare Aspekte tragen dazu bei, ob jemand Hilfe bekommt: Ist das Problem auch für die helfende Person interessant? Möchte die Person ein für sie einfach zu lösendes Problem behandeln oder selbst etwas Neues lernen? Geht es also um einen Wissenstransfer in *eine* Richtung oder um das gemeinsame Aneignen eines Themas? Es stellt auch einen Unterschied dar, ob es zwischen den Involvierten bereits persönliche Beziehungen gibt, auf die aufgebaut werden kann.

Eine weitere Hürde bei der Hilfesuche ist das adäquate Formulieren des technischen Problems:

Ich spreche diesen A48 darauf an, dass mein Rechner irgendwie immer langsamer werde. Er bereite sich doch auf die LPI-Prüfung vor und hätte dazu doch vielleicht eine Idee. Er fragt, seit wann das so sei. Ich sage, keine Ahnung, vielleicht seit dem Upgrade auf *Debian-Testing*. Manche Programme starteten sehr langsam. Er sagt so was wie, ich müsse mal lernen, Probleme zu formulieren. Wie Leute, die anriefen, der Rechner gehe nicht (LUG 2008d).

¹²Ich habe *nie* beobachtet, dass jemand eine_n Helfer_in explizit auffordert zu bleiben, bis das Problem gelöst ist. Es gibt vielmehr einen impliziten Konsens, dass die Hilfe keine einforderbare Leistung ist.

„Es geht nicht“ wird in vielen Situationen nicht als angemessene Problembeschreibung akzeptiert. Schon bei der Problembeschreibung zeigen Linux-Nutzer_innen, wie viel sie bereits über das aufgetretene technische Problem wissen, ob sie selbst schon dazu recherchiert haben und wie weit sie dabei gekommen sind. Dave Yeats untersucht dieses Thema im Hinblick auf die Softwareentwicklung eines bestimmten Projektes. Er sieht sich an, welche an das Softwareentwicklungsteam gemeldeten Fehler wie aufgenommen werden.

„The most well-received kinds of bugs are those submitted by what Eric Raymond calls 'source-aware' users. These users not only understand that the open-source community encourages users to participate in the development process by reporting problems with the programs, but also they understand the programming itself (...). The most acceptable kind of bug report is one that includes a sufficient description of the problem. According to developers, a sufficient description may not include much information produced by a human (or, at least, it's less important). Instead, developers look for a (...) computer-generated report of what happened during the failure of the program“ (Yeats 2006: 100).

Beim technischen Support in der LUG wird einer Problembeschreibung nicht derselbe Grad an Perfektion abverlangt wie in der Kommunikation mit Entwickler_innen. Dennoch erwarten viele Helfende eine Bereitschaft zu selbständigem Lernen, dass der_die Hilfesuchende sich schon über das Problem informiert hat, so weit es möglich war, oder zumindest Interesse für Lösungswege aufbringt (vgl. Weller/Meiszner 2008: 10). Dabei definieren in der LUG die Helfenden, ob dies bei den Fragenden der Fall ist oder nicht. Es wird nicht gern gesehen, wenn Menschen dort mit der Einstellung hinkommen, sie könnten sich bei der LUG bedienen lassen wie bei einem kostenlosen oder einforderbaren Dienstleistungsunternehmen. Dasselbe gilt für Hilfesuchende, die komplette → *Server* für berufliche Zwecke aufsetzen möchten.¹³ Hier hat jemand nach möglichen Ursachen für sein Problem gesucht, um dem Helfer mit Vorannahmen gegenüberzutreten:

A50 sagt, er hätte jetzt herausgefunden, dass sein Laptop ohne → *X-Server* eine Netzwerkverbindung hätte, mit grafischer Oberfläche jedoch nicht. Er warte jetzt auf A39,

¹³Ein genannter Grund hierfür war die Verantwortung für einen solchen Server. „Eine kurze Diskussion kommt noch auf, als A2 irgendwann reinkommt und A3 anspricht. Es kämen immer öfter Leute, die Hilfe beim Aufsetzen eines kompletten Servers wollten. Allein heute seien schon 3 da gewesen. Er schicke die immer weg, weil er sich nicht dafür zuständig fühlt, aber vielleicht müsse das diskutiert werden, ob alle das so sehen. Er scheint v. a. keine Verantwortung für anderer Leute Server übernehmen zu wollen“ (LUG 2008a).

immerhin sei das Problem jetzt schon eingegrenzt (LUG 2009i).

Einige meiner Beobachtungen deuten jedoch darauf hin, dass nicht von allen gleichermaßen erwartet wird, dass sie sich bereits informiert haben. Diese unterschiedlichen Erwartungen und Vorannahmen strukturieren die Hilfesituationen und sind Gegenstand der folgenden Abschnitte. Wie gesagt, gehe ich davon aus, dass im technischen Support mehr als nur technisches Wissen vermittelt wird: Die Beteiligten handeln darin gleichzeitig soziale Beziehungen aus.

5.2.2. Vorwissen und Annahmen über Vorwissen

Wer erwartet von wem welches Vorwissen? Wer traut wem kein oder kaum Vorwissen zu? Wovon werden diese Erwartungen beeinflusst? Warum wurde ich beispielsweise auf einem Community-Event gefragt, ob auf meinem Notebook *Windows* installiert sei? Um einen Erklärungsansatz vorwegzunehmen: Die Annahmen über das Vorwissen anderer sind geschlechtlich konnotierte Vorannahmen. Ich konnte sehr wenig Frauen in der LUG beobachten. Dennoch ergibt sich in der Tendenz das Bild, dass genau von ihnen keine eigenen Recherchen erwartet werden. Dahingegen habe ich keine Protokollausschnitte, in denen Männern von vornherein Unwissenheit unterstellt wird.

Die Personen habe ich glaub ich noch nie gesehen: (...) Eine junge Frau, um deren Laptop es geht, A78. Sie hat einen Helfer (A79), den ich auch noch nicht kenne. Er rollte das ‚r‘ und spricht evtl. auch mit leichtem Akzent, hat kurze Haare und einen kurzen Vollbart, eine Brille. (...) Mit A78s Hardwareuhr stimmt etwas nicht, was beim grafischen Login zu Problemen zu führen scheint. A79 arbeitet auf einer → *Shell* und gibt ein *hwclock set*. (...) A78 sieht zu und sagt, irgendwann beschäftige sie sich auch mal mit „Codes“ und nehme sich dafür zwei bis drei Monate frei, in denen sie nichts anderes tue. „Ich bin froh, dass ich Euch hab.“ A79 gibt noch etwas mit *tzconfig* ein und tippt dann *reboot*. (...) Jetzt hat sie weitere Fragen. (...) Sie brauche noch *Firefox*. A79 arbeitet an ihrem Rechner, sie sieht zu. Er startet eine Paketverwaltung und wählt *Firefox* zum Installieren aus. (...) Ihre *Gnome*-Leiste oben ist auf englisch und sie fragt, ob man das ändern könne. A79 durchsucht die Einstellungsmenüs. Den Daumen hat er auf der linken Maustaste, den Zeigefinger auf dem Trackpad. Sie schlägt einen Menüpunkt vor, in dem es vielleicht sein könnte, doch er geht darüber hinweg. Er installiert ein Sprachpaket über die Paketverwaltung und startet auch den Download von Updates (LUG 2009j).

Die Frau, die mit ihrem Notebook gekommen ist, um sich helfen zu lassen (und die — wie sie an anderer Stelle erwähnte — bereits zuvor zweimal die LUG aufgesucht hatte), übergibt ihren Rechner dem Helfer, verfolgt mit, was er tut, macht einmal einen Vorschlag, den er nicht beachtet. Sie installiert die Sicherheitsupdates nicht selbst und fragt im später folgenden Teil des Protokolls, ob sie den Updates immer getrost zustimmen könne. Sie äußert ein grundsätzliches Interesse insofern, dass sie sich, wenn sie mal irgendwann Zeit hätte, auch gerne mehr mit dem Betriebssystem beschäftigen würde. Teile ihres Wissensstandes gibt sie darüber bekannt, dass sie nicht weiß, wie sie ein Programm installieren kann oder dass es ratsam ist, die Sicherheitsupdates zu installieren. Zudem bezeichnet sie den Befehl, den A79 eingibt, als „Codes“, was ich so deute, dass sie zwischen der Einrichtung des Betriebssystems und dem Programmieren nicht unterscheiden kann. Der Helfer verweist sie dennoch nicht auf sich selbst zurück. In einer anderen Situation bekommt eine Frau von einem LUG-Mitglied Linux installiert. Sie möchte Linux nutzen, sagt jedoch, dass sie sich nicht mit Details auskenne, sie sei kein „Technikmensch“ (LUG 2009h). In beiden Situationen treffen die Frauen auf männliche Helfer, die ihnen bereitwillig die Rechner einrichten und auch „Anfängerfragen“ akzeptieren. Dieses Verhalten der Helfer ist ein starker Kontrast zu Szenen wie der obigen (S. 102), in der der als Mann wahrgenommene A53 von zwei potenziellen Helfern unabhängig voneinander aufgefordert wird, die Antworten auf seine Fragen selbst zu recherchieren.

Es wird nicht von allen Vorwissen erwartet: Während manche für „dumme Fragen“ sanktioniert werden, bekommen andere niedrigschwellige, praktische Hilfe. In Anbetracht der dazu beobachteten Situationen liegt von meinem Standpunkt aus die Interpretation nahe, dass diejenigen, die als Frauen wahrgenommen werden, sich bedienen lassen können. Von ihnen werden nicht dasselbe technische Interesse und derselbe Einsatz erwartet wie von Männern. An dieser Stelle besteht die Schwierigkeit, dass ich die von mir als „faktisch“ wahrgenommene Geschlechterdifferenz zum Ausgangspunkt nehme, um die Herstellung von Geschlecht zu analysieren. Damit laufe ich Gefahr die Zweigeschlechtlichkeit zu reifizieren (vgl. Gildemeister 2007; Hagemann-White 2007; Villa 2007; Rostock 2005; Gildemeister/Wetterer 1992). Um die Konstitution von Geschlecht im technischen Support zu untersuchen, muss ich jedoch auf die gesellschaftlich manifeste Geschlechterdichotomie zurückgreifen, ohne sie damit biologisieren zu wollen. Ich betrachte sie vielmehr als soziale Konstruktion, deren Regeln in lebenslanger Teilnahme erlernt, verfestigt, aber auch verändert werden. Zu zeigen, *wie* diese Differenz praktisch hergestellt wird, ist eine Voraussetzung dafür, diese Prozesse zu verändern (vgl. Becker-Schmidt 2007; Gilde-

meister 2007;Gottschall 1995). Technischer Support kann in diesen Szenen somit als eine Form der Konstitution von Männlichkeit gelesen werden. Männlichkeit wird situativ darüber produziert, ob anderen technisches Wissen oder Interesse abverlangt wird oder nicht: Männlich eingeordneten Personen gegenüber männlich aufzutreten, bedeutet tendenziell, an die Fähigkeit zu appellieren, sich ein technisches Thema autodidaktisch anzueignen.¹⁴ Weiblich eingeordneten Personen gegenüber männlich aufzutreten, heißt dagegen, großzügig zu helfen und dabei auch das Problem selbst zu bearbeiten, statt z. B. das nötige Wissen dazu zu vermitteln. Dies ist einerseits eine männliche Macherrolle, andererseits aber auch eine umsorgende Form von Männlichkeit, die in Bezug auf Technologien „Geborgenheit“ geben möchte und die Verantwortung für das Tun übernimmt. Die darüber gleichzeitig hergestellte Form von Weiblichkeit ist eine hilfebedürftige, nicht aber unbedingt eine lernfähige. (Im folgenden Abschnitt (5.2.3) wird genauer herausgearbeitet, wodurch lernende Linux-Nutzer_innen tendenziell selbst in die Lage versetzt werden, technische Probleme zu lösen und wodurch zwar lokale und momentane Probleme gehoben werden, die Fragenden aber eher keinen eigenen Erkenntnisgewinn mitnehmen.) Diese Interpretation des Materials sieht eine Geschlecht konstituierende Komponente im technischen Support. Sie ist vor allem auf die Beobachtung gestützt, dass die „extremste“ Form von Geholfenbekommen bzw. sich Bedienen Lassen nur von einem Geschlecht besetzt wurde. Zudem gab es in dieser LUG nur eine Frau, die selbst im technischen Support aktiv war. Wie groß der Interpretationsspielraum jedoch oft bleibt, macht ein anderes Beispiel deutlich: Wenn mir in einer Situation jemand möglicherweise qua geschlechtlicher Zuschreibung einen Windowsrechner unterstellt hat, so wurde ich in einer anderen Situation unfreundlich darauf hingewiesen, dass ich mein technisches Problem adäquat beschreiben müsse (S. 104). Diese zweite Situation kann einerseits so lesen werden, dass ich wie viele andere regelmäßige Teilnehmer_innen behandelt werde. Andererseits kann man auch die geschlechtlich konnotierte Unterstellung in den Mittelpunkt der Interpretation stellen, nach der ich nicht in der Lage bin, ein technisches Problem zu formulieren. Länger andauernde soziale Beziehungen verändern die Vorannahmen, die Einschätzungen des Wissens anderer. „Sonderbehandlungen“ können z. B. bei regelmäßiger Teilnah-

¹⁴Bei einem Treffen der LPI-Lerngruppe wurde deutlich, dass Neulinge sich insgesamt gerne mehr bedienen lassen würden, gerne mehr Wissen aufbereitet bekämen. „A5 (...) steht mit diesem A75 zusammen und es geht um die Mitgliederversammlung. (...) A5 berichtet ein bisschen, was diskutiert wurde. Dass z. B. zur LPI-Gruppe dauernd Leute aufgetaucht waren, die einen Linux-Kurs gesucht hätten. Sie hätten deshalb auf der Website schon vor einer Weile präzisiert, dass das Treffen sich an Autodidakt_innen richte“ (LUG 2008c). Die LUG-Mitglieder verstehen — wie erwähnt — ihren technischen Support jedoch nicht als Dienstleistung.

me an den LUG-Treffen oder einem ungefähr bekannten Wissensstand aufhören.¹⁵ Die Interpretation der einzelnen Situation ist damit selten eindeutig.

Wissenstransfer und Problembhebung sind zudem dadurch komplex, dass Helfende situativ herausfinden müssen, auf welchem Wissensstand die Fragenden sind. Da die Lernprozesse im Zusammenhang mit Linux dermaßen individualisiert sind, können sie dabei nicht von einheitlichen oder linearen „Etappen“ ausgehen, in dem Sinne, dass ein Verständnis für ein Thema ein anderes unbedingt voraussetzt. Bei einem tatsächlichen oder angenommenen starken „Wissensgefälle“ finden oft Übersetzungsprozesse statt, in denen die Helfer_innen situativ Informationen geben, vorenthalten oder stark vereinfachen (ob das bewusst geschieht, sei dahingestellt).

A54 hat eine Klarsichthülle in den Händen, in der eine CD und die Zugangsdaten von ihrem Provider drin sind. Sie spielt mit der Hülle herum und weist schließlich darauf hin, dass das die Installations-CD dafür sei. A1 erklärt, dass sie unter Linux die CD nicht brauche, nur die Zugangsdaten. Der Rechner „popelt ’n bisschen vor sich hin, dann geht das.“ (...) Auf dem Bildschirm erscheinen jetzt Textzeilen, während → *grml* bootet. A54 äußert irgendetwas, dass sie das nicht verstehe, was da steht. A1 sagt, das seien „kryptische Meldungen, die will man gar nicht verstehen“, der Rechner zeige dort an, was er genau tue. (Ich bin mir sicher, dass er so ziemlich alles versteht, was der Rechner dort ausgibt.) (LUG 2009h).

Zweimal übersetzt A1 für A54, technische Prozesse in einer Weise, die ihr (A54) zu verstehen gibt, dass sie sich damit nicht befassen muss. Die Provider-Software für den Internetzugang sei nicht nötig; mit den Zugangsdaten allein „popete“ der Rechner „vor sich hin“, dann sei sie online. Die Meldungen, die beim Hochfahren des Rechners erscheinen, nennt A1 „kryptisch“, wozu ich den Kommentar vermerkt habe, dass sie für ihn höchstwahrscheinlich verständliche Informationen darstellen. Das kann so interpretiert werden, dass er mit seinen Übersetzungen Themen für A54 ausklammert, die sie überfordern könnten. Eine andere Interpretation kann lauten, dass A1 damit zu einer Mystifizierung der Technologie beiträgt. Aus derselben Szene stammt ein Materialausschnitt, der in Kapitel 4 angeführt wurde, als es um die Verwendung von Fachsprache vor Lai_innen ging. A1 nutzt nämlich *gleichzeitig* in der Kommunikation mit *anderen* Anwesenden humorvoll technische Ausdrücke, die A54 außen vor lassen. Er produziert eine Distinktion, indem er einerseits ihr gegenüber ein technisches Artefakt stark vereinfacht bzw. mystifiziert, und ihr

¹⁵Vgl. jedoch die Szenen mit dem weiblichen LUG-Mitglied A52 weiter unten, in denen dies auch wieder nicht der Fall ist.

andererseits zeigt, dass er sich sehr wohl gut damit auskennt. Zwischendurch stellt er über einen Kommentar auch zu mir einen Kontakt her:

A54 glaubt, WLAN gehe auch über Satelliten, doch A1 schickt sie in den Nachbarraum, um den → *Access Point* in der Zimmerecke zu betrachten. Sie steht auf und geht rüber. A1 sagt unterdessen zu mir: „Ach, → *Debian* funktioniert einfach immer“ (LUG 2009h).

Er bezieht sich auf den vorangegangenen misslungenen Versuch, eine andere Distribution zu installieren, für die das Notebook keine ausreichenden Kapazitäten hatte. A1 baut darauf, dass ich *Debian* Linux ebenfalls zu schätzen weiß. Eine ähnliche Wirkung hatte eine andere Szene, in der ich eine Frage an jemanden gerichtet hatte. Ein Dritter empfahl ihm daraufhin, er solle mir „was Grafisches“ geben. Ich verstand das so, dass er mir ein Desinteresse an textbasierten Programmen oder Hintergrundinformationen unterstellte. Es findet in solchen Situationen ein „Expertengespräch“ statt, das so angelegt ist, dass die Lernenden nicht mit einbezogen werden.

5.2.3. Learning by doing or learning by watching?

Nicht nur im Umgang miteinander werden unterschiedliche Lernsituationen geschaffen, sondern auch im Umgang mit den Maschinen, die im Mittelpunkt des Geschehens stehen: den Computer. Hier ist ein weiterer Aspekt prägend für die Situationen: Personen, die „sich umsorgen“ lassen, geben ihre Computer an die Helfer_innen ab (siehe A79 auf S. 106 und A54 auf S. 109). Während die wenigen regelmäßigen Teilnehmerinnen der LUG-Treffen diese „Sonderbehandlung“ nicht erfahren (A5, ich selbst), habe ich mehrfach beobachtet, dass es bei A52 sehr wohl so ist. Die schätzungsweise über 65 Jahre alte Linux-Nutzerin hat auf ihrem Notebook mehrere Betriebssysteme parallel installiert, ein Windows und mehrere Linux-Distributionen.

A52 sagt mit einer fast weinerlichen Stimme: „A3!“ (...) A3 setzt sich neben sie. Sie hat von einer → *Knoppix-CD* gebootet. Sie erzählt ihm, was geschah, irgendetwas startet nicht mehr. Währenddessen gibt er schon die ersten Befehle auf einer → *Shell* im *Knoppix* ein, *mount*, *df*, *fdisk -l /dev/hda*. Er fragt: „Wo ist */boot*?“ Sie antwortet, auf der → *Partition /dev/hda8*. Er will die Partition von Hand mounten, doch sie ist bereits gemountet. Er durchsucht sie und stellt fest, dass das eine Windows-Partition ist. A52: „Dann probier' *hda7*, nee 6, nee 7.“ Er findet schließlich auf einer Partition

das Linux-Boot-Verzeichnis. Sie zeigt auf einige Zeilen und sagt, jemand anders hätte sie dort eingefügt. Der hätte auch irgendein Passwort gesetzt, dass sie jetzt nicht wisse. A3 öffnet eine neue Shell und vergewissert sich mit dem Befehl *id*, dass er root ist. Dann ruft er nochmal *fdisk* auf. Mit dem Trackpad markiert er Partitionen und A52 erklärt ihm, was wo installiert ist. Dabei zeigt sie mit dem Finger auf dem Display, von welcher sie spricht. Es ist laut, weil gegenüber am Tisch A4 und A39 reden. A3 hat eine Theorie, auf welcher Partition → *OpenSUSE* installiert ist. A52 sagt, das könne nicht sein. Sie sucht in ihrem Hefter nach einem Zettel, auf dem sie das aufgeschrieben habe. A3 fragt, welches System sich booten lasse. Sie sagt, keins. Er stellt fest, dann müsse der → *Bootloader grub* noch einmal installiert werden. Er fragt sich, wie das nochmal geht, mit dem *chroot*. Er ruft *grub-install -help* auf und liest nach. A52 hält ihm einen Ausdruck zu der Frage aus ihrem Hefter hin und redet. Er hält sich die Ohren zu. A52 verstummt und A3 versucht, sich zu konzentrieren. Er hat beide Hände auf der Tastatur. Beim Lesen der Ausgabe auf dem Bildschirm reckt er den linken Zeigefinger mitlesend Richtung Bildschirm. Er erklärt ihr den *chroot*-Befehl und zeigt dabei auf den Bildschirm. Sie schreibt sich in ihren Hefter, wie der richtige Befehl lautet. A3 startet den Rechner neu, A52 nimmt die *Knoppix-CD* heraus und der Bootloader erscheint wieder. Sie schlägt die Hände in einer dankbaren Geste zusammen und sagt: „Ich danke dir!“ (LUG 2009j).

A52 hat bereits versucht, sich ihrem Problem selbst zu nähern, indem sie das Notebook von einer Live-CD gestartet hat und die Partition, die sie für die richtige hält, eingebunden hat. Dann zieht sie einen Helfer hinzu, den sie direkt mit seinem Namen anspricht. Damit und mit ihrem „weinerlichen“ Ton fordert sie ihn auf, ihr zu helfen. Ich interpretiere die Szene besonders im Hinblick auf die unterschiedlichen Geschwindigkeiten der Beteiligten und auf die explizierten Informationen: Der Tempounterschied ist groß: A52 ist noch bei der Problembeschreibung, als A3 sich schon ihren Rechner genommen hat und ihn untersucht. Zur Beantwortung seiner gezielten Fragen sucht sie nach einem Notizzettel, den A3 schon nicht mehr benötigt, als sie ihn gefunden hat. Was den Wissenstransfer in dieser Situation angeht, kann A52 sich am Ende notieren, wie A3 den Bootloader neu installiert hat. Seinen Rechercheprozess macht er ihr nicht explizit transparent und sie fragt auch nicht danach: Wie er sich z. B. die Partitionen angesehen hat, um die *boot*-Partition zu finden, erläutert er nicht.¹⁶ Wie sehr oft, basiert die entscheidende Frage, welches Betriebssystem überhaupt noch startet, wahrscheinlich auch auf seinen praktischen

¹⁶Dazu muss gesagt werden, dass die beiden sich aus der LUG bereits kennen und dass A52 nicht unbedingt dafür bekannt ist zuzuhören.

Erfahrungen: Erfahrungen mit Installationen und Erfahrungen bei der Eingrenzung von Problemen. Dieses Wissen *ist* sehr schwer weiterzuvermitteln, da es situations- bzw. problemabhängig ist. Gleichzeitig ist es eine der „Schlüsselkompetenzen“, um die es in der Linux User Group geht. Die Tatsache, dass der Helfer den Rechner bedient, trägt allerdings dazu bei, dass er sein Erfahrungswissen nicht explizieren muss: Wenn A52 ihr Notebook selbst in der Hand hätte, müsste er ihr die Schritte zumindest sagen, durch die sie an die Informationen kommen könnte.

Es zeigen sich unterschiedliche Praktiken des Helfens daran, *ob* Helfende fremde Rechner „übernehmen“ oder ob sie sich die Zeit nehmen, der anderen Person jeden Schritt zu erklären oder sie zumindest vorzugeben, so dass sie die Schritte zur Problemlösung selbst eingeben kann. Für den die Helfer_in wird die Situation durch das Selbstmachen erleichtert: Gerade, wenn man sich selbst noch nicht sicher ist, will man die Person, der man helfen möchte, nicht durch falsche Fährten verwirren, sondern den Weg zeigen, der es dann auch ist. Um andere Möglichkeiten auszuschließen, schaut man also schnell selbst nach.¹⁷ Zudem müssen Helfer_innen, die selbst Hand anlegen, sich nicht in demselben Maße mit dem Wissensstand der fragenden Person auseinandersetzen; sie können leicht Schritte unerwähnt tätigen und abwarten, ob überhaupt eine Erklärung eingefordert wird.¹⁸

In einer anderen Situation wird an der Körperhaltung des Helfers deutlich, dass er den Rechner unter Kontrolle hat, nicht die Besitzerin: Er stützt sich so auf den Tisch vor dem Rechner, dass er ihr den Zugriff verweigert; er besetzt dadurch den Raum um den Laptop herum, gestattet ihr den Zugang für den Moment der Passworteingabe und schränkt ihn dann gleich wieder ein. Den eingenommenen Raum am Platz von A79 steckt er mit seinem Körper ab. Sein eigener Rechner dient ihm derweil als Platzhalter an einem anderen Platz am selben Tisch:

Er [der Helfer] stützt sich mit einem Arm auf den Tisch und zwar so, dass er Raum zwischen ihr und ihrem Rechner einnimmt. Sie sitzt versetzt zum Rechner, nicht di-

¹⁷In manchen Fällen wird so viel ausprobiert, dass am Ende gar nicht klar ist, welcher Schritt eigentlich das Problem behoben hat: „Ich frage A26, was nun der Clue war, warum es plötzlich ging. Er sagt, er weiß es nicht. Er habe einfach herumprobiert und plötzlich sei es gegangen. (Ich frage mich, wie ich das lernen soll, wenn der, der es schafft, es nicht erklären kann)“ (Fir 2008c).

¹⁸Hinzu kommt, dass manche Personen, die sich bei der Problemlösung helfen lassen, offenbar ein schlechtes Gewissen haben, weil sie so viel Zeit der Helfer_innen beanspruchen. Dass das Problem an einem Abend noch gelöst werden kann, rückt dann u. U. in den Vordergrund vor dem Anliegen, auch zu verstehen, *wie* es gelöst wird.

rekt davor. Er bedient den Rechner mit seiner freien Hand und editiert die Netzwerkverbindung über den → *Network-Manager*. Dann startet er eine → *Shell* und startet das Netzwerk neu. Er nimmt seinen Arm weg, damit sie das Passwort eingeben kann, Dann stützt er sich wieder auf den Tisch. Sie hatte ihm kurz vorher auch angeboten, sich auf ihren Stuhl zu setzen, aber er bleibt stehen. (...) Er beugt sich abwechselnd über A78s Laptop und tippt, geht dann wieder zwei Plätze weiter, wo sein Notebook steht, schaut dort etwas nach, beugt sich wieder rüber, tippt weiter, ca. drei- oder viermal, dann ist der Befehl vollständig, der die Uhrzeit umstellen soll (LUG 2009j).

Hier hat der Helfer seinen Rechner zwei Plätze weiter auf dem Tisch stehen und geht zwischen den beiden hin und her. Das führt einerseits dazu, dass er seinen Sitzplatz für später reserviert, indem der Computer markiert, dass der Platz besetzt ist.¹⁹ Andererseits nimmt A79 viel Raum ein, während er in dem eher schmalen Raum der LUG ständig den Platz wechselt und zwei Rechner parallel bedient, anstatt seinen Rechner, der als Informationsquelle dient, an A78s Platz zu holen. Seine Hilfepraxis findet innerhalb einer „gegenständlich-technischen Ordnung“ (Schmidt 2008: 291) aus Hardware und Möbeln im Raum statt, die den Rahmen dieses kurzzeitigen Arrangements festigt und ihm Raum gibt.

Die Helfer_innen nehmen zu den ihnen anvertrauten Computern und darüber auch zu den Fragenden verschiedene Verhältnisse ein: Während die einen ausschließlich selbst an den fremden Rechner agieren, „übernehmen“ andere Helfende sie nach und nach.²⁰ Manche diktieren Befehle und greifen der anderen Person bei Fehlern oder aus Ungeduld in die Tastatur.²¹ Wieder andere holen sich eine Erlaubnis oder bieten zumindest durch eine rhetorische Frage die Möglichkeit, den Eingriff abzulehnen.²² Helfer_innen, die im Besitz des fremden Rechners sind, installieren teilweise ohne Rückfrage Programme²³, teilweise holen sie sich auch hier die Erlaubnis.²⁴

¹⁹Für diesen Interpretationsvorschlag danke ich Frank Kleemann.

²⁰„Zuerst fordert A77 A76 auf, seine Hardwaredetails ausgeben zu lassen, mit *lspci*. A76 tippt ‚ls pci‘. A77 sucht eine Zeile heraus (ich glaub, es war der Prozessor) und sagt: ‚Tante Google‘. A76 tippt auch einmal, ohne den Cursor in der Shell zu haben. Nach ein paar Minuten steht A77 vorgebeugt frontal vor dem Laptop. Seitlich oder dahinter sind wir anderen“ (LUG 2008c).

²¹„A87 greift diverse Mal auf A86s Tastatur, wenn es ihm nicht schnell genug geht“ (LUG 2009f). Umgekehrt habe ich einen Fragenden beobachtet, der durch seine Körperhaltung solche Eingriffe verhindert: „A53 erläutert ihm die Schritte und zeigt dabei mit den Fingern auf A6s Bildschirm. A6 hört zu und hält dabei seine Finger still direkt über der Tastatur. (Ich denke: So verhindert er, dass jemand ihm einfach in die Tastatur greift. Ob ihm das bewusst ist? Ich vermute, nicht.)“ (LUG 2009g)

²²„Bei Übergriffen auf die Tastatur von A7 fragt er: ‚Darf ich mal?‘ oder ‚Mal kurz gucken...‘“ (LUG 2007d).

²³„Mir fällt ein, ja, A32 benutzt den Editor *joe*. Er hat ihn mal bei mir installiert, als er mir WLAN eingerichtet hat, weil ihm der fehlte — eine der Installationen, die auf meinem Rechner ohne Nachfrage erfolgten (Fir 2008e).

²⁴„Er sagt, es gebe kein *rsync* auf A44s Rechner. A44: ‚Hols dir!‘ (LUG 2009h)

A6 sitzt neben einem etwas älteren Mann. Ich frage mich, ob es derselbe ist, der mir vor einigen Wochen schon aufgefallen war. (...) A6 tippt auf seinem Thinkpad den Befehl *apt-cache policy* ein. A51 beugt sich leicht herüber, liest ihn und gibt ihn dann auf seiner eigenen Tastatur ein. Er vergisst das Leerzeichen. A6 sagt es ihm und greift auf A51s Tastatur. Mit der Pfeiltaste bewegt er den Cursor an die entsprechende Stelle und sagt A51 auf englisch: „Space.“ A51 versteht offenbar nicht, dass der Cursor schon richtig positioniert ist, sondern greift daraufhin zu seiner Maus. Er führt sie an die Stelle, an die das Leerzeichen soll und klickt. Nichts geschieht und A51 probiert das gleiche noch einmal. Ich bekomme mit, dass sie nachsehen, ob das Brennprogramm *k3b* bei A51 schon installiert ist, es dann installieren und etwas brennen wollen. Sie machen es mehrmals so, dass A6 einen Befehl bei sich eingibt und A51 ihn dann abschreibt. Dabei schickt A6 den Befehl nicht unbedingt auch bei sich ab, sondern nutzt seine Shell nur als Vorlage für A51. (...) (Beim Protokollieren finde ich es bemerkenswert, dass A6 A51 diese Schritte auf der Kommandozeile gezeigt hat, nicht mit einem grafischen Programm. Ich bin mir sehr unsicher, wie viel A51 von den Befehl versteht oder behält. Er stellt keine Nachfragen.) A6 sieht genauer hin, was A51 eingetippt hat und greift wieder auf dessen Tastatur hinüber und löscht die letzten Buchstaben. Er weist A51 darauf hin, dass das Programm *k3b* heißt (LUG 2009i).

In dieser Sequenz zeigt A6 A51, wie man ein Programm installiert. A51 ist offensichtlich den Umgang mit der Shell nicht gewohnt, er weiß z. B. nicht, dass er den Cursor nicht mit der Maus positionieren kann. Sein Helfer schwenkt dennoch nicht auf ein grafisches Programm um, mit dem die Aufgabe auch (und wahrscheinlich schneller) zu lösen wäre, sondern schreibt den korrekten Befehl als Vorlage in seiner Shell auf. Er lässt damit A51 weitgehend die Hoheit über seinen Computer, greift jedoch manchmal einfach korrigierend ein, wenn er — so vermute ich — merkt, dass A51 ihm nicht folgen kann, steht er doch vor der Situation, dass A51 keine Fragen einwirft.

Während ich in der IT-Abteilung, beim → *Pair Programming*, eine ganz klare Rollenverteilung beobachtet habe²⁵, die durch den professionellen Rahmen der Situation gegeben sein mag, werden die Rollen der Beteiligten in der LUG ständig praktisch bestimmt und neu ausgehandelt. Dabei geht es nicht nur darum herauszufinden, auf welchem Wissensstand die fragende Person ist, sondern auch um eine Klärung, wer wie auf dem Computer dieser Person agiert, was so gut wie nie explizit besprochen wird, sondern sich in der Praxis ergibt. In der LUG ist das Vertrauen in die Helfer_innen ein Bestandteil die-

²⁵ „Wer an der Tastatur sitzt, hat diese auch allein in der Hand. Nie versucht der jeweils andere, etwas zu tippen, es werden nur mündlich Vorschläge beigesteuert oder auch diktiert“ (Fir 2008f).

ses Prozesses. Während auf manchen Community-Events tatsächlich Computer über das Veranstaltungs-WLAN gehackt werden und der in Abschnitt 6.2.1 beschriebene Hacking-Contest zeigt, wie viele Anwesende ein Verständnis von → *Backdoors* haben, wird in der LUG darauf vertraut, dass diejenigen, die Zugriff auf fremde Rechner haben, diesen nicht missbrauchen.

In diesem Abschnitt ging es darum, wer bei einer Hilfestellung den Rechner bedient. Bei der Analyse der breit gefächerten Praktiken aus diesem Zusammenhang wurde deutlich, wie komplex die Kleingruppenarbeit in Linux User Groups ist: Je nach dem, wer vor dem zu behandelnden Computer sitzt, entstehen unterschiedliche Konsequenzen für Lernende und Helfende. Helfer_innen eröffnen einen engeren oder breiteren Horizont. Ihre Gründe oder Intentionen sind nicht Gegenstand des Interesses. Die teilnehmende Beobachtung hat jedoch neben der Vielzahl verschiedener Praktiken bei den Helfer_innen eine große Bandbreite an Geduld, Zeitbudget, Wissensständen, Problemlösungskompetenzen, pädagogischen Methoden oder Erfahrungen zutage gefördert, die mit darüber entscheiden können, wie die Kooperation ablaufen wird.

Zu einem engeren Horizont, der einem_r Hilfesuchenden eröffnet wird, gehört es, dass ein technisches Problem direkt von den Helfer_innen behoben wird. Teilweise machen sich die Personen, die mit der Frage gekommen waren, Notizen. Zu einem breiteren Horizont gehört nicht nur, dass die fragende Person ihren Rechner selbst bedient, sondern z. B. die Unterstützung bei der Eingabe von Befehlen und dem Erlernen korrekter Syntax, aber auch das schrittweise Eingrenzen eines technischen Problems. Das Erlernen der Fehlersuche ist ein praktischer, problembezogener Prozess, der oft beim technischen Support in der LUG nicht oder nur ein Teil expliziert wird. Hier fließen auch die Selbstverständlichkeiten und das Grundlagenwissen ein, die Helfer_innen schon nicht mehr für erklärungsbedürftig halten. Es wurde deutlich, dass es keine einheitliche Praxis im Umgang mit fremden Computern gibt: Manche Helfer_innen lösen Probleme selbst und bedienen dabei nicht nur den Rechner, sondern markieren auch körperlich, dass dies jetzt ihre Domäne ist. Andere sind viel zurückhaltender und lassen den Fragenden ihren Raum, ihren Rechner und/oder ihre eigene Geschwindigkeit. Insgesamt bleibt beim Support ein großer Teil dessen, was die praktische Problemlösungskompetenz der Helfer_innen ausmacht, unausgesprochen, so dass oft unsichtbar oder implizit bleibt, *wie* sie eigentlich Probleme lokalisieren und lösen. Wieviel Problemlösungskompetenz übertragen wird, hängt mit davon ab, ob die Fragenden selbst die Maschine bedienen.

Die Helfer_innen sind jedoch auch mit Situationen konfrontiert, die ihnen sehr viel abverlangen. Sie erbringen unzählige kleine Leistungen, die *ebenfalls* meistens unsichtbar bleiben. Ein Blick auf diese Leistungen kann Einblicke in die praktische Logik des technischen Supports gewähren, ein Support, bei dem häufig Informationen verkürzt werden oder zwischendurch „Expertengespräche“ geführt werden, die ähnlich wie Witze einen „*I-get-it*“-Effekt unter Ebenbürtigen herstellen. Die Informalität des Settings führt für diejenigen, die um Hilfe gebeten werden, in sehr komplexe Lagen: Sie haben selbst individuelle Lernprozesse durchlaufen, sind auf unterschiedlichen persönlichen Wissensständen und werden von oft Unbekannten angesprochen. Sie werden gebeten, spontan konkrete Probleme zu lösen oder zu erklären oder eine ganze Frageliste mit jemandem durchzugehen. Sie müssen dabei jedoch (soweit sie neben einer unbekannt Person sitzen) erst situativ ausloten, welches Vorwissen die Person mitbringt, wie viele Informationen sie sich wünscht, wann sie überfordert ist. Zudem ist das Angebot an freier Software so groß, dass sich dabei ein großer Teil der Fragen auf konkrete Programme bezieht, die dem_der Helfer_in nicht bekannt ist. Versuche, diese praktische Logik zu explizieren, habe ich nicht beobachtet (vgl. Bourdieu 1999: 151 ff.).²⁶

Helfer_innen stehen also vor der Herausforderung, ein technisches Problem zu behandeln, dabei jemandem eine passende Fülle an Informationen dazu zu geben und darauf zu achten dass die Kleingruppenarbeit in einem angemessenen zeitlichen Rahmen bleibt. Sie filtern Informationen, schränken die mögliche Reichweite eines Themas ein, laufen die Gefahr, dass Wissenslücken bleiben. Sie wissen wenig über ihre wechselnden Gegenüber und unterstellen ihnen Wissensstände, zu denen die fragenden Personen sich verhalten können.

Die Helfer_innen stellen implizit vielfältige Differenzen in der Praxis des technischen Supports her: Ihr Verhältnis zu der fragenden Person gestaltet sich entlang der Fragen, ob sie den fremden Rechner einfach nehmen, eine Erlaubnis einholen oder die Person beim Agieren anleiten, ob sie körperlich viel Raum um den Computer einnehmen, ob sie ihr eigenes Tempo setzen oder sich auf eine geringere Geschwindigkeit einlassen, ob sie Schritte erläutern und wie viele. Diese Unterschiede enthalten aufgrund des informellen Lernkontextes unausgesprochene Annahmen über die Personen, denen sie helfen. Sie spiegeln wider, wie viel sie ihnen zutrauen, wie viel sie von ihnen erwarten und produzie-

²⁶Es gab jedoch teilweise kleine Vorbesprechungen oder Voruntersuchungen eines Problems, auf die eine feste Verabredung für die nächste Woche folgte, weil sich ein_e Helfer_in vorbereiten wollte (vgl. LUG 2008d).

ren damit Normvorstellungen dessen, was an Vorwissen vorhanden sein müsste.²⁷ Die Wissensunterschiede sind die Basis, auf der der Austausch in der LUG stattfinden kann, die konkreten Praktiken der Wissensvermittlung wirken jedoch sehr unterschiedlich. Auf mich persönlich wirkten sie öfter so, dass ich mich nicht ernst genommen fühlte (z. B. „Gib ihr was Grafisches“, S. 110). Die Frage, entlang welcher Kategorien diese Differenzierungsprozesse laufen, ist kaum eindeutig zu beantworten: Die vorgestellten Materialauschnitte ergeben höchstens eine *Tendenz*, dass Frauen häufiger ihre Computer den Helfer_innen (und in diesen Szenen männlichen Helfern) überlassen. (Ein Gegenbeispiel ist A44 (S. 101), der sein Notebook ebenfalls verschiedenen Helfer_innen übergibt.) Die einzelnen Schritte selbst unter Anleitung einzugeben, ist eine Lernendenrolle, die in dieser Tendenz häufiger von Männern eingenommen wurde. Deutet man das als Herstellungsprozesse von Geschlecht im Lernprozess, so bedeutet Männlichkeit hervorzubringen hier *tendenziell*, die Hoheit über das eigene technische Gerät aufrechtzuerhalten. Jemanden als männlich wahrzunehmen, impliziert eher, den Rechner der Person nicht einfach zu vereinnahmen. Dabei gibt es jedoch — entgegen der Eindeutigkeit — graduelle Unterschiede, etwa jemandem grundsätzlich die Hoheit über den Computer zu lassen, dann aber in die Tastatur zu greifen.

Die Rolle der technischen Artefakte wird später genauer unter die Lupe genommen. Dabei wird es auch um die Frage gehen, wie Artefakte die Situationen mit strukturieren. An dieser Stelle nehmen sie die Rolle passiver, herumreichbarer Gegenstände ein: Sie sind das Material, über das Differenz in Form von Rollenverteilungen produziert wird, für die einen eine *black box*, für andere beherrschbare Maschinen.

5.2.4. Technischer Support als Selbstinszenierung vor Publikum

Die kleinteiligen Differenzierungsprozesse in der Praxis des technischen Supports in der LUG ergeben sich, wie oben gezeigt, oft aus den Wissensunterschieden zwischen den Beteiligten. Sie laufen auf einer sehr impliziten Ebene ab, stehen unter dem Zeichen der informellen Organisation von Lernen und den hohen Anforderungen, denen sowohl Helfer_innen als auch Hilfesuche sich aussetzen. Im Folgenden wird eine weitere Dimension in diesen Praktiken beleuchtet, die stilistische Distinktion.

²⁷Hier wird z. B. auf ein vermutetes Vorwissen aus dem Windows-Bereich zurückgegriffen: „A54 fragt, was *Nautilus* ist. A1 ist gerade zurück am Tisch und sagt ‚ein Arbeitsplatz‘, so wie sie unter Windows einen ‚Arbeitsplatz‘ kenne“ (LUG 2009h).

Ich sehe, dass A86 jetzt neben einem jungen Mann (A87, Mitte 20, blonde Haare, etwas länger und strähnig, ein strubbeliger Vollbart, freundliches Gesicht) sitzt und von ihm etwas erklärt bekommt. (...) A86 hat eine Shell offen. Der junge Mann fällt mir total mit seiner Redeweise auf. Ich vermute, dass er A86 bestimmte Aktionen bereits erklärt hat und jetzt nicht jeden Befehl diktieren, sondern das vermittelte Wissen abrufen möchte. Er sagt zum Beispiel, A86 solle in das Verzeichnis *media* gehen, A86 tippt ein ‚media‘. Daraufhin wiederholt A87: „Gehe in das Verzeichnis! Wie gehst Du in ein Verzeichnis?“ Er erinnert mich ein bisschen an einen Trainer, der nicht unbedingt freundlich ist, aber vielleicht dennoch Dinge so vermittelt, dass sie hängen bleiben. Teilweise scheint A87 aber auch lustig sein zu wollen für die Mithörenden. Mir kommt es so vor, als sehe er mich manchmal beifallheischend an. Ich versuche, seinem Blick dann nicht zu begegnen, weil mir der Umgang mit dem Lernenden nicht so ganz geheimer ist. „Keine Rückmeldung ist eine gute Rückmeldung.“ „Leerzeichen sind böse“ - der Spruch bezieht sich eigentlich auf Pfade, A86 fehlt diese Erklärung jedoch, so dass er keine Leerzeichen zwischen dem Programm und dem Argument eingibt. A86 soll, wenn es nach A87 geht, gleich etliche Dinge auf einmal lernen: wie er sich per Befehl in der Verzeichnisstruktur bewegt, wie er mit der Tabulatortaste Befehle oder Pfade vervollständigt (bzw. wann das sinnvoll ist und wann nicht), wie er den Editor *vim* bedient. A87 greift diverse Mal auf A86s Tastatur, wenn es ihm nicht schnell genug geht. Um die Datei */etc/fstab* zu editieren, gibt A86 nur den Dateinamen ein. A87 sagt, davor müsse ein „Editor der Wahl“ stehen. Dies sei der Editor seiner Wahl - und gibt *vim* ein. Dann geht auch gleich mit ‚i‘ in den *Insert*-Modus, ohne diesen Schritt zu erklären (LUG 2009f).

In dieser Szene möchte A87 der Person, der er hilft, bestimmte Befehle beibringen, z. B. wie man durch Verzeichnisse navigiert oder Dateien von der → *Kommandozeile* aus in einem Editor öffnet. Er fragt bereits Erklärtes ab und formuliert „Merksätze“. „Keine Rückmeldung ist eine gute Rückmeldung“ bezieht sich beispielsweise darauf, dass ein Befehl erfolgreich ausgeführt wurde, wenn keine Rückmeldung in der Kommandozeile erscheint. Meiner Interpretation nach erklärt er dabei zu wenig: A86 missversteht, wann Leerzeichen zu setzen sind und wann nicht. A87 trifft mit dem Editor *vim* eine Auswahl für A86. Er öffnet nicht nur selbst den Editor, sondern versetzt ihn auch unkommentiert in den richtigen Modus, damit A86 überhaupt etwas eingeben kann.²⁸ Neben einer Ungeduld, die

²⁸Praktiken mit dem Texteditor *vim* werden in Abschnitt 6.1.2 genauer betrachtet. Der Editor ist nach dem Programmstart nicht sofort im Eingabemodus, was für Anfänger_innen ungewohnt ist. Dass A87 seinen Eingriff nicht erläutert, hat mit ziemlicher Sicherheit zur Folge, dass A86 diesen Schritt später nicht reproduzieren kann.

demonstriert, dass A87 den Rechner selbst schneller bedienen könnte, ist die Kommunikationsweise von A87 auffällig. Seine „beifallheischenden“ Blicke auf Zuhörer_innen lassen viel Deutungsspielraum: Will er zeigen, dass seine Lehrmethoden erfolgreich sind? Will er sein eigenes technisches Wissen betonen? Welchen Kontakt sucht er zu den Zuhörern? Seine Intentionen müssen unklar bleiben, seine Lehrpraxis hatte aber aus meiner Perspektive eine deutliche Komponente von Selbstinszenierung als Experte. Bei allem Wissenstransfer, den er in dieser Situation leistet, macht er sich gleichzeitig die Person, die er mit technischem Support unterstützt, zunutze, um Aufmerksamkeit oder Anerkennung derjenigen zu bekommen, die nicht unmittelbar beteiligt sind. Inszenierungen vor Publikum und überhaupt den Eindruck, *dass* ein Publikum anwesend ist, habe ich auch in anderen Situationen dokumentiert.²⁹

Eine Szene zeigt einen kleinen Wettbewerb um den besten Lösungsweg. Sie ereignete sich nicht in einer LUG, sondern auf einem Community-Event, wo vor der Keysigning-Party (vgl. Abschnitt 7.2) Fragen rund um das Thema Keysigning gestellt werden konnten. Es geht um die Bedienung des Programmes *caff*, mit dem das Signieren der überprüften Schlüssel automatisiert werden kann. Aus der Teilnehmer_innenliste einer Keysigning-Party mit den ganzen Key-Informationen sollen nur die Key-IDs in eine Datei geholt werden, damit *caff* sie verarbeiten kann.

Irgendwann, als es gerade um das Programm *caff* geht, mit dem man automatisiert Keys signieren kann, mische ich mich ein, indem ich frage, wie man das Programm verwendet. Sie erklären es zunächst verbal, man brauche eine Textdatei, in der nur die Key-IDs stehen. Die könnte man ja sehr einfach aus der Teilnehmer_innenliste erstellen. A57 sagt, er habe dazu 3x cat (oder cut?) hintereinander gehängt, aber das gehe wahrscheinlich auch einfacher. A21 (und auch die anderen) scheint der Vorschlag zu belustigen. Sie beginnen zu diskutieren, wie jeder von ihnen das lösen würde. A10 fragt schließlich, wer einen Rechner dabei habe. (...) A21 sitzt am Rechner, stellt eine WLAN-Verbindung zum Internet her und lädt die Teilnehmer_innen-Liste der heutigen Keysigning-Party herunter. Er öffnet eine → *Konsole* und beginnt in etwa so:

```
sed '/^pub/p,s/.*\|\\(usw.) .*/
```

Die anderen lesen mit und kommentieren. Schließlich steht da

```
sed -n '/^pub/p,s/.*\|\\(usw.) .*\|1/' datei.txt
```

²⁹A1 kommt in diesen Szenen öfter vor als andere LUG-Mitglieder, mir sind jedoch auch andere aufgefallen: „Ich sehe, dass schräg hinter mir A41 eine kleine ‚Show‘ vor Publikum hat. Dort versuchen sie etwas mit *ndiswrapper*“ (LUG 2009j).

Der Befehl wird nicht so ausgeführt, wie die drei es erwartet hatten. A21 probiert weiter herum, die anderen helfen bei der Fehlersuche. A10 kann es kaum erwarten, selbst an das Notebook zu kommen, um seinen Lösungsweg vorzuführen. (...) A21: „Einen Versuch hab ich noch.“ Er probiert es anders:

```
awk -F "/" '{ print $2 }' usw.
```

doch auch das klappt nicht so wie es soll. „Jetzt darfst du.“ A10 setzt sich an das Notebook. Er durchsucht mit dem Programm *grep* die Datei nach dem Begriff ‚pub‘ und leitet das Ergebnis weiter in den Editor *vim*. Dort markiert und löscht er alle überflüssigen Spalten mit ein paar Shortcuts. Was übrig bleibt, sind die Key-IDs. Einer der anderen kommentiert: „Sieht Kacke aus.“ Vermutung: Er gönnt es A10 nicht, dass der so einen schnellen Lösungsweg gefunden hat, der viel einfacher gedacht ist als die langen Befehle von A21. Er kritisiert, um das mitzuteilen, die Darstellungsform der IDs, die hintereinander weg stehen, nicht untereinander, eine pro Zeile. Inzwischen füllt sich der Raum mit den Teilnehmer_innen für die Keysigning-Party. A10 sagt darauf: „Einen hab ich noch.“ Er greppt wieder und holt mit „| vim“ die Ergebnisse in den Editor. Dann formuliert er schnell ein kleines Skript:

```
for key in datei.txt; do gpg --recv-keys $key; done
```

Damit lädt er alle Schlüssel herunter. Er sagt, dasselbe könnte man noch machen, um die Keys zu signieren. A21 fragt: „Kann ich nochmal ran?“ Er will es mit *sed* noch einmal probieren. „Ach so, ich weiß, warum das nicht geht.“ A10 schlägt vor: „Du kannst auch pipen.“³⁰ A21: „Das sieht uncool aus.“ A57: „Jede Lösung, die geht, ist cool“ (KSP 2008c).

A21 versucht, das erwünschte Ergebnis durch eine einzige komplexe Befehlszeile zu erhalten. A10 nimmt dagegen einen Editor zur Hilfe und löscht von Hand Teile des Dateiinhaltes. Obwohl er sehr schnell zum benötigten Zwischenergebnis kommt, wird sein Lösungsweg kritisiert, weil er nicht gut aussehe. Derjenige, der die Kritik äußert, findet wohl, dass ein gut aussehendes Ergebnis eine ID pro Zeile ausgibt, nicht einen Block von IDs, die nur durch Leerzeichen getrennt sind.

Darüber hinaus geht es hier, so meine Sicht auf dies Szene, um die Eleganz des Lösungsweges: Wer schafft es, die IDs mit nur einem Befehl aus der Datei herauszufiltern? Wer beherrscht die Komplexität textbasierter Programme? Ein weiteres Kriterium für diese

³⁰Es geht um die Aneinanderreihung mehrerer *sed*-Befehle, durch Pipe-Zeichen getrennt (|). Dabei wird die Ausgabe des ersten Befehls direkt als Eingabe für den zweiten verwendet etc.

Art von Ästhetik ist, ob sich ein Lösungsweg automatisieren lässt, z. B. in einem → *Shell-skript*. Dies ist nicht der Fall, wenn zwischendurch eine Datei von Hand bearbeitet werden muss.³¹ Hier wird ein Unterschied zwischen verschiedenen Lösungswegen mit einer Wertung versehen.³² Das manuelle Vorbereiten der Originaldatei mit einem Editor wird gegenüber dem direkten Herbeiführen des Ergebnisses durch einen Befehl von A21 abgewertet. A57 lässt dagegen jeden funktionierenden Weg gelten. Damit misst er dem manuellen Weg von A10 Bedeutung bei, denn die Versuche von A21 führen ja zu keinem Ergebnis, weil keiner der Anwesenden so versiert mit Programmen wie *sed* oder *awk* umgehen kann, dass das Problem in einigen Minuten gelöst wird. Einige halten also den komplexeren Lösungsweg für den besseren oder ästhetischeren, obwohl er zur Lösung dieses konkreten Problems *nicht* spontan erfolgreich eingesetzt werden kann. A10 beweist zwar seinen praktischen Sinn, stößt jedoch auf einen abweichenden ästhetischen oder technologischen Sinn einiger anderer. Es entsteht eine Konkurrenzsituation, die sicher auch davon berührt ist, dass sich die Beteiligten bereits kennen. Sie lässt sich aus der beobachtenden Perspektive jedoch vor allem an der Konkurrenz dieser verschiedenen Herangehensweisen begreifbar machen: einer einfacheren, praktisch funktionierenden und einer technisch komplexeren, die jedoch nicht erfolgreich zu Ende geführt wird.

Hinten links sitzen A5, A6 und ein älterer Mann (A93, wie sich herausstellt). Sie haben eine *Ubuntu*-→ *Live-CD* laufen und diskutieren über die Partitionierung der Festplatte von A93s Rechner. A5 sitzt in der Mitte und hat die Maus in der Hand. Sie redet auch am meisten und diktiert A93 Befehle: „*fdisk -l*, ja, mit Leerzeichen.“ Das Gespräch wiederholt sich: A93 erklärt mehrfach, wie es dazu kam, dass die Partitionierung komisch lief, es seien *Windows ME* und *Windows XP* installiert, die funktionierten auch weiterhin. Mit der *OpenSuse-CD* habe er partitioniert, doch die Installation sei dann abgebrochen. A5 wiederholt sich auch und sagt, sie sehe, dass da etwas nicht stimme, aber dass sie die *Windows*-Partitionen nicht kaputt machen wolle, deshalb würde sie lieber an A1 abgeben. (...) Schließlich kommt A1 herein und alle rufen ihn herbei. (Interpretation: Jetzt hat er mal wieder einen Auftritt. Er wird wenige Minuten brauchen, um das Problem zu lösen und die Installation zu starten.) Er geht an den Rechner und ruft *cfdisk* auf, also ein anderes Partitionierungsprogramm für die Kommando-

³¹Die hier angedeutete Vorstellung von schönem Code oder einem eleganten Lösungsweg wird in Abschnitt 6.2 unter dem Aspekt ästhetischer Vorstellungen vertieft.

³²In anderen Situationen wurden nicht nur funktionierende Lösungen unterschiedlich bewertet, sondern auch ein Unterschied geltend gemacht zwischen Lösungen, „quick and dirty hacks“ und Workarounds. Ein „quick and dirty hack“ ist eine schnell umzusetzende, aber unsaubere (bestimmten Konventionen widersprechende) eher temporär gedachte Lösung eines Problems. Ein Workaround löst ein Problem gar nicht, sondern umgeht die Fehlerursache.

zeile. Er bespricht mit A93, was er da sieht, welche Partitionen primär und welche logisch sind. Ich stehe dabei, der Neue auch, der fragt auch nach, warum A1 die Platte mit dem Dateisystem *ext3* formatiert. A1 antwortet (wie ich finde) unkonventionell: Weil er *ext3* gut finde, so wie er *Gnome* als Desktopmanager gut finde. Der Neue: „Das ist eine gute Antwort.“ (Ich denke: Hm, ich hätte jetzt erwartet, dass er was von *Journaling* sagt.) Mir fällt auf, dass A5 sich an einen anderen Tisch zurückzieht und die Lösung des Problems nicht bis zu Ende verfolgt. Ich frage mich, wie sie eigentlich zu A1 steht. Dann löscht er die Linuxpartitionen nochmal und legt neue an. Er hantiert extrem schnell mit dem Programm, ich sehe, dass er es sehr gut kennen muss. (...) Es ist noch ein etwas älterer Mann aufgetaucht, schätzungsweise zwischen Mitte und Ende 50. Der scheint auch neu zu sein und fragt uns alle nacheinander, was wir da gerade machen und wie lange wir dabei sind. Mir fällt auf, dass A6 sagt, er sei „blutiger Anfänger“ (LUG 2008d).

Zur Interpretation: In dieser Szene ist die einzige Frau, die regelmäßig an den LUG-Treffen teilnimmt, zunächst die Haupthelferin, was durch ihr technisches Wissen, ihren zentralen Sitzplatz und den direkten Zugang zum fraglichen Computer markiert wird. Sie möchte das Problem delegieren und zieht sich zurück. A1 wird von der ganzen Kleingruppe herbeigerufen, als er hereinkommt, wodurch ein Publikumseffekt entsteht. Vor vier Personen führt er seinen routinierten Umgang mit dem textbasierten Partitionierungsprogramm vor.³³ Auf die Zwischenfrage eines Zuschauers hin wartet er nicht mit einer technischen Erklärung der Vorzüge eines bestimmten Dateisystems auf. Stattdessen beruft er sich auf seinen Geschmack, der für den Fragenden Gewicht hat. Interessant ist, dass A6 sich im Nebengespräch als „blutigen Anfänger“ bezeichnet. Da ich ihn bereits zwei Jahre zuvor regelmäßig bei Fachvorträgen in einem anderen Community-Treffpunkt gesehen hatte, erscheint seine Selbsteinschätzung in dieser Situation wie eine Bezugnahme auf A1s Versiertheit (vgl. zu Selbst- und Fremdeinschätzungen den folgenden Abschnitt).

Zu der Betrachtung von technischem Support als Selbstinszenierung vor Publikum bleibt festzuhalten: In diversen Situationen wird technische Unterstützung gegeben, doch gleichzeitig findet eine Inszenierung persönlicher Kompetenzen durch die Helfer_innen statt. Das müssen nicht unbedingt technische Kompetenzen sein, sondern A1s geschmacksbetontes Urteil über ein Dateisystem erscheint eher als soziale Kompetenz, als „menschlicher“ Ratschlag. Der Eindruck, dass es sich um eine Inszenierung, eine persönliche „Show“, handelt, kann nur entstehen, wenn auch ein Publikum da ist, das die

³³Schon die Verwendung textbasierter Programme selbst wirkt auf Neulinge oft als Zeichen von Expertise. Darauf wird in Abschnitt 6.1 genauer eingegangen.

vorgeführten Kompetenzen wertschätzen kann, das also eine bestimmte *illusio* bzw. die zugrundeliegenden Wahrnehmungsstrukturen teilt (vgl. Bourdieu 1998b: 171). Das Publikum besteht oft nicht nur aus den Fragenden, sondern auch aus anderen Anwesenden, die das zu lösende Problem selbst nicht lösen könnten und zuhören, um etwas zu lernen.³⁴ Sie teilen ein Interesse an der Software und oft eine Faszination von technischen Möglichkeiten oder von der Versiertheit der Helfer_innen, unabhängig davon, ob alle die Lösungswege inhaltlich mitverfolgen können. Die Inszenierung von technischem Interesse oder technischer Kompetenz ist auch in anderen Kontexten bereits untersucht worden: Ursula Offenberger und Julia Nentwich haben z. B. vergleichbare Beobachtungen bei Paaren gemacht, die sich vor dem Kauf einer Heizungsanlage beraten lassen (vgl. Offenberger/Nentwich 2009).

In diesen Selbstdarstellungen akkumulieren die Helfer_innen soziales und symbolisches Kapital. Statt direktem „Applaus“ bekommen sie Supportanfragen, ihnen werden verschiedene Kompetenzen zugeschrieben, ihre Spezialisierungen sprechen sich in der Gruppe herum. Hier geht es nicht nur um technische Kompetenzen, sondern auch darum, wie sie erklären und wie sie dabei auftreten. Der Ruf, den sie sich erarbeiten, kann teilweise Effekte über die LUG hinaus zeitigen und sich in ökonomisches Kapital umwandeln lassen.

Er schien gar nicht darüber hinweg zu kommen, dass all diese hochkompetenten Leute ihm halfen. Er sagte so etwas wie: „Das sind ja Kompetenzen hier“, fragte A2, ob man ihn anheuern könne und ließ sich seine Visitenkarte geben“ (LUG 2008i).

A1 möchte eine Testmail an sich verschicken und beginnt, seine Adresse einzugeben. A54 fragt, ob die Adresse dann gespeichert würde, falls sie Kontakt aufnehmen möchte. A1 bricht also ab und trägt sich zunächst in das Adressbuch ein, mit seinem Namen, hinter den er in Klammern ‚der Linuxtyp‘ setzt. Seine dienstliche Telefonnummer trägt er auch ein (LUG 2009h).

Dies trifft freilich nicht nur auf FLOSS-Communities zu: Ronald Hitzler und Michaela Pfadenhauer untersuchen z. B. die oft „unsichtbaren Bildungsprogramme“ von Jugendszenen und zeigen auf, dass sehr viele Kompetenzen, die durch ein Engagement in einer

³⁴Hierzu muss angemerkt werden, dass ich durch meine Beobachterinnenrolle in sehr vielen Situationen ganz explizit als interessiertes Publikum erkennbar war. Es ist schwer dingfest zu machen, wie ich die Situationen dadurch beeinflusst habe, ob und wie sehr ich Inszenierungen provoziert habe. Es gibt jedoch auch hier ganz verschiedene Praktiken des Umgangs: Nicht alle, die wissen, dass jemand zuhört, beginnen, sich in Szene zu setzen.

Jugendszene als informellem Kontext erworben wurden, auch außerhalb der Szene berufspraktisch relevant werden können (vgl. Hitzler/Pfadenhauer 2006: 242).

Manche inszenieren sich so, dass sie an der Grenze dazu sind, sich auf Kosten der Person, der sie helfen, Geltung zu verschaffen, wie z. B. A87, der auffälligen Blickkontakt zu seinem Publikum sucht, während er A86 „trainiert“. Spätestens dort ist die Selbstinszenierung auch eine Distinktionspraxis, die das Wissen oder die Anliegen der Hilfesuchenden abwertet. Dazu verhalten sich teilweise auch die Zuschauer_innen, indem sie (wie A6) ihr eigenes Wissen gegenüber dem vorgeführten Wissen unter den Scheffel stellen oder indem sie der Selbstinszenierung einer Person Grenzen setzen. Die Inszenierung der einen bringt bei anderen u. U. ein Gefühl relativer Inkompetenz hervor. Und nicht zuletzt riskieren diejenigen, die ihr Können vorführen etwas, indem sie Lösungswege präsentieren. Sie fordern implizit das Publikum zum Transfer von Anerkennung auf, riskieren aber, dass ihre Defizite sichtbar werden, z. B. der Weg bestimmten Bewertungskriterien nicht entspricht.

5.2.5. Maßstäbe für Wissen im informellen Lernkontext

Die Praktiken des technischen Supports werden von unterschiedlichen Vorannahmen über das Wissen von Fragenden geprägt. Zudem wird durch stilistische Distinktion oft implizit mit Wissen und Wissensmaßstäben gespielt. Diese Maßstäbe stehen im Zentrum dieses Abschnitts. Die eben genannte Selbsteinschätzung von A6, der sich möglicherweise angesichts der vorgeführten Problemlösungskompetenz anderer als „blutigen Anfänger“ einstuft, habe ich im Protokoll überrascht dokumentiert, weil ich ihn nach meiner Beobachtung nicht so einschätzte. Im informellen Lernkontext Linux User Group ist die Einschätzung technischer Expertise sehr subjektiv und situationsabhängig. Die Frage nach der Kompetenz anderer stellte ich mir, nachdem mir aufgefallen war, dass sich manchmal Personen in Szene setzen, die *keine* funktionierende Lösung für ein Problem anzubieten haben. Zu Beginn meiner Beobachtungen kam es mir noch so vor, als wüssten diejenigen, die sich zu einem Problem äußern, immer auch sehr gut darüber Bescheid. Ich versuchte deshalb festzustellen, wer „tatsächlich“ Ahnung hat und wer sich „nur aufspielt“. Es hängt stark vom Wissensstand der anderen jeweils Anwesenden ab, ob eine Inszenierung ohne fundiertes Wissen zum Thema funktioniert. Es gibt in sehr vielen sozialen Zusammenhängen Personen, die sich gerne inszenieren. Dennoch gibt es eine Besonderheit im

Computerbereich, nämlich dass das Themenspektrum so breit ist und so viele Möglichkeiten zu Spezialisierung bietet. Die Chance auf ein nicht so informiertes Publikum zu stoßen, ist deshalb relativ hoch. Zunächst konnte ich selbst nicht einschätzen, ob jemand „tatsächliche Expertise“ in Szene setzt oder nicht; ich ging einfach unhinterfragt davon aus. Aus der Perspektive musste jemand, der etwas fragte, was „selbst ich“ beantworten konnte, ein Neuling sein.³⁵ Dahinter stand die Vorstellung von einem linearen Lernprozess, in dem viele Linux-Nutzer_innen sich die verschiedenen Themen in der gleichen oder in einer ähnlichen Reihenfolge aneignen wie ich selbst. Bei den Einschätzungen des Wissens anderer nahm ich damit jeweils meinen Wissensstand und meinen Lernprozess als Maßstab.³⁶ Für die Vorannahme, dass das ein übertragbarer Maßstab ist, gibt es angesichts der Informalität und der Individualisierung des ganzen Lernprozesses jedoch keine Grundlage. Mit der Zeit, mit wachsendem Wissen und Gespür für die Habitus relativierte sich meine Leichtgläubigkeit³⁷, weil mir Fehler auffielen und weil meine eigenen Lösungsvorschläge mehrfach ignoriert wurden. Dies spiegelt sich in Teilen meiner Beobachtungsprotokolle deutlich wider: Ich habe oft vermerkt, ob ich die an einer Situation Beteiligten inhaltlich für kompetent halte oder nicht. Erst in spät angefertigten Protokollen habe ich derartige Bemerkungen als unreflektierte Klassifizierungen eher zensiert; sie sollen hier jedoch thematisiert werden. Ob andere in derselben Weise Wissensstände einzuordnen versuchen, ist kaum beobachtbar und wurde in der LUG oft nicht so explizit ausgesprochen wie in der Selbstbeurteilung von A6. Lediglich einmal äußerte A77 mir gegenüber im Zweiergespräch deutlich, dass er einen anderen an den LUG-Treffen teilnehmenden Systemadministrator für inkompetent halte. Ein rauer Umgang mit unterschiedlichen Wissensständen wird ansonsten eher in Online-Foren deutlich, während in der LUG-Kommunikation wesentlich impliziter mit Annahmen über Wissen oder Unwissenheit umgegangen wird. Aus diesem Grund zeige ich das Klassifizieren und Klassifiziertwerden kritisch anhand meiner eigenen Kommentare. Sie belegen einerseits, dass ich mir ein Bild vom Wissensstand anderer Personen zu machen versuche, und andererseits, dass

³⁵ „Einer fragt A23 ‚umount muss man mit *sudo* machen?‘ A23: ‚Ich glaube ja.‘ Es scheinen also auch Dummy-Fragen erlaubt zu sein“ (LUG 2007a). Dass ich eine Frage als „Dummy-Frage“ klassifizierte, zeigt, wie stark ich von einem linearen Lernprozess ausging. Dass mir auffällt, dass eine solche Frage „erlaubt“ ist, ist von meinen Beobachtungen in Online-Foren geprägt, wo Fragen von Anfänger_innen sehr oft abqualifiziert und sanktioniert werden.

³⁶ „Schwer einzuschätzen, wie fit A44 ist. Mit bestimmten Sachen scheint er sich gut auszukennen, aber wie man zwei Rechner miteinander verbindet, weiß er nicht...“ (LUG 2008j).

³⁷ „Ich werde den Eindruck nicht los, dass er in Momenten, in denen er nicht weiß, wie er präzise antworten soll, ins Gelaber ausweicht: So könne man das schon mal nicht sagen, weil ... Also Wissen abzuspulen, dass eher nur so ganz grob in die Richtung geht“ (LUG 2008c).

ich offenbar von anderen teilweise als inkompetente Helferin eingeordnet werde.

Er stellt mir gleich am Anfang eine so dumme Frage, dass ich es kaum glauben wollte, dass er so schwierige Vorträge im Szenetreffpunkt Y3 gehört hatte, von denen ich oft nichts verstanden habe. Er gehörte dort ja schon zum Stammpublikum. Nämlich: „Silke, wo finde ich bei gmx die Zugangsdaten für *pop3* und *smp*?“ Hier verwechselte er die Reihenfolge der Buchstaben und meint *sntp* (LUG 2008i).

Ich klassifiziere seine Frage hier als „dumm“: Sie ist nicht linuxbezogen, dafür abhängig von einem speziellen Diensteanbieter. Zudem steht sie für mich in einem Kontrast dazu, dass ich A6 als Stammgast von Fachvorträgen beobachtet hatte. Ich unterstelle ihm damit, dass er mir die Frage gestellt hat, weil er das selbst nicht herausfand.³⁸ In einer später dokumentierten Szene bezeichnet jemand anders A6 als einen Linux-Experten, woraufhin ich wiederum Überraschung dokumentiere, da ich diese Einschätzung zu dem Zeitpunkt nicht mehr teile (vgl. LUG 2009g). Meine Einschätzung des Wissens anderer verschiebt sich nicht nur, weil ich sie inzwischen öfter beobachtet habe, sondern weil ich selbst dazugelernt habe und sich damit mein linear angelegter Maßstab mitverschoben hat. Wer ein Jahr zuvor nach diesem Maßstab „Expertenstatus“ für mich hatte, entspricht dem inzwischen nicht mehr unbedingt. Umgekehrt werde auch ich von anderen klassifiziert. In der folgenden Szene geht es — so meine Interpretation — darum, bei der Person, die um Hilfe bittet, Glaubwürdigkeit als kompetenter Helfer zu erlangen.

Schließlich dreht sich ein Mann, auf dessen Konsole „A76“ steht, um. Er sitzt direkt daneben, mit seinem Laptop, bisher war er uns abgewandt oder nur halb zugewandt. Er fragt, woran das denn liege, dass er im Programm *gnome-commander* nicht in den Admin-Betrieb wechseln kann, obwohl die Funktion da eigentlich eingebaut sei. Er demonstriert uns, dass er nach dem Passwort gefragt wird, dann aber nicht root sei. A1, A5 und ich steigen darauf ein. Eine_r der beiden fragt, welche → *Distribution* A76 installiert habe: *Debian Sid*.³⁹ Ich sage, dass sei instabile Software, da sei es nicht verwunderlich, wenn noch nicht alles funktioniere. Ich rate ihm, *Debian Stable* zu installieren. Er sagt, das habe nicht funktioniert, da habe er die Festplatte nicht verschlüsseln können und steigt dann ein, was er sich da für einen → *Kernel* kompiliert habe.

³⁸Ob es andere Gründe für Fragen geben kann, z. B. mit Umsitzenden ins Gespräch zu kommen, muss im Bereich der Spekulation verbleiben. Ich bezog solche Möglichkeiten bei meiner Einordnung jedoch gar nicht mit ein.

³⁹Die *Distribution Debian* liegt zu jedem Zeitpunkt in verschiedenen Versionen vor, die sich in unterschiedlichen Entwicklungsstadien befindet: *Stable* ist die als stabil erklärte Veröffentlichung, *testing* ist auf dem Weg, die nächste stabile Veröffentlichung zu werden, *sid* („*still in development*“) gilt als instabile Software. Viele verwenden sie dennoch, da die Programmversionen aktueller sind.

Kurz darauf kommt der Mann von vorhin (...) vorbei (A77). Er (...) will sich verabschieden, wird von A76 aber gleich einbezogen. A77 reagiert mit der Frage, wozu A76 denn den *gnome-commander* nutze und nicht die Shell. Ich höre A1 und A5 reagieren: A1 schnauft auf, als rege ihn die Antwort auf. A5 sagt, das sei keine Lösung des Problems. A77 fragt dann recht zielstrebig, was auf dem Notebook laufe und sagt, *Sid* ist nicht „stable“, er solle *Etch* (die stabile Debian-Version) installieren. Ich denke, das kommt mir ja irgendwie bekannt vor. Da steigt A76 aber darauf ein und beginnt Nachfragen zu stellen (LUG 2008c).

Ich äußere hier einen Erklärungsansatz für das Problem. A76 stuft dies als nicht hilfreich ein („das habe nicht funktioniert“). Er schweift von der eigentlichen Frage ab, indem er über Festplattenverschlüsselung und selbst kompilierte Kernel⁴⁰ spricht, bringt also seine eigene technische Kompetenz ins Spiel. Er richtet seine Frage an eine weitere Person, die das grafische Programm, mit dem es ein Problem gibt, abqualifiziert und zum textbasierten Arbeiten rät. Die beiden anderen Anwesenden signalisieren, dass sie seine wertende Äußerung nicht akzeptieren. Daraufhin stellt A77 die gleiche Rückfrage wie sie zu Beginn und schlägt den gleichen Lösungsansatz vor wie ich zuvor. A76 interessiert sich nun dafür. Die Szene zeigt, dass sich sowohl A76 als auch A77 abschweifend mit ihrem eigenem Wissen zu anderen Themen inszenieren, bevor sie sich mit dem fraglichen Problem auseinandersetzen. Ob die beiden sich bereits kannten und A77s Glaubwürdigkeit darauf oder auf seiner Inszenierung basierte, geht nicht aus dem Protokoll hervor, sondern lediglich, dass A77 und ich mit dem gleichen Hinweis nicht die gleiche Glaubwürdigkeit erfahren.

Das Klassifizieren des vermuteten Wissens von Personen erfolgt auf einer unbekanntem Wissensskala, sehr subjektiv, situativ und relativ zu anderen Anwesenden. Der *explizite* Umgang mit unterschiedlichen Wissensständen, Interessen und Spezialgebieten ist in der LUG ein anderer:

A45 fragt, was ich in der IT-Abteilung mache, ich erzähle von der Mischung aus First-Level-Support und Zuarbeiten für Softwareentwickler, z. B. Latex-Vorlagen. Das macht A45 und A1 aufmerksam, Latex sei ja schon recht anspruchsvoll. (Ich frage mich, ob die Bemerkung heißt, dass sie mir etwas Anspruchsvolles nicht zugetraut hätten.) Es geht dann um Latex, A1 sagt: „Dann weiß ich ja, wen ich da mal fragen kann.“ (Hier fällt mir ein, dass das oft vorkommt: Ich wurde schon bei Y3, aber auch

⁴⁰Um Linux zu nutzen, ist es nicht nötig, sich selbst einen Kernel zu kompilieren. Man kann das aber tun, um den eigenen Kernel genau auf die vorhandene Hardware zuzuschneiden, unnötige Module bzw. Treiber hinauszwerfen und so zu mehr Geschwindigkeit zu kommen.

bei Y2 gefragt, was „man mich denn fragen könnte“, also welches meine Spezialgebiete oder Steckenpferde seien.) Ich freue mich auch ein bisschen, weil ich das als Anerkennung werte (LUG 2008i).

Dieser Umgang hebt nicht die Wissenslücken hervor, sondern die persönlichen Interessen und die Themen, mit denen bestimmte Personen sich besonders gut auskennen. In einer diversifizierten Softwarelandschaft, die niemand komplett kennen kann, werden so Einzelne zu „Expert_innen“ für Themenbereiche, die mit ihrem Wissen der Community weiterhelfen können. Im Großen zeigt sich dies daran, dass diese „Expert_innen“ auf Community-Events oder in der LUG Vorträge halten, daraufhin erneute Anfragen erhalten. Sie werden oft dann als „Gurus“ für ihre Themen bezeichnet.

Für technisches Wissen und Linux-Expertise gibt es in LUGs als informellen Zusammenhängen keine allgemein gültigen Maßstäbe. Helfer_innen hantieren zwar mit impliziten Annahmen über das Wissen der Fragenden, die wie oben gezeigt mit Zuschreibungen von Geschlechterrollen zu tun haben. Diese Annahmen und Einschätzungen erfolgen jedoch situativ anhand persönlicher Wissensskalen. Meine Skala war an meinem eigenen Wissensstand und meinen Erinnerungen an meinen Lernprozess orientiert. Diesen Prozess setzte ich lange als einen linearen und übertragbaren voraus, was nicht nur einer inhaltlicher Hinterfragung bedarf: In LUGs erwerben die Involvierten ein Wissen, das sie interessiert oder durch Fragen anderer an sie herangetragen wird. Selbst Grundlagenwissen über das Betriebssystem werden nie in derselben Reihenfolge erlernt, da es kein Curriculum dafür gibt. In der Praxis des technischen Supports in LUGs spielen die wenigen existierenden formalen Qualifikationen keine große Rolle: Ob jemand ein LPI-Zertifikat erworben hat oder eine Stelle als Systemadministrator_in bekleidet, hat nicht immer etwas mit der situativen praktischen Problemlösungskompetenz zu tun, die in der LUG einer der wichtigsten Einsätze ist. Mein Anliegen einer impliziten persönlichen Wissensskala an die Beobachteten muss jedoch auch als intransparente, aber ordnende Praxis meinerseits benannt werden. Was „Expertenwissen“ ist, schwimmt damit in LUGs: Alle, die sich — in welchem Umfang auch immer — in ein Thema einarbeiten, können damit jemanden unterstützen, was auch so praktiziert wird, mit der Möglichkeit, Themen zu delegieren. Es werden durchaus Unterschiede zwischen Lösungswegen gemacht, wenn sie nach verschiedenen Kriterien bewertet werden. Sich diese Kriterien anzueignen ist dabei Teil des Lernprozesses. Gleichzeitig ist das Fehlen klarer Kriterien auch die Grundlage dafür, „Expertenwissen“ für die Distinktion zu nutzen.

5.2.6. Ausprobieren: Zentrale Lernpraxis und Problemlösungsstrategie

Bei der Lösung von Problemen und dem Lernen allgemein kommen viele Strategien zum Einsatz: Neben dem (sprachlichen) Formulieren von Problemen recherchieren die LUG-Mitglieder nach Informationen (v.a. im Internet, aber auch in Zeitschriften, Büchern oder Softwareanleitungen, den so genannten → *man pages*. Sie machen Notizen auf Papier, in Dateien auf ihren Rechnern oder in öffentliche Wikis, sie zeichnen Schemata zur Veranschaulichung, lassen sich Befehle diktieren, sie erarbeiten gemeinsame Konfigurationen oder greifen auf Vorlagen zurück, die sie dann anpassen. Eine besonders auffällige Lern- bzw. Problemlösungsstrategie ist jedoch das Ausprobieren. Huysman / Lin haben „trial and error“ bereits als eine wichtige Problemlösungsstrategie benannt. Meine Empirie untermauert diesen Befund. Dass die LUG-Mitglieder so viel ausprobieren, erklären die Autorinnen damit, dass die Probleme so lokal sind, also stark von der speziellen Hard- und Software abhängen. Helfer_innen könnten also aus Erfahrungen („tacit knowledge“) schöpfen, müssten die Problemursachen aber auch immer wieder erraten (vgl. Huysman/Lin 2005: 62 ff.).

Im bisher Geschriebenen kam das Experimentieren mit Software und Problemlösungswegen schon an vielen Stellen vor. Rein quantitativ betrachtet, habe ich beim Kodieren den Code „Ausprobieren“ so oft vergeben, dass Praktiken des Ausprobierens in FLOSS-Communities als zentral bezeichnet werden müssen. Claus Tully beschäftigt sich mit der Aneignung von computerbezogenen Kompetenzen und macht einen deutlichen Unterschied zwischen dem Ausprobieren mit Computerprogrammen „nach dem einfachen Motto ‚Probieren geht über’s Studieren‘“ und fachlicher Kompetenz (Tully 1994: 24). Bei der Untersuchung von Lernprozessen in FLOSS-Communities schwimmt dieser Unterschied: Ausprobieren bedeutet nicht unbedingt, dass Personen die fachliche Kompetenz fehlt. In diesem Abschnitt wird gezeigt, wie Linux-Nutzer_innen ausprobieren, unabhängig von der Menge an Fachwissen, über die sie verfügen.

In der beobachteten IT-Abteilung schreibt ein Administrator ein → *Shellskript*, also ein Programm, in dem verschiedene textbasierte Programme zusammengesetzt werden, um Aufgaben zu automatisieren.

Gestern habe ich A27 einen Moment lang beim Schreiben eines Skriptes zugesehen. Es sollte für die neue Foto-Upload-Plattform sein, die hochgeladenen und in Ordner verteilten Fotos aussagekräftig umbenennen und verkleinern. Mir ist aufgefallen, dass

A27 immer erst einzelne Befehle ausführt, um zu sehen, ob sie zum gewünschten Ergebnis führen, z. B. die Zeit anzeigen, zu der das Originalfoto gemacht wurde statt der Zeit, zu der es auf den Server hochgeladen wurde. Dieses Ausprobieren von Befehlen geht sehr schnell. Er tippt sehr schnell und hat das meiste im Kopf. Er hat (immer beim Arbeiten) (...) vier Shells offen, zwischen denen er hin- und herschaltet. Manchmal öffnet er zu einem Befehl, den er braucht, in einer freien Shell die → *man page*. Es sieht mir so aus, als ob er viel und schnell herumprobiert, statt einmal gründlich den Befehl zu durchdenken: Er tippt etwas ein, schaut die Ausgabe an, ruft den letzten Befehl nochmals auf, ändert etwas und sieht sich erneut an, was dabei herauskommt. Das tut er solange, bis das gewünschte Ergebnis zurückgegeben wird. Er scheint dabei Flüchtigkeitsfehler zu machen und daran herumzubessern, dann mal etwas in der *man page* nachzuschauen, aber immer erstmal drauflos (Fir 2009).

Der Verfasser des Skriptes wählt nach und nach die Befehle aus, die in das Skript eingefügt werden sollen und bringt sie zunächst einzeln in eine Form, die das gewünschte Ergebnis bringt (z. B. die Sortierung der Fotos nach einem bestimmten Zeitstempel). A27 hat dazu vier → *X-Terminals* nebeneinander geöffnet, in einem bearbeitet er das Skript, in anderen probiert er die einzufügenden Teile aus oder ruft die dazugehörigen Handbuchseiten auf. Durch die Zerlegung der Programmieraufgabe in Schritte, die einzeln erledigt und zusammengesetzt werden, gibt er dem Ganzen eine Systematik. Das schnelle Eingeben von Befehlen, das testweise Ansehen der Ausgaben, das Nachlesen im Handbuch und das darauf folgende erneute Eingeben eines abgewandelten Befehls wirkt gleichzeitig auch unsystematisch, tastend. Die Ausgaben eines Befehles müssen jedoch in einem ganz bestimmten Format vorliegen, in dem sie im weiteren Verlauf des Skriptes weiterverarbeitet werden können. Da A27 dieses genaue Format nicht jeweils auswendig vorhersagen kann, vergewissert er sich jedes Schrittes.

In der zweiten Szene nutzt eine Softwareentwicklerin, die zu Gast bei einem LUG-Treffen ist, eine von LUG-Mitgliedern selbst entwickelte Software. Dabei stößt sie zufällig auf ein Problem, das sie dann durch Tests eingrenzt.

Kurz darauf sitzt A42 wieder vor dem Rechner und klickt herum, sie ruft verschiedene Optionen auf, indem sie mit der Maus über die kreisförmig angeordneten Icons fährt und doppelklickt. Plötzlich stürzt die Software ab. Eine Meldung erscheint mit der Frage, ob sie Informationen speichern möchte. A43 [einer der Entwickler dieser Software] steht daneben und (...) scrollt durch die Meldung, sagt aber, er müsse sich das

in Ruhe durchlesen. A42 startet die Software dann gleich noch einmal und reproduziert den Fehler: Sie sagt, sie habe auf diese Option geklickt und wieder abgebrochen, dann auf diese Option und wieder abgebrochen usw. bei einer bestimmten Option sei es abgestürzt. Es passiert wieder. A43 reagiert bestürzt. A42 sagt, sie habe drei Jahre lang als Software-Testerin gearbeitet, sie ziehe so etwas an. Sie braucht nach meiner Schätzung (und auch zu meiner und A43s Begeisterung) vielleicht höchstens zehn Minuten, um herauszufinden, was das Problem ist. Sie geht immer wieder die Klicks durch, wandelt die Reihenfolge ab, denkt laut mit, was sie ausschließt. Sie kommt darauf, dass es nur beim Klick auf die erste Option in dem Kreis zum Absturz kommt, die beim Programmstart immer dem Betrachter / der Betracherin zugewandt ist. Sie wirft dann die Frage auf, ob es mit den Mausbewegungen zu tun hat und probiert herum, wieder kommt sie ganz schnell darauf, dass es nicht funktioniert, wenn sie das Rad, auf dem die Optionen kreisförmig angeordnet sind, mit der Maus etwas zu weit zieht, dann zurückzieht und daraufklickt. Sie probiert das in beide Richtungen und grenzt das Problem so weiter ein. Sie sagt A43, der zwischendurch mit A2 auf ein anderes Problem geguckt hat, das Problem auf den Kopf zu: An dem Punkt, an dem das Rad in die eine Richtung über das Ende hinausgedreht werde, stürze das Programm ab. A43 ist erfreut, er bedankt sich zweimal dafür, dass sie das herausgefunden hat und sagt, er wisse schon, was er ändern müsse (LUG 2009e).

Hier wird deutlich, dass Ausprobieren nicht nur bedeutet, mit einer Software nicht umgehen zu können und hilfeschend herumzuklicken. Ausprobieren kann auch sehr voraussetzungsvoll sein, z. B. beim Programmieren oder Testen von Software. So gehört es auch zum Softwareentwickeln dazu, entsprechende Tests zu schreiben, die das Funktionieren von Zwischenschritten abfragen (vgl. Fir 2008a; siehe auch Strübing 1992: 66 ff.).⁴¹

In Gesprächen erfuhr ich auch, dass Einzelne an den LUG-Treffen kritisieren, dass dort sehr viel Energie in die Neuinstallation und Konfiguration von → *Distributionen* gesteckt wird. Auch Christopher Kelty traf bei seiner Ethnografie über die kulturelle Bedeutung von freier Software auf Menschen, die er als „strangely obsessed with configuring their Linux distributions“ bezeichnet (Kelty 2008b: 38). Dabei geht es nicht um eine Kritik daran, dass

⁴¹Die Komplexität des Testens von Software haben John Rooksby, Mark Rouncefield und Ian Sommerville ethnografisch untersucht. Sie betonen, dass die Effizienz von Tests nicht allein eine als komplexe *technische* Aufgabe betrachtet werden darf. „Testing no longer simply focuses on the technology alone but on socio-technical issues such as acceptability, usability and fitness for purpose“ (Rooksby et al. 2009: 561). Sie charakterisieren das Softwaretesten im beruflichen Umfeld als eine Aufgabe, zu der ein hoher Grad an Organisiertheit und auch eine Portion Spekulation über die zukünftige Nutzung nötig seien. Diese Arbeit finde unter ständigem Zeitmangel statt (vgl. ebd. 573 ff.).

Neulinge eingeführt werden, sondern daran, dass manche Menschen in der LUG auf ihren *eigenen* Rechnern immer wieder andere oder neuere Distributionen installieren und diese konfigurieren und auch optisch an ihren Geschmack anpassen. Über die Zeit meiner Beobachtungen hatten zwar diverse regelmäßig Anwesende tatsächlich neue Rechner oder solche, die sie bei einem ganz bestimmten Gebrauchthändler erstanden hatten. Für die Einrichtung nutzten sie die Zeit in der User Group. Abgesehen davon gibt es aber auch Etliche, die einfach verschiedene Distributionen installieren, um sie zu vergleichen, mit ihnen herumzuspielen, sich Meinungen über sie zu bilden. Verschiedene Personen haben mehrere Distributionen parallel auf demselben Rechner installiert, andere sind im Beobachtungszeitraum von einer Distribution auf eine andere umgestiegen.⁴² Die „extremste Spielerei“ mit Distributionen habe ich bei A6 beobachtet:

Als ich zurück zu A6 komme, sehe ich auf seinem Bildschirm den blauen Installer [Installationsassistenten]. Ich frage A51, was A6 da mache. Er sagt, A6 installiere → *Ubuntu* Linux, er sei mit dem → *Kubuntu* nicht zufrieden gewesen. (Ich wundere mich, warum er nicht → *Gnome* nachinstalliert, statt gleich alles neu aufzusetzen.) A6 fragt mich, womit ich ‚gerade‘ arbeite. Ich sage → *Debian*, und dass ich das immer hätte und nicht wechseln würde. Er setzt mir daraufhin einen Unterschied zwischen den Network-Managern⁴³ in *Debian* und *Ubuntu* auseinander. (...) Seine Installation ist fertig. (...) Ich frage ihn, was er jetzt mit seinen Daten gemacht habe, wenn er mal eben schnell neu installiert. Er sagt, da seien keine drauf, dieser Rechner sei nur zum Spielen da. (LUG 2009i).

A6 geht und ich sitze mit seinem Rechner, dem T61, da. Ich gucke mich darauf um und stelle fest, dass überhaupt keine Daten darauf sind, nur ein gespeichertes Passwort für einen DSL-Zugang ist in der Passwörter-Verwaltung. Sonst ist kein E-Mail-Programm eingerichtet, keine Daten außer dem Hintergrundbild (einem dunklen Bild mit der Aufschrift *Ubuntu 9.04*) im persönlichen Ordner. Als er später zurückkommt, spreche ich ihn darauf an, er sagt, die Daten habe er auf einem USB-Stick. Er erinnert mich daran, dass ich auch sein root-Passwort wisse. Ich erinnere mich tatsächlich (LUG 2009e).

Der Besitzer dieses Notebooks hat verschiedene Rechner. Den T61 nutzt er zum Herumspielen, für Testinstallationen. Konsequenterweise speichert er darauf keine Daten, so

⁴² „Aus dem Augenwinkel sehe ich, dass A80 an seinen Rechner kommt (...). Er hat *Sidux* laufen (nicht mehr *Debian*) (LUG 2009j).

⁴³ Ein Network-Manager ist ein Programm, über das man die Netzwerkverbindungen des Rechners einstellen kann, z. B. zwischen verschiedenen WLAN-Netzen auswählen kann.

dass in der LUG auch andere den Rechner zwischendurch nutzen dürfen und etlichen Personen sein root-Passwort bekannt ist. A114 hat seine Kritik an diesem Zeitvertreib von LUG-Mitgliedern einmal deutlich geäußert: Er lebt seit Jahren von *Hartz IV* und würde gerne sein Wissen über Linux für einen beruflichen Quereinstieg nutzen. Er hätte sich gewünscht, in der LUG dafür Mitstreiter_innen zu finden, um beispielsweise eine Genossenschaft zu gründen. Doch dort seien viele mit immer den gleichen Tätigkeiten beschäftigt, zu denen er nicht erforderliche Neuinstallationen und übermäßige Konfigurationsarbeit zählt (vgl. Inf 2010). Auch ein anderer Gesprächspartner hat sich in einem Gespräch derart geäußert, dass es nicht soviel bringe, sich so viel mit spielerischen Neuinstallationen zu beschäftigen. Für A6 kann dies Spiel, Experiment, eine Form des autodidaktischen Lernens oder alles davon sein.

Ausprobieren ist in LUGs eine zentrale Praxis. Sie spielt einerseits bei der Aneignung von Linux eine wichtige Rolle als Lernstrategie. Andererseits sind Tests und Experimente inhärenter Bestandteil einer situativen praktischen Problemlösungskompetenz. Im Ausprobieren spiegelt sich vieles wider, was die intensive Beschäftigung mit freier Software ausmacht: Die Software ist nicht immer perfekt, sondern wird angepasst, verbessert, selbst geschrieben. Diese Prozesse sind ergebnisoffen, es gibt meistens verschiedene technische Lösungswege für eine Aufgabe und viele mögliche Ursachen für ein Problem. Die systematische Zerlegung in Teilschritte und das Experimentieren mit Zwischenergebnissen strukturiert diese Lösungswege. Ausprobieren von Fachkompetenz zu unterscheiden, wie Claus Tully es tut, ist hier nicht trefflich, da es auf allen Wissensständen und in ganz verschiedenen Arbeitsabläufen praktiziert wird. In Bezug auf Differenzen müssen jedoch die Bedingungen für *erfolgreiches* Ausprobieren beleuchtet werden: Es erfordert nicht nur ein unter Umständen großes Zeitbudget und viel Geduld, ein Problem zu ergründen, es einzugrenzen und zu lösen, über das nicht alle Linux-Nutzer_innen verfügen. Das Experimentieren hat auch um so größere Erfolgchancen, je mehr Wissen, Erfahrung und praktische Problemlösungskompetenz jemand mitbringt. Wer um die Spezifika von Hardware weiß, wer weiß, wo Konfigurationsdateien gespeichert sind, welche Argumente einem Befehl *wahrscheinlich* mitgegeben werden müssen, wer *man pages* versteht, hat mehr Möglichkeiten, ein Problem einzugrenzen. Die Kompetenzen, erfolgreiche Tests systematisch durchzuführen und über das Durchprobieren verschiedener Möglichkeiten ein Problem einzugrenzen, sind Bestandteile der situativen praktischen Problemlösungskompetenz. All diese Kompetenzen müssen als Kontinuum gedacht werden; in Bezug auf freie Software ist es nicht sinnvoll, einen klaren Kontrast zwischen „Wissen“ und „Ausprobieren“

aufzumachen. Das Wissen kommt vielmehr im Ausprobieren zum Ausdruck.

5.3. Empirische Zwischenergebnisse

In diesem Kapitel stand der Wissenstransfer in Linux User Groups im Mittelpunkt. Vor dem Hintergrund ihres Anliegens, Wissen über freie Software zu verbreiten, wurde genauer danach gefragt, über welche Praktiken wie welches Wissen vermittelt wird und ob und wie in diesen Prozessen gleichzeitig soziale Differenzierungen stattfinden, die abgrenzenden oder sogar ausgrenzenden Charakter haben. Die teilnehmende Beobachtung von technischem Support in Linux User Groups hat gezeigt, wie komplex die Situationen sind, in denen kulturelles Kapital in Form von technischem Know-how vermittelt wird. Das Setting ist sehr informell: Während manche mit konkreten Fragen zum Treffen kommen, sind viele regelmäßige Gäste auch da, um sich mit Gleichgesinnten zu treffen. Dann ergeben sich spontan Situationen, in denen sie zu Fragenden oder Helfer_innen werden; diese Rollen sind vorab mitnichten klar und wechseln von Situation zu Situation.⁴⁴ Diejenigen, die häufiger als andere die Helferrolle einnehmen, akkumulieren darüber soziales Kapital. Hilfesuchende suchen sich selbst die Personen, die ihnen bestimmte Fragen beantworten können, sie müssen dabei implizit den Grad an Hilfsbereitschaft ausloten, z. B. wie viel Zeit, Tiefe und Kontinuität sie erwarten können. Der genaue Rahmen einer solchen Situation wird in den seltensten Fällen explizit verabredet, so dass die Zusammenarbeit stark von den Helfer_innen bestimmt wird und ständig von Unterbrechungen durch neu Hinzukommende oder durch Raucherpausen bedroht ist, in denen andere Anwesende ihre Fragen an dieselben Helfer_innen herantragen. Die Helfer_innen müssen ihrerseits erst herausfinden, wie viel die Fragenden schon wissen. *Wie* sie helfen, scheint oft auf ihren impliziten Vorannahmen über das Wissen der Fragenden zu basieren — Vorannahmen, die von gesellschaftlicher Ordnung beeinflusst sind. Dies konnte ich vor allem an der Kategorie Geschlecht am Material zeigen. Der Horizont, der Fragenden durch bestimmte Hilfspraktiken eröffnet wird, hat Konsequenzen darauf, inwiefern sie eine Erweiterung ihres eigenen momentanen und künftigen Handlungsspielraums in Bezug auf Linux erfahren. Dabei zeichnen sich die Helfer_innen durch ein sehr unterschiedliches in-

⁴⁴Ruediger Glott, Andreas Meiszner und Sulayman Sowe bestätigen dies auch für die freie Softwareentwicklung: Die Mitglieder von Projekten entschieden sich selbst für Rollen und Verantwortungen, die sie übernehmen wollten. Dies verändere sich mit der Zeit und je nach Kontext, wodurch die Communities eine sehr lebhaft und volatile interne Struktur hätten (vgl. Glott et al. 2007: 68).

dividuelles Maß an Geduld, Zeitbudget, Wissensständen, Problemlösungskompetenzen, pädagogischem Gespür und Erfahrungen aus.

Was beim Support in der LUG zählt, ist, ob jemand es schafft, ein technisches Problem zu lösen oder zur Lösung beizutragen, kurz: Es geht um das erfolgreiche Machen. Dies bezeichne ich als *situative praktische Problemlösungskompetenz*. Diese Kompetenz ist *situativ*, weil ein Problem mit Software von vielen Faktoren abhängen kann, von der verwendeten Hardware, von der Distribution, von der Konfiguration uvm. Die jeweilige Ursache muss sehr „lokal“ in jeder Situation gefunden werden. Ich bezeichne die Problemlösungskompetenz als *praktisch*, weil sie stark auf praktischen Erfahrungen basiert, was Fehlersuche und -eingrenzung angeht. Es geht weniger darum, Lösungswege auswendig einzuschlagen, sondern es geht darum, sie am jeweiligen Problem finden zu können. Welche Schritte die Helfer_innen dazu gehen, erläutern sie oft nicht oder nur teilweise, sie folgen der praktischen Logik, die sie eingeübt und inkorporiert haben. Zur Problemlösungskompetenz gehört auch das Ausprobieren, das auf den verschiedensten Wissensständen und in den verschiedensten Arbeitsabläufen praktiziert wird. Je mehr situative praktische Problemlösungskompetenz die Nutzer_innen haben, desto erfolgreicher können sie auch ausprobieren. Die spezifischen Kompetenzen, die beim Support zum Einsatz kommen, sind Teil eines Habitus der Linux-Nutzer_innen, der unterschiedlich stark verinnerlicht ist.

Es ist ein Ziel der Arbeit, die Praktiken von FLOSS-Communities im Detail auf soziale Differenzierung hin zu untersuchen und aufzuzeigen, entlang welcher Kategorien Unterschiede hergestellt werden. Das auf der Mikroebene erhobene Datenmaterial mit makrosoziologischen Kategorien zu konfrontieren, stellt sich als schwierig heraus, da das Spektrum der beobachteten Praktiken so breit, das konkrete Machen im informellen Kontext so individuell und situationsabhängig ist.⁴⁵ Differenzierungsprozesse und stilistische Distinktion laufen in den Support-Praktiken sehr kleinteilig, situiert, oft mehrdeutig und komplex ab. Dies führt zunächst zu der These, dass nicht von *einem* Habitus gesprochen werden kann, sondern dass die Beteiligten vielmehr *Habitusfragmente* in unterschiedlichem Maße inkorporiert haben. Im Folgenden werde ich deshalb den Begriff Habitus-

⁴⁵Glott / Meiszner / Sowe arbeiten mit Typologien, die sich anhand meines Materials — wie gesehen — nur unter sehr starker Vereindeutigung überhaupt aufstellen lassen würden. Sie ordnen diejenigen, die kontinuierlich an freien Softwareprojekten mitarbeiten, z. B. nach ihren Motivationen in vier Typen ein (deren prozentuale Anteile sie auch angeben): „skill improvers“, „ideologists“, „software improvers“ und „recognition seekers“ (vgl. Glott et al. 2007: 21). Wenn ich auch „skill improvers“ und „recognition seekers“ in meinem Sample haben werde, so sind das aber möglicherweise dieselben Personen, keine trennscharfen, verschiedenen Typen.

fragmente weiterverwenden. Er soll ausdrücken, dass Habitus aus vielen verschiedenen Anteilen besteht, ohne jedoch zu unterstellen, dass man diese Fragmente wie Bruchstücke zu einem kohärenten Ganzen zusammenfügen könnte. Demnach können keine verallgemeinerbaren Ergebnisse formuliert werden, wohl aber detaillierte *mögliche* Kriterien, denen situativ eine soziale Bedeutung verliehen werden kann: Zunächst basiert der komplette Support auf Wissensunterschieden. Diese können jedoch unterschiedlich gehandhabt werden und die Beziehungen zwischen Beteiligten unterschiedlich strukturieren (vgl. Holzkamp 1995: 510 f., dazu ausführlicher Abschnitt 5.4). Was hier als Zwischenergebnis festgehalten werden kann, ist das spezifische Zusammenspiel von situativer praktischer Problemlösungskompetenz, ihrer Inszenierung vor Publikum, dem Glauben an Meritokratie und geschlechtlich konnotierten Vorannahmen über das Wissen von Anwesenden:

Die Problemlösungskompetenz steht im Zentrum des technischen Supports in der LUG. Distinktion entlang der Kategorien Klasse oder Bildung waren der teilnehmenden Beobachtung zwar nicht direkt zugänglich, da ich die beobachteten Praktiken nicht durch Befragungen an die sozioökonomischen Hintergründe der Personen rückgebunden habe. Die oben umrissene Problemlösungskompetenz lässt jedoch zumindest Aufschlüsse über diesen Hintergrund zu, da sie ein szenespezifisches inkorporiertes Kulturkapital ist. Viele Personen haben es über einen längeren Zeitraum und oft autodidaktisch bzw. in informellen Lernkontexten erworben. Dieses Kapital ist nicht nur sehr unterschiedlich ausgeprägt, sondern es wird auch unterschiedlich stark in Szene gesetzt: Die Maßstäbe für Expertise sind ständige Verhandlungssache, sie sind situations- und themenabhängig. Obwohl *alle* Beteiligten sich fortlaufend Wissen aneignen, und zwar in dem selbst geschaffenen informellen Kontext der LUG, inszenieren einige Helfer_innen ihre persönlichen Kompetenzen besonders wirkungsvoll vor den Fragenden und Umstehenden oder treten mit anderen in Konkurrenz um den besten Lösungsweg. Damit konstituieren sie sich situativ als Expert_innen. Diese Inszenierungen rund um Wissenserwerb bewirken zugleich, dass sich Community-Mitglieder nach Jahren der Beschäftigung mit Linux noch immer als „Newbies“ bezeichnen. Sie schreiben sich selbst damit eine unzureichende Problemlösungskompetenz zu, obwohl aufgrund der Informalität des ganzen Zusammenhangs kaum Maßstäbe für Expertise existieren. Es konstituiert das Feld FLOSS-Communities mit, dass viele Mitglieder sich wenig technisches Know-how zutrauen und einige sich viel zutrauen.⁴⁶ Hier muss auch bedacht werden, dass Helfer_innen unterschiedlich

⁴⁶Dies ist eine Analogie zu Bourdieus Feststellung über das politische Feld: Politische Kompetenz beruhe

glaubwürdig wirken: Meine Beobachtungen erlauben keine genauere Untersuchung von Praktiken, die zu Glaubwürdigkeit führen, sie zeigen lediglich, dass inhaltlich richtige Lösungsvorschläge allein nicht immer genügen. Bestimmte Personen verfügen zusätzlich zur situativen praktischen Problemlösungskompetenz über ein symbolisches Kapital, das die Fragenden diese Kompetenz auch glauben lässt. Dieses symbolische Kapital ist unter den Helfenden ungleich verteilt, ebenso, wie die Fähigkeit der Fragenden, die Glaubwürdigkeit zu bewerten, sie wahrzunehmen. Die wenigen formalisierten Maßstäbe, die es in FLOSS-Communities gibt (LPI, Hacking-Contest) sind in der alltäglichen Praxis der Gruppen nicht relevant: Anerkennung wird nicht über eine bestandene Zertifizierungsprüfung erworben, sondern über situative praktische Problemlösungskompetenz. Am eindeutigsten kann von Distinktion gesprochen werden, wenn Einzelne sich oder ihre Kompetenzen in Szene setzen und damit ihr Publikum zum Transfer von symbolischem Kapital auffordern.

Dies zeigt, dass in Praktiken des Supports eben nicht nur kulturelles Kapital transferiert wird, sondern in die umgekehrte Richtung auch symbolisches und soziales Kapital: Erfolgreiche Helfer_innen tragen Anerkennung, einen Ruf davon. In einigen Fällen akkumulieren sie soziales Kapital in Form von ausgetauschten Kontaktdaten, das möglicherweise zu einem späteren Zeitpunkt in ökonomisches Kapital umgewandelt werden kann. Dies widerspricht dem Ergebnis der Befragung von Lakhani/von Hippel (2003: 927): Laut ihnen ist eine Motivation von Helfer_innen die Erwartung von Gegenseitigkeit, was den Austausch technologiebezogener Leistungen betrifft. Wissen werde innerhalb der gesamten Szene ausgetauscht; für kulturelles Kapital werde zu einem anderen Zeitpunkt und eventuell von anderen Personen kulturelles Kapital zurückgegeben. Mit Ambitionen auf soziales oder symbolisches Kapital hielten die Befragten eher hinter dem Berg. Dadurch rückt aus dem Blick, dass viele Hilfesituationen *in sich* bereits ein Austausch von Kapitalformen sind.

Hier kommt die verbreitete Vorstellung von Meritokratie ins Spiel: Helfer_innen, die sich in diesem informellen Kontext darauf berufen, dass sie den Support freiwillig leisten und den Anspruch auf Einforderbarkeit ablehnen, können eine Hilfesituation jederzeit verlassen. Fragende bleiben in solchen Fällen mit ihren Schwierigkeiten und ihrer Bittstellersituation allein. Analog zu Programmierer_innen, die in FLOSS-Projekten sehr oft darüber entscheiden, ob sie gewünschte Funktionalitäten einbauen oder nicht (vgl. Jung 2006:

ebenfalls auf sozialen Bedingungen, zu denen schon das Gefühl gehört, ob Personen sich eine politische Meinung zutrauen oder nicht (vgl. die Einleitung von Philippe Fritsch in Bourdieu 2000: 10).

241 f.; Nafus et al. 2006: 31), können auch beim technischen Support die Helfer_innen abstecken, wie weit ihre Hilfe geht. Der Bezug auf Meritokratie kann damit ungleiche Möglichkeiten zur Gestaltung der Lehr-Lern-Beziehung eröffnen. Mit Bourdieu kann diese Meritokratie jedoch hinterfragt und als Tauschökonomie gelesen werden. Dann wird nämlich nicht nur kulturelles Kapital in Richtung der Fragenden transferiert, sondern die Anteile, die die Fragenden und das Publikum an den Verdiensten der Helfer_innen haben, rücken in den Blick: Sie sind entscheidend für den Kapitalzugewinn ihrer Helfer_innen, bedarf es doch derjenigen, die die Verdienste anerkennen.

Situativ konstituieren die Beteiligten im technischen Support verschiedene Männlichkeiten und Weiblichkeiten. Die Tendenz geht dahin, dass den Personen, die als Frauen wahrgenommen werden, zunächst weniger Know-how, weniger Interesse am Lösungsweg und am Aufbau eigener Problemlösungskompetenz unterstellt wird. Dies wurde oben auf gesellschaftliche Stereotype und ihr offensichtliches Gerinnen in Vorannahmen über Weiblichkeit und Selbsteinschätzungen in Bezug auf eigene Kompetenzen zurückgeführt. Helfende agieren Männlichkeit dann in Abhängigkeit von Vorannahmen unterschiedlich aus, indem sie entweder großzügig helfen und das Problem selbst in die Hand nehmen oder die fragende Person auf sich selbst zurückverweisen. Dabei wird autodidaktisches Lernen mit Männlichkeit assoziiert — dies bedeutet jedoch nicht, dass es ausschließlich von „Männern“ erwartet wird. Autodidaktisches Lernen wird durch die unterschiedlichen Supportpraktiken tendenziell jedoch als Norm in Linux-Communities hervorgebracht, mit der dann gebrochen wird, wenn dem Gegenüber Weiblichkeit zugeschrieben wird. Auch ist dies keine einseitige Praxis von männlich auftretenden Helfer_innen, sondern in den untersuchten Situationen traten die jeweiligen Gegenüber mit komplementären Selbstzuschreibungen auf, indem sie ihre Rechner abgaben, Lösungswege nicht mitverfolgten oder artikulierten, sie hätten wenig Wissen über Computer. Die Involvierten verhalten sich innerhalb einer Supportsituation nach impliziten Vorannahmen übereinander, die *auch* geschlechtlich konnotiert sind bzw. Geschlecht und Geschlechterunterschiede mit hervorbringen.

5.4. Theoretische Reflexion: Zur sozialen Selektivität informeller Lernkontexte

Die Praktiken des Wissenstransfers in LUGs sollen abschließend theoretisch reflektiert werden. Dass Wissenstransfer eines der zentralen expliziten Anliegen von FLOSS-Communities ist, legen andere Forschungsarbeiten bereits dar (vgl. Ghosh et al. 2002: 44; Huysman/Lin 2005; Lin 2004a; Glott et al. 2007; Weller/Meiszner 2008; Ghosh/Glott 2005; Sowe et al. 2008) — dies deckt sich mit meinen Erhebungen. Gestützt auf meine Erkenntnisse vertrete ich hier die These, dass sich die Informalität des Lernkontextes LUG sozial selektiv auswirkt. Da die Erhebungsmethode der teilnehmenden Beobachtung strukturelle Dimensionen wie Bildungshintergründe nur so weit erfassen kann, wie sie in der Praxis auftauchen, beziehe ich mich im Folgenden auf bereits existierende Forschung aus erziehungswissenschaftlichen, medienpädagogischen und soziologischen Kontexten. (Linux-bezogene Literatur aus der hier eingenommenen Perspektive ist mir nicht bekannt.)

Zunächst gilt es, informelles Lernen genauer zu bestimmen und zu Linux User Groups in Beziehung zu setzen: Informelles Lernen ist in den letzten Jahren in den Fokus der Erziehungswissenschaften gerückt. Helmut Bremer sieht eine Verlagerung in erziehungswissenschaftlichen Diskussionen und Ansätzen, die „in Bezug auf das Lernen das ‚Selbst‘ bzw. das Subjekt in den Vordergrund stell[t]“ (Bremer 2007: 210).⁴⁷ Bei all diesen Ansätzen habe sich das Interesse „von den Rahmungen, d. h. den Bildungsinstitutionen, Lehrenden, strukturellen und sozialen Bedingungen des Lernens hin zur Eigentätigkeit (und Eigenverantwortung) der Individuen verlagert“ (ebd.). Informelles Lernen nennt Bernd Overwien (2005) alle Aktivitäten, die außerhalb von institutionalisierten Lehrangeboten nach dem Erwerb von signifikantem (also längerfristig Bestand habendem) Wissen, Fähigkeiten oder Erkenntnissen streben. Ziele, Inhalte, Dauer des Lernprozesses usw. würden von den Lernenden individuell oder kollektiv festgelegt. Die Lernenden eignen sich die Inhalte selbstständig, ohne Lehrkräfte, an. Der Unterschied zu Alltagswahrnehmungen oder Sozialisation liege darin, dass die Lernenden ihre Aktivitäten bewusst als Lernprozess einordneten. Demgegenüber könne von ‚informeller Bildung‘ gesprochen werden, wenn Lehrende oder Mentor_innen die Lernprozesse gestalteten (vgl. Overwien 2005: 344 f.).⁴⁸ Die Informalität wird also durch die Selbstorganisation des kompletten Prozesses durch

⁴⁷Bremer fasst unter diese Diskussionen etwa selbstorganisiertes oder selbstgesteuertes Lernen, selbstbestimmtes oder selbstreguliertes Lernen, informelles Lernen und Kompetenzentwicklung. Diese einzelnen Konzepte sind für meine Arbeit nicht relevant.

⁴⁸Overwien bezieht sich mit dieser Definition auf David Livingstone (1999: 68f.).

die Lernenden bedingt und stark an der Präsenz von Lehrkräften festgemacht. In der LUG beinhaltet diese Selbstorganisation einerseits die Vereinsaktivitäten, durch die Räume und regelmäßige Ansprechpersonen überhaupt erst da sind. Andererseits — und darum geht es mir hier — heißt Selbstorganisation auch, dass Einzelne sich die benötigte Hilfe selbst verschaffen, indem sie mit einer konkreten Frage oder mit Neugierde dorthin kommen, sich Gehör verschaffen, erfolgreich Hilfe acquirieren, ihr technisches Problem in einer akzeptierten Weise formulieren.

Was die Präsenz von Lehrkräften in der LUG angeht, wurde gezeigt, dass die Lehr-Lernkonstellationen situativ sehr unterschiedlich sind. Fast jede Person übernimmt mal die eine, mal die andere Rolle. Lässt sich in dieser Gemengelage von informellem Lernen sprechen? Ist genau die Ungeklärtheit der Konstellationen Teil des informellen Lernkontextes? Um die beobachteten Konstellationen in Begriffe fassen zu können, ist ein kurzer Exkurs in Lernverhältnisse erforderlich. Dabei beziehe ich mich auf den kritischen Psychologen Klaus Holzkamp sowie auf Jean Lave und Etienne Wengers erziehungswissenschaftlichen Ansatz der *Communities of Practice*, auf den sich große Teile der FLOSS-Forschung beziehen. Für mögliche Lernkonstellationen hat Klaus Holzkamp angemessene Begriffe geprägt, nämlich Lehrlernen, partizipatives Lernen, kooperatives Lernen und autonomes Lernen. Er geht davon aus, dass Lernende ihre eigenen Handlungsspielräume erweitern möchten und verschiedene Lernkonstellationen dies in unterschiedlichem Maße begünstigen oder beeinträchtigen können. In meinem Zusammenhang geht es den Lernenden um die Handlungsfähigkeit im Umgang mit technischen Artefakten. Die Lernkonstellationen in der LUG umfassen sowohl ein eher frontales Lehrlernen an Vortragsabenden⁴⁹, als auch autonomes Lernen, nämlich dort, wo Einzelne in Auseinandersetzung mit ihren Computern selbst etwas herausfinden, z. B. durch Ausprobieren oder durch die Lektüre von Dokumentationen. In LUGs werden technische Probleme erst dann gemeinsam bearbeitet, wenn jemand allein nicht weitergekommen ist. Eine gewisse Bereitschaft zu autonomem Lernen wird in FLOSS-Communities vorausgesetzt (vgl. Holzkamp 1995: 513 ff.).

Der Großteil der Situationen bewegt sich — mit Klaus Holzkamp und Etienne Wenger betrachtet — zwischen kooperativem und partizipativem Lernen. Kooperativ Lernende definierten miteinander eine Lernproblematik, die sie dann ebenbürtig angingen. Dies ist etwa gegeben, wenn eine Kleingruppe in der LUG etwas Neues ausprobiert, das in einer

⁴⁹Auch der von mir nicht beobachtete Einsteiger_innenkurs, den die LUG organisiert hat, kann als Lehrlernen bezeichnet werden.

Fachzeitschrift vorgestellt wurde. Die Divergenzen, die sich durch unterschiedliche Perspektiven auf die Problematik oder durch unterschiedliche Wissensstände ergeben, seien nicht überwindbar, sollten das Lernverhältnis untereinander aber nicht dauerhaft gleich strukturieren, sondern immer wieder im Dialog ausgetragen werden. Die inhaltliche bzw. sachliche Überlegenheit Einzelner bleibe argumentativ hinterfragbar und müsse sich immer wieder neu erweisen, da die Verteilung von Wissen und Können im Fluss sei (vgl. Holzkamp 1995: 509 ff.).

Der Begriff des partizipativen Lernens stellt das Hineinwachsen in die Praxis einer Community (oder einer Berufsgruppe) in den Mittelpunkt. Etienne Wenger versteht unter *Communities of Practice* Gruppen, die sich — ob auf der Arbeit oder in der Freizeit — durch drei Dimensionen auszeichnen: Die Mitglieder teilen ein Unterfangen, das sie ständig neu definieren und verhandeln (1). Dazu gehen sie verbindliche Beziehungen ein (2). Auf Dauer entsteht dadurch ein geteilter Bestand an Ressourcen, z. B. Wissen, Routinen, ein Sprachgebrauch (3) (vgl. Wenger 1998: 73 ff.). Lave/Wenger sehen Lernen nicht nur als Wissenserwerb, sondern stellen es als situiertes und praxisgenerierendes Handeln in den sozialen Kontext der jeweiligen *Community of Practice*, es erweitert die Handlungsspielräume der Neulinge in der Community (vgl. Lave/Wenger 1991: 29, 49 f.). Ihre Fallstudien zu Lernkonstellationen hantieren begrifflich mit den Rollen klassischer Ausbildungsverhältnisse, in denen „Meister“ Lehrlinge ausbilden, die soweit in die Praxis hineinwachsen, bis sie selbst zu vollen Community-Mitgliedern werden (vgl. ebd. 53). Das Stadium der Ausbildung nennen die Autor_innen „legitimate peripheral participation“: Die Neulinge im Feld nehmen bereits an der Praxis teil, allerdings von einer peripheren Position aus, in der sie nicht die volle Verantwortung tragen und sich legitimerweise zum Lernen zurückziehen können (vgl. Holzkamp 1995: 502). Peripheralität ist positiv und dynamisch gemeint: Wer in einer peripheren Position sei, könne nach und nach mehr Zugang zu relevanten Ressourcen (zu Aktivitäten, Alten/„Meistern“, anderen Mitgliedern, zu Informationen, Teilnahmegelegenheiten, Artefakten) erlangen (vgl. Lave/Wenger 1991: 100 f.).

Die bisherige Forschung zu Lernprozessen in Open-Source- oder FLOSS-Communities bezieht sich stark auf den Ansatz der *Communities of Practice*, beschäftigt sich jedoch nicht genauer, geschweige denn kritisch mit den konkret beobachtbaren Lernverhältnissen.⁵⁰ Informelles Lernen in Open-Source-Projekten haben Rishab Ghosh und Rue-

⁵⁰Die Arbeiten legen ganz anderen Schwerpunkte: Während es Lin (2004b) z. B. um die Veränderung persönlicher Identitäten in Lernprozessen innerhalb einer Community of Practice geht, diskutieren Fried (2008) und O'Neil (2009) die Übertragbarkeit des Konzeptes auf Online-Gemeinschaften. Fried kommt

diger Glott aus der Perspektive von Softwareentwickler_innen und Arbeitgeber_innen untersucht. Die befragten Softwareentwickler_innen äußerten, dass sie viele Fähigkeiten in Communities besser erlernen könnten als im Rahmen von formalen Bildungsangeboten. Das gelte im technischen Bereich besonders für die Wiederverwendung von → *Quellcode* und im rechtlichen Bereich um Lizenzfragen. Diejenigen, die formalisierte Kurse durchlaufen hatten, präferierten das interaktive Lernen innerhalb einer Community, z. B. bei der Fehlerbehebung (Bugfixing) (vgl. Ghosh/Glott 2005: 29).⁵¹ Die Rollen der Beteiligten seien nicht an feste Personen gebunden wie in formalen Lernzusammenhängen (Lehrer_innen, Schüler_innen) und die Verteilung von Aufgaben und Verantwortlichkeiten sei nicht so rigide (vgl. Glott et al. 2007: 68). Die Lernmaterialien kämen direkt aus der Community selbst und würden dort fortlaufend weiterentwickelt. Anders als in formellen Settings, wo Kohorten oft voneinander isoliert lernen, könnten in Communities die Neueren direkt von den Erfahrungen der langjährig Aktiven lernen (vgl. Weller/Meiszner 2008: 3).

Der *Community of Practice*-Ansatz stößt bei der Analyse der Lernverhältnisse in LUGs jedoch an Grenzen: Das hier beschriebene partizipative Lernen ist *eine* wichtige Lernkonstellation in LUGs. Die Linux-Nutzer_innen bzw. LUG-Mitglieder und -Gäste haben gemeinsame Anliegen, gehen zur Umsetzung dauerhafte und verbindliche Beziehungen ein und bauen einen gemeinsamen Korpus an Wissen und Praxis auf. Jedoch ist die Rollenaufteilung zwischen Alten und Neuen bzw. Meister_innen und Lernenden nicht klar gegeben, sondern im ständigen Fluss. Die Kleingruppen ergeben sich meist spontan und themenbezogen. Deshalb kann nicht davon gesprochen werden, dass Einzelne eine „permanente unhinterfragte Überlegenheit“ (Holzkamp 1995: 514) etabliert haben, wie sie eigentlich die „Meister_innen“ im partizipativen Lernverhältnis einer *Community of Practice* hätten. Selbst diejenigen, die über ein relativ hohes Maß an Problemlösungskompetenz verfügen, sind nicht auf alle Themen gleichermaßen spezialisiert. Die unterschiedlichen Spezialisierungen sind vielmehr ein Charakteristikum der Gruppen. Der *Community of Practice*-Ansatz kann Hierarchien, die für Neulinge hinderlich sein können, aufzeigen. Lave/Wenger thematisieren am Rande die Machtbeziehungen in Communities: Peripheralität wirke sich dann negativ auf den Lernprozess aus, wenn der neuen Person Res-

zu dem Ergebnis, dass sich auch im Internetkontext von solchen Communities sprechen lässt, weil sie alle Dimensionen einer *Community of Practice* (s.u.) aufwiesen. O’Neil, der sich mit Verhältnissen von Autonomie und Autorität in „Online Tribes“ beschäftigt, verwirft dagegen das Konzept, weil es die spezifische „affektive Qualität“ von Internetgemeinschaften nicht erfasse (vgl. O’Neil 2009: 26).

⁵¹Im Bezug auf Arbeitgeber_innen stellen die Autoren eine wachsende Anerkennung für Open-Source-Communities fest, die sich in auch in einer Anerkennung von dort erworbenen, nicht zertifizierten Qualifikationen zeige (vgl. ebd.).

sources vorenthalten würden (vgl. Lave/Wenger 1991: 36). In den beobachteten Szenen ist dies z. B. dann der Fall, wenn Helfer_innen anstelle der Fragenden Schritte ausführten, ohne sie zu erklären, so dass die Nachvollziehbarkeit in Frage steht. Holzkamp gesteht dieser Konstellation — im Vergleich zum hierarchisch angelegten Lehrlernen — zwar zu, nicht dauerhaft dichotom strukturiert zu sein, weil die Lernenden nach und nach selbst zu „Meistern“ werden. Er kritisiert dennoch das traditionell organisierte Verhältnis, in dem der Meisterstatus durch Neulinge grundsätzlich anzuerkennen sei (vgl. Holzkamp 1995: 506 ff.). Der Ansatz allein kann die Vielfalt der beobachteten Lernkonstellationen (Lehrlernen, autonomes Lernen, kooperatives und partizipatives Lernen) jedoch nicht ausreichend abbilden. Eine LUG ist zugleich der Ort des technischen Supports und der Problemlösung als Community-Praxis *und* ein Ort der Weiterbildung abseits von konkreten Fragen oder Problemen: Anwesende probieren ebenso spontan und ebenbürtig Software zusammen aus oder beschäftigen sich den ganzen Abend allein mit ihrem Computer.

Zurück zur Informalität des Lernzusammenhangs einer Linux User Group: Informelles Lernen wird unter anderen Faktoren an den Konstellationen festgemacht, in denen Menschen lernen, und am Grad der Institutionalisierung dieser Konstellationen. Mit den eben angeführten Theorien wurde gezeigt, wie breit das Spektrum der Lernverhältnisse in LUGs ist, ohne dass es dafür explizite Vereinbarungen gibt. Dies wird auch der Tatsache geschuldet sein, dass LUGs sich nicht auf ihre Pädagogik hin befragen, sondern sich über ihre gemeinsamen Praktiken konstituieren. Wie voraussetzungsvoll der technische Support sowohl für Helfende als auch für Fragende ist, wurde anhand der untersuchten Situationen deutlich, auch wenn dies in der Praxis implizit bleibt. Die LUG kann als Kontext informellen Lernens bezeichnet werden. Die oben zitierten Texte sehen dies als besonderen Vorteil von FLOSS-Projekten im Gegensatz zu formalen Weiterbildungsmöglichkeiten (vgl. Ghosh/Glott 2005; Glott et al. 2007). In der erziehungswissenschaftlichen Literatur wird informellem Lernen jedoch ein hoher Grad an sozialer Selektivität nachgewiesen. Dies gilt schon für institutionelle Weiterbildungsangebote, die in der Erwachsenenbildung als „doppelt selektiv“ bezeichnet werden.

„Ausgedrückt wird damit, dass vor allem diejenigen an institutionalisierter Weiterbildung partizipieren, die ohnehin schon über höhere Bildungsabschlüsse verfügen und dadurch in (gehobenen) beruflichen Bereichen tätig sind, die wiederum höhere Bildungsaktivitäten erfordern“ (Bremer 2007: 22).

Gerade für den Computerbereich wurde jedoch gezeigt, dass die jetzt Erwachsenen, die

sich als Jugendliche ihre ersten Kompetenzen in den 1990er Jahren oder weitaus früher angeeignet haben, diese oft außerhalb der Schule und auf eine geschlechtlich konnotierte Art und Weise erworben haben. Dabei wurden neben Peergroups Väter als zentrale Vermittler des Computers an ihre Söhne ausgemacht:

Der Jugendsoziologe Claus Tully hat informelles Lernen im Computerbereich bei Jugendlichen erforscht: In den 1990er Jahren hätten sich Jugendliche Computerwissen hauptsächlich außerschulisch angeeignet, da die Schule aus verschiedenen Gründen Schwierigkeiten mit der Vermittlung von Computerkompetenzen hatte.⁵² Die Jugendlichen hätten sich zentrale Kompetenzen außerhalb der Schule angeeignet, z. B. durch Bildungsangebote, aber auch durch Computerzeitschriften oder durch spielerisches Ausprobieren in der Peergroup, im praktischen Umgang mit Computern (vgl. Tully 1994: 23, 179 ff.). Tully stellt heraus, dass diese Informalisierung eine privatisierte Zuständigkeit für Wissens- und Kompetenzerwerb im Bereich Computer hervorgebracht habe. Er sieht als mögliche Konsequenz dieser Art von Eigenverantwortung eine soziale Differenzierung, da die materiellen, sozialen und persönlichen Ressourcen Einzelner den Kompetenzerwerb bedingen (vgl. Tully 1994: 184). Diejenigen, die bereits in ihren Elternhäusern Zugang zu Rechnern hatten, die Geld in zusätzliche (nicht unbedingt informelle) Bildungsangebote investieren konnten, Zeit in eigenes Experimentieren mit Hard- und Software stecken konnten, Helfer_innen aus ihrem persönlichen Umfeld kannten, waren im Vorteil.

Das informelle Lernen im Elternhaus erfragten auch Jane Margolis und Allan Fisher in ihrer gendersensiblen Untersuchung unter US-amerikanischen Informatikstudent_innen (Margolis/Fisher 2002). Der Frauenanteil unter den Absolvent_innen lag in der zweiten Hälfte der neunziger Jahre unter 20 Prozent und viele Frauen brachen den Studiengang ab („leaky pipeline“) (vgl. Margolis/Fisher 2002: 2 f.). Die Rolle von Computern im Elternhaus wird von den Autor_innen als sehr wichtig eingeschätzt. 40 Prozent der männlichen Befragten und 65 Prozent der weiblichen waren in Haushalten groß geworden, in denen einer oder beide Elternteile beruflich oder hobbymäßig mit Computern zu tun hatten (vgl. ebd. 20). Schichtzugehörigkeit bzw. Einkommen der Eltern spielt hier ebenfalls eine große Rolle: Kinder aus Familien mit mittleren und höheren Einkommen würden zu Hause an Computer herangeführt, lange bevor sie in Bildungsinstitutionen kämen. Zudem hätten in

⁵²Als Probleme führt Tully an, dass Computer erst ab Mitte der 1990er Jahre überhaupt an Schulen aufgetaucht seien und dass die Technologien sich viel schneller entwickelten als Lehrpläne anpassbar seien. Darüber hinaus seien die mannigfaltigen Anwendungsfälle, auf die Schüler_innen künftig treffen könnten, durch Unterricht kaum abdeckbar. Er vernachlässigt hier m. E. das damalige Fehlen geschulten Lehrpersonals.

den späten 1990er Jahren „weiße Haushalte“ mehr als doppelt so häufig Computer besessen wie „schwarze“ oder „lateinamerikanische“ Haushalte, eine Spaltung, die sich durch alle Einkommensgruppen gezogen hätte (vgl. ebd. 20 f.). Eine ältere Arbeit von Joseph Giacquinta et al. (1993) hat ergeben, dass über die Hälfte der befragten Familien den Computer an einem Ort in der Wohnung platziert hatte, an dem er in erster Linie dem Vater oder den Söhnen zugänglich war, etwa im Arbeits- oder Schlafzimmer der Erwachsenen oder im Kinderzimmer des Sohnes (vgl. Giacquinta et al. 1993: 84; zitiert bei Margolis/Fisher 2002: 23). Bei der Frage, woher die Befragten ihre ersten Instruktionen zum Computer bekommen hätten, belegen die Autor_innen „father-son internships“ (ebd. 24); die Heranführung der Jungen an Computer erfolgt viel häufiger durch die Eltern und besonders durch den Vater als bei Mädchen. Mädchen, die dazu ermutigt werden, sich mit dem Rechner auseinanderzusetzen, seien in der Regel um einiges älter; dabei gehe es in erster Linie um die Zeit, in der sie Informatik in der Schule wählen könnten (vgl. Margolis/Fisher 2002: 24 f.; siehe auch Haralanova 2010: 79 ff.).

Im Jahr 2010 gibt es laut der *JIM-Studie (Jugend, Information, (Multi-) Media)* des Medienpädagogischen Forschungsverbundes Südwest unter den 12–19-Jährigen inzwischen keine nennenswerten Unterschiede zwischen Schüler_innen verschiedener formaler Bildungsstände und Geschlechter mehr, was den reinen Zugang zu Computern und dem Internet betrifft. Im Hinblick darauf, ob sie eigene Computer besitzen (vgl. Kutteroff et al. 2010: 25) und vor allem, *wie* und *wofür* sie das Internet nutzen, wurden sehr wohl Unterschiede festgestellt: So spielt der Bildungshintergrund eine Rolle dafür, ob, wie gezielt und wie kritisch hinterfragend sie das Internet nutzen, um Informationen zu recherchieren (vgl. ebd. 33), welche Kommunikationskanäle sie nutzen und in welche sozialen Netzwerke sie sich begeben (vgl. ebd. 41). Soziale Netzwerke hätten zudem verschiedene Reputationen, an denen Jugendliche sich bei der Wahl stark orientierten (vgl. Pilarczyk 2011).

Die in LUGs beobachteten Szenen informellen Lernens stehen vor dem Hintergrund bereits früher erworbenen Kulturkapitals aus anderen Lebenskontexten. Da die Erhebungsmethode es nicht systematisch erlaubt, für die konkreten Personen Aussagen über Vorbildung im Computerbereich oder formale Bildungsabschlüsse zu treffen, wurde dieser Hintergrund über die herangezogene Literatur abgebildet. Die Ergebnisse sind bei der Untersuchung von informellem Lernen in LUGs sehr relevant, auch wenn die bisherige FLOSS-Forschung sie nicht aufgegriffen hat: Die jetzt Erwachsenen haben ihre Fähigkei-

ten *tendenziell* in einem informellen, außerschulischen Kontext erworben.⁵³ In die Heranführung von Kindern an Computer werden dabei in vielen Fällen Vorannahmen eingeflossen sein, die in Abhängigkeit vom zugeschriebenen Geschlecht Interesse oder Fähigkeiten in Bezug auf Computer unterstellten. Hier überschneiden sich informelles Lernen und Sozialisation. Während bis in die 1990er Jahre der Zugang zu Computern selbst sehr ungleich verteilt war, spielt bis heute der Bildungshintergrund für die Ausbildung spezifischer Nutzungsgewohnheiten eine große Rolle. Wer bereits entsprechendes kulturelles und soziales Kapital akkumuliert hat, etwa durch die soziale Herkunft, kann von Weiterbildungsangeboten besser profitieren und sich im informellen Kontext einer LUG besser orientieren. Die Szenen in der LUG basieren auf den sozialen und kulturellen Voraussetzungen, die die Beteiligten mitbringen. Diese Voraussetzungen tragen dazu bei, dass sie sich in informelle Lernkontexte überhaupt hinbegeben und dass sie dort bestimmte Rollen mit bestimmten Erfolgschancen einnehmen. Vorhandene soziale Ungleichheiten werden — auch dadurch, dass der technische Support nicht explizit methodisch reflektiert ist — tendenziell stabilisiert und im Bereich der freien Software fortgeschrieben, auch wenn eine LUG weniger homogen in Bezug auf Bildungshintergründe ist als Ghosh et al. (2002) es für FLOSS-Programmierer_innen herausgestellt haben.

⁵³Hier möchte ich nicht so verstanden werden, dass allein eine Formalisierung des Kontextes die sozialen Schief lagen beseitigen würde. Auch Schulerfolg hängt maßgeblich vom Elternhaus ab.

6. Techniknutzung und -anpassung

In diesem Kapitel rücken die Habitusfragmente von FLOSS-Communities im Umgang mit freier Software in den Mittelpunkt. Dazu werden Praktiken der Techniknutzung und -anpassung anhand von drei Komplexen aus den Beobachtungsprotokollen herausgearbeitet: Erstens geht es um die Nutzung textbasierter Programme. Zweitens werden kreative Nutzungspraktiken und Einschreibungen ästhetischer Vorstellungen in Software beleuchtet. Drittens werden Zeitverhältnisse beim Umgang mit Technologien untersucht. In jedem dieser Abschnitte geht es abschließend um die Frage, ob und wie in diesen Nutzungspraktiken Differenzen produziert werden.

6.1. Die Konstitution von technischer Expertise über textbasiertes Arbeiten

Bei der Nutzung von freier Software zieht sich die deutlichste Distinktionslinie zwischen grafischen und textbasierten Programmen. Den Bemerkungen von Laien, die mit der Eingabe von Befehlen nicht vertraut sind, ist oft zu entnehmen, dass sie denken, jemand programmiere, wenn sie eine → *Konsole* sehen, obwohl es sich eigentlich um die Bedienung des Computers mit anderen Werkzeugen handelt. Strenge Verfechter_innen von textbasierten Programmen äußern sich mit Bezeichnungen wie „Klicki-Bunti“ teilweise abfällig über aufwändig gestaltete Benutzeroberflächen. Sie werfen grafischen Programmen vor, dass nicht die Funktionalität, sondern die grafische Aufmachung im Vordergrund stehe.¹ Was das nötige Wissen angeht, gibt es bei der Bedienung einen deutlichen Unterschied zwischen grafischen und textbasierten Programmen: Grafische Programme können auf der Benutzeroberfläche über Menüs gefunden und gestartet werden.² Die Oberfläche ist

¹Es wird ferner unterschieden zwischen simplen, „schlanken“ Windowmanagern und vollen → *Desktopoberflächen*, die z. B. auch grafische Effekte darstellen können und deutlich ressourcenaufwändiger sind.

²Man muss nicht wissen, wie genau das Programm heißt, sondern das Menü liefert Zusatzinformationen. Wer den Bereich „Grafik“ anklickt, findet eine Auflistung der installierten Programme mit der Funktion, z. B. „GIMP Bildbearbeitung“.

in der Regel durch Konventionen in der Menüführung „intuitiv“ gestaltet, beschriftete und bebilderte Schaltflächen erlauben auch Neulingen eine Orientierung. Claus Tully betont, wie stark sich die Lernzeiten für Computerprogramme verkürzt haben, seit Menüführungen diese Prozesse vereinfachen (vgl. auch Turkle 1998: 51). Er sieht diese Erleichterungen gleichzeitig als eine „Entwertung“ vormals angeeigneten Geschicks“ derjenigen, die als frühe EDV-Nutzer_innen vormals Expertenrollen innehatten (Tully 1994: 24).

Textbasierte Programme sind standardmäßig nicht in den Startmenüs erwähnt, sie werden in Konsolen ausgeführt, indem man den Programmennamen und eventuelle weitere Informationen dort eingibt. Es ist nicht auf den ersten Blick ersichtlich, ob das gewünschte Programm installiert ist.³ Darüber hinaus ist ein Wissen um die richtige Syntax nötig: Die Programme, die über einen Befehl ausgeführt werden, bieten oft zahlreiche Optionen an, die über so genannte Argumente dem Befehl gleich mitgegeben werden. Abb. 6.1 zeigt dies am Beispiel *ls*, das den Inhalt eines Ordners anzeigen kann. Mit Argumenten, z. B. *-lht* (Abb. 6.1b), können die Fülle und Reihenfolge der Informationen bestimmt werden.⁴

```
silke@lovelace:~/computer/dokus$ ls
Beginners Guide To Drupal.pdf  drupal6          latex
debian-packaging.pdf          galileo-linux-buch  mailman-admin.pdf
```

(a) *ls*

```
silke@lovelace:~/computer/dokus$ ls -lht
insgesamt 1,7M
drwxr-xr-x 2 silke silke 4,0K 2011-11-19 10:23 latex
-rw----- 1 silke silke 1,3M 2011-10-12 14:41 Beginners Guide To Drupal.pdf
drwxr-xr-x 2 silke silke 4,0K 2010-11-19 14:29 drupal6
-rw----- 1 silke silke 160K 2010-08-06 16:38 mailman-admin.pdf
drwxr-xr-x 5 silke silke 20K 2009-09-28 08:44 galileo-linux-buch
-rw----- 1 silke silke 286K 2009-09-10 09:28 debian-packaging.pdf
```

(b) *ls -lht*

Abb. 6.1.: Anzeigen eines Verzeichnisinhaltes und Regulierung der gewünschten Informationen

Über alle textbasierten Programme hinweg existieren zwar Ähnlichkeiten in der Bedeutung von Argumenten, es gibt jedoch keine Konventionen, die aus verschiedenen Gründen auch schwer einzuführen wären. Üblicherweise wissen Nutzer_innen textbasierter

³Ob ein Programm installiert ist, findet man z. B. heraus, indem man den Anfang eines Programmnamens in die Konsole eingibt und durch Drücken der Tabulatortaste eine Übersicht der Programme erhält, deren Namen so anfangen. Das erfordert bereits das Wissen darum, dass in die entsprechende → *Shell* eine Art Autovervollständigung eingebaut ist, die mit der Tabulatortaste aufgerufen wird. Diese Information hatten viele in der LUG beobachtete Nutzer_innen nicht.

⁴Die Optionen in diesem Beispiel: „l“ steht für „long“, also eine ausführliche Darstellung jeder Datei (mit Zugriffsrechten, Besitzer_in, Dateigröße, Zeit des letzten Zugriffs). „h“ steht für „human readable“ und bezieht sich auf die Darstellung der Dateigröße in leicht lesbaren Kilobyte- und Megabyte-Angaben, „t“ bezieht sich auf die Sortierung nach Änderungsdatum, so dass die zuletzt veränderte oder hinzugefügte Datei ganz oben steht.

Programme einige häufig verwendete Argumente auswendig. Für den Rest greifen sie auf die Programmdokumentationen zu, die so genannten *man pages*⁵, die automatisch mit den Programmen zusammen installiert werden. *Man pages* erläutern die Möglichkeiten und die Bedienung des Programms, manchmal auch mit Beispielen. Sie verstehen zu können, erfordert jedoch eine gewisse Übung, zumal sie aus den Federn der jeweiligen Entwickler_innen stammen und zwar eine einheitliche Informationsabfolge einhalten, sich in Qualität und Umfang aber stark unterscheiden. Textbasierte Programme sind aus verschiedenen Gründen so verbreitet: Sie laufen nicht nur problemlos auf älteren Computern, sondern sind auch in jeder → *Distribution* gleich, während die verschiedenen grafischen Oberflächen sich stark unterscheiden. Zudem sind textbasierte Programme für systemadministratorische Tätigkeiten essenziell: Wer → *Server* betreut, arbeitet in der Regel „remote“ mit einem textbasierten → *ssh*-Login, wo nur textbasierte Programme zur Verfügung stehen.⁶ Des Weiteren lassen sich diese Programme in → *Shellskripte* einbauen. Dies ist auch vor allem für die Arbeit von Systemadministrator_innen wichtig, um wiederkehrende Aufgaben, z. B. Datensicherungen, zu automatisieren.

In den folgenden Abschnitten wird die Nutzung textbasierter Programme daraufhin beleuchtet, wie sie zum „Einsatz im Spiel“ (Bourdieu) werden: Wie wird *über* textbasiertes Arbeiten Differenz zwischen Nutzer_innen hergestellt?

Vorab noch eine allgemeinere Beobachtung: Bei der Nutzung von → *X-Terminals* war auffällig, dass viele Nutzer_innen ihr *X-Terminal* so eingerichtet haben, dass es für Betrachter_innen schwer leserlich ist, vor allem für Personen, die nicht direkt vor dem Rechner sitzen, sondern zuschauen.

Dabei stelle ich fest, dass ich sowohl bei A48 als auch bei A39 ziemlich schlecht lesen kann, was auf ihren *Shells* steht. (...) Beide haben einen schwarzen Hintergrund mit weißer Schrift eingestellt und zwar recht kleine Schrift, vor allem A39 (LUG 2009d).

Er hat über seinem Webbrowser ein hellgraues *X-Terminal* eingeblendet, das transparent ist. Ich frage mich, wie er die Schrift in dem Terminal erkennt (LUG 2009l).

Die Befehlseingabefenster sind in vielen Fällen so eingerichtet, dass genau die Person, die vor dem Rechner sitzt, sie lesen und bedienen kann: Die Schriftgröße und -farbe und der Kontrast zum Hintergrund verhindern, dass Beobachtende mitlesen können. Damit will

⁵ *Man pages* ist eine Kurzbezeichnung für „manual pages“, Handbücher.

⁶ Der Server steht also z. B. in einem weit entfernten Rechenzentrum, über die Netzwerkverbindung wird meist keine komplette grafische Oberfläche übertragen.

ich nicht unterstellen, dass die jeweiligen Einstellungen bewusst so gewählt werden, um Mitlesen zu verhindern. Eher werden diese Effekte als schön oder „stylish“ empfunden und fallen damit in den Bereich des Geschmacks. Da diese Beobachtung ausgerechnet bei Konsolen keine Seltenheit war, wirft sie die Frage auf, ob die Einstellungen, die zu Unleserlichkeit führen, widerspiegeln, was Konsolen für Außenstehende sowieso sind: Black Boxes. Gleichzeitig sind Konsolen für diejenigen, die sich ihrer zu bedienen wissen, genau das, was Linuxrechner von Black Boxes unterscheidet: Sie sind *die* Zugangsmöglichkeit zum Betriebssystem.

6.1.1. Distinktion gegenüber Nutzer_innen grafischer Programme

Die Verwendung textbasierter Programme wird teilweise explizit der Verwendung grafischer Programme vorgezogen. Ein Beispiel dafür ist eine Frage aus einem Prüfungssimulator für die Zertifizierungsprüfung des → *Linux Professional Institute*. In dem Simulator kann auf die Frage, mit welchem Programm man sich ins Internet einwählen kann, nur eine Antwort angekreuzt werden. Die als richtig gewertete Antwort ist das textbasierte Programm *wvdial*. Das grafische Programm *kppp*, das ebenfalls zur Auswahl steht, dient dem gleichen Zweck und wäre demnach auch eine korrekte Antwort, wird im Simulator aber nicht akzeptiert.

Auch in der LUG gibt es Situationen, in denen Bezüge auf textbasierte Programme hergestellt werden, die eine mehr oder weniger deutliche distinguierende Wirkung haben. In einer bereits erwähnten Situation stellt ein Helfer die Rückfrage, warum der Fragende überhaupt mit einem grafischen Programm arbeite:

A76 fragt, woran das denn liege, dass er im Programm *gnome-commander* nicht in den Admin-Betrieb wechseln kann, obwohl die Funktion da eigentlich eingebaut sei. Er demonstriert uns, dass er nach dem Passwort gefragt wird, dann aber nicht root sei. (...) A77 reagiert mit der Frage, wozu A76 denn den *gnome-commander* nutze und nicht die Shell. Ich höre A1 und A5 reagieren: A1 schnauft auf, als rege ihn die Antwort auf. A5 sagt, das sei keine Lösung des Problems (LUG 2008c).

Ich interpretiere die Rückfrage von A77 als Hinweis darauf, dass sich A76 nicht mit einem Programm abmüht, das die Mühe nicht wert sei, zumal es nicht einmal fehlerfrei funktioniere. Er wertet A76s Nutzungspraxis ab und stellt sich selbst in das Licht des Systeme-

administratoren, der textbasiert arbeitet.⁷ Gleichzeitig verweigert er in diesem Moment die Hilfe. Von den anderen Anwesenden wird dies jedoch nicht akzeptiert; sie ergreifen Partei für eine Fragekultur in der LUG, in der Fragende nicht abfällig behandelt werden sollten. A1, ebenfalls Systemadministrator, ist unter ihnen. In einer anderen Situation äußert er sich über seine Gewohnheit, textbasiert zu arbeiten: Er ist in der Helferrolle. Auf seinem Notebook ist der Bildschirmhintergrund, der → *Desktop*, zu sehen, auf dem extrem viele Dateien abgelegt sind.

Er beschwert sich über die Unordnung auf seinem eigenen Rechner und fragt, warum es eigentlich keine „Putzfrauen“ gebe, die Rechner aufräumen. (...) A54 sagt, sie ordne immer alles auf ihrem Rechner und lege es gleich ab. Er erklärt A54, „normale Leute“ würden mit ihrem grafischen Desktop arbeiten und das über den Dateimanager machen. Er arbeite aber meistens mit der Konsole, da sei ihm egal, wie viel z. B. im Ordner ‚Desktop‘ liege. Aber auf dem Desktop sei Chaos zu sehen (LUG 2009h).

A1 formuliert einen Unterschied zwischen „normalen Leuten“, die ihre Dateien mit einem grafischen Dateimanager ordnen und zu denen er A54 zählt, und Leuten wie ihm selbst, die textbasiert arbeiten. Damit produziert er einen Unterschied, der nicht unbedingt als Abwertung grafischer Programmen gelesen werden muss. Gleichzeitig wird jedoch in dem Protokollausschnitt deutlich, dass er eine andere Vorstellung davon hat, was der Desktop ist. A54 scheint den Desktop eher als eine Arbeitsfläche, ähnlich einer Schreibtischoberfläche, zu begreifen, auf der nichts herumliegen soll. Dateien legt sie in anderen Ordnern ab, während der Desktop quasi den Eingang in den Rechner symbolisiert. A1 dagegen betrachtet den Desktop als *einen Ordner unter vielen* in seinem persönlichen Ordner. Er scheint die Ordnerstruktur des Unix-Dateibaum vor Augen zu haben, in der ‚Desktop‘ ein Ordner ist. Die grafische Darstellung auf dem Bildschirmhintergrund interessiert ihn nicht und das führt er auf sein textbasiertes Arbeiten zurück. Seine Gewohnheit, auf dem Desktop keine Programmstarter abzulegen, schließt daran an:

A1 zeigt ihr, dass sie *Icedove* jetzt im Menü findet, zieht es ihr in die obere Leiste und zeigt ihr, dass sie es auch auf den Desktop legen könne. Von dort wirft er es aber wieder in den Papierkorb und sagt, er finde es schön, wenn die Programmstarter oben in der Leiste seien (LUG 2009h).

Wenn er den Desktop als beliebigen Ordner betrachtet, widerspricht es seiner praktischen Logik, ausgerechnet dort Startknöpfe für grafische Programme abzulegen. Die distinguiere-

⁷A77 ist von Beruf auch Systemadministrator.

rende Wirkung findet hier auf einer anderen Ebene statt als zwischen A77 und A76: Es geht nicht darum, welches Programm verwendet werden soll, sondern darum, in welchen Kategorien der Linux-Computer begriffen wird.

Das Konzept eines grafischen Desktop gibt es erst seit 1984, als die Firma *Apple* es auf seinen *Macintosh*-Rechnern realisierte. Bis dahin waren Computer nur über die → *Kommandozeile* zu bedienen. Die US-amerikanische Soziologin und *Science-and-Technology*-Forscherin Sherry Turkle hat diesen grundlegenden Wandel in der Computernutzung wie folgt beschrieben: Der *Macintosh* habe die Nutzer_innen dazu ermuntert, „auf der Oberflächenebene der visuellen Repräsentation zu verbleiben“ (Turkle 1998: 49). Dadurch sei es zwar für viele einfacher geworden, die Maschine zu bedienen, die Nutzer_innen seien aber nicht mehr eingeladen worden, sich mit Ebenen der Technologie unterhalb dieser Benutzeroberfläche zu beschäftigen. Mehr noch, das Bedienen eines Rechners über die grafische Oberfläche habe den Nutzer_innen ganz das Gefühl genommen, einer Maschine Befehle zu erteilen, weil er so interaktiv gestaltet sei. Dies sei fast eine Erfahrung gewesen, die der zwischenmenschlichen Kommunikation ähnele, wo man auch nicht immer genau wisse, was die andere Person denke.

„[A]uf ähnliche Weise brachte uns der Macintosh bei, zu verhandeln, statt zu analysieren. (...) Er förderte das Spielen und Improvisieren. Den Macintosh zu beherrschen bedeutete, sich einen Überblick über die Lage der Dinge zu verschaffen, und nicht, die Hierarchie der zugrundeliegenden Struktur und der Grundregeln aufzuklären“ (Turkle 1998: 51).

Das Verständnis des Desktop als Ordner, das A1 formuliert, reflektiert, dass er sich dieser „Hierarchie der zugrundeliegenden Struktur“ bewusst ist. Turkle fasst hier sehr trefflich eine in FLOSS-Communities wirkende Distinktion: Das Betriebssystem ist grundsätzlich so angelegt, dass es Nutzer_innen dazu einlädt, sich mit allen Ebenen der Software auseinanderzusetzen, für die sie sich interessieren oder auf denen sie ein technisches Problem lösen möchten. Viele nutzen diese Möglichkeit, tauchen an unterschiedlichen Stellen tiefer ein (vgl. Abschnitt 6.2.1). Dazu gehört meistens ein Interesse für die Software und den Aufbau des Betriebssystems *jenseits* der grafischen Oberfläche. Dies spiegelt sich auch im Logo eines Community-Events wider, dessen Motto „Wissen, was drin steckt“ war (vgl. Abb. 6.2). Das Maskottchen, der Pinguin *Tux*, hat in der Abbildung mit seinem Schnabel die Oberfläche aufgebohrt und schaut hindurch. Damit symbolisiert er die Gestaltbarkeit von freier Software unterhalb des zuerst Sichtbaren.

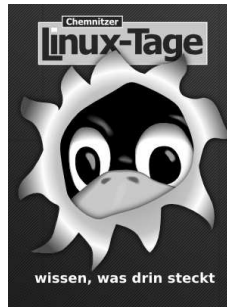
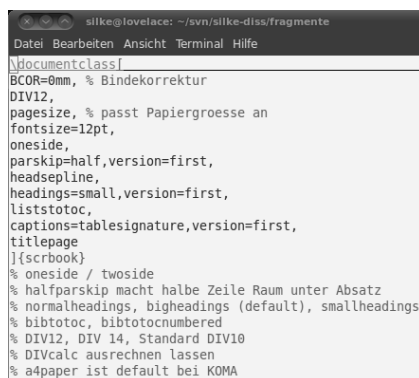


Abb. 6.2.: Logo eines Community-Events: „wissen, was drin steckt“

In den unterschiedlichen Software-Nutzungspraktiken in der LUG wird gleichzeitig sichtbar, dass viele die spezifischen Möglichkeiten, die Linux bietet, nicht nutzen. Längst nicht alle haben Interesse oder Zeit, sich mit textbasierten Programmen zu beschäftigen, viele brauchen keine individuellen Anpassungen, sondern nutzen die Standardprogramme so, wie sie sind. Sie fragen sich nicht, ob sie die Software gerne anders hätten. Die Entwicklung der grafischen Oberflächen ist seit etlichen Jahren so weit, dass nicht nur alle täglichen Arbeitshandlungen mit grafischen Programmen erledigt werden können, sondern auch viele administratorische Aufgaben wie Firewallkonfiguration oder Systemüberwachung. Gerade dadurch gibt es viele Hilfesituationen, in denen das textbasierte Arbeiten sehr distinguierend wirkt: Für Hilfesuchende ohne Kommandozeilenerfahrung sind die Befehle unverständlich, so dass diese Wirkung schon eintreten kann, wenn ihre Helfer_innen nur ein Terminal öffnen. Sie erkennen teilweise keinen Unterschied zwischen Programmieren und dem Bedienen des Rechners mit gegebenen textbasierten Programmen. („A79 arbeitet auf einer → *Shell* und gibt ein *hwclock set*. (...) A78 sieht zu und sagt, irgendwann beschäftige sie sich auch mal mit ‚Codes‘ und nehme sich dafür zwei bis drei Monate frei, in denen sie nichts anderes tue. ‚Ich bin froh, dass ich Euch hab‘“ (LUG 2009j).) In Situationen, in denen Praktiken des textbasierten Arbeitens so wirken, scheint Tullys These nicht zu stimmen, dass grafische Oberflächen das Können derjenigen entwerten, die Expert_innen waren, als noch ausschließlich textbasiert gearbeitet wurde (vgl. Tully 1994: 24). Im Gegenteil verhilft diese Praxis weiterhin zu einer Expert_innenrolle. Umgekehrt war ich nach Beobachtungen in verschiedenen Community-Kontexten beim → *Bug Jam* überrascht, dass so viele der fortgeschrittenen User_innen, die sich dort zum Korrigieren von Fehlern in Software getroffen hatten, grafische Programme verwendeten (LUG 2009l): Es kann nicht davon gesprochen werden, dass ab einem bestimmten „Niveau“ alle Nutzer_innen textbasiert arbeiten.

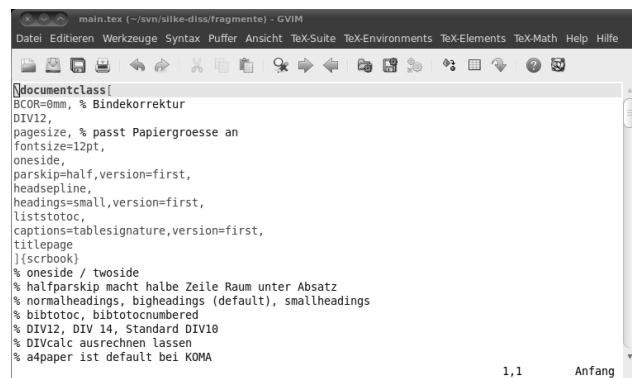
6.1.2. Spezialwissen auf die Spitze treiben: Der Texteditor *vim*

Texteditoren zu verwenden, ist eine grundlegende Praxis von FLOSS-Communities. In Editoren können nicht nur eigene Textdateien verfasst werden, sondern sie dienen in erster Linie zum Bearbeiten von Konfigurationsdateien und teilweise zum Programmieren. Unter Linux gibt es eine große Auswahl an Editoren, die jeweils über bestimmte Funktionalitäten verfügen. In der Regel können sie z. B. die Syntax von Programmiersprachen für eine größere Übersichtlichkeit farblich gestaltet anzeigen, sie verfügen teilweise über Schnittstellen für ganz bestimmte Zwecke. Manche Editoren sind standardmäßig in jeder Linuxdistribution enthalten. Neben streng textbasierten und rein grafischen gibt es auch Mischformen: Der Editor *mcedit* etwa bietet trotz textbasierter Oberfläche Schaltflächen für Mausklicks. Die beiden bekanntesten, verbreitetsten und umfangreichsten Editoren heißen *vi* / *vim* und *Emacs*. Für sie gibt es jeweils grafische Umsetzungen, die alternativ zur textbasierten Oberfläche benutzt werden können, so dass Schaltflächen und Menüs zur Verfügung stehen (Abb. 6.3).



```
silke@lovelace: ~/svn/silke-dis/fragmente
Datei Bearbeiten Ansicht Terminal Hilfe
\documentclass[
BCOR=0mm, % Bindekorrektur
DIV12,
pagesize, % passt Papiergrösse an
fontsize=12pt,
oneside,
parskip=half,version=first,
headsepline,
headings=small,version=first,
liststotoc,
captions=tablesignature,version=first,
titlepage
]{scrbook}
% oneside / twoside
% halfparskip macht halbe Zeile Raum unter Absatz
% normalheadings, bigheadings (default), smallheadings
% bibtotoc, bibtotocnumbered
% DIV12, DIV 14, Standard DIV10
% DIVcalc ausrechnen lassen
% a4paper ist default bei KOMA
```

(a) Textbasiert: vim



```
main.tex (~/svn/silke-dis/fragmente) - GVIM
Datei Editieren Werkzeuge Syntax Puffer Ansicht TeX-Suite TeX-Environments TeX-Elements TeX-Math Help Hilfe
\documentclass[
BCOR=0mm, % Bindekorrektur
DIV12,
pagesize, % passt Papiergrösse an
fontsize=12pt,
oneside,
parskip=half,version=first,
headsepline,
headings=small,version=first,
liststotoc,
captions=tablesignature,version=first,
titlepage
]{scrbook}
% oneside / twoside
% halfparskip macht halbe Zeile Raum unter Absatz
% normalheadings, bigheadings (default), smallheadings
% bibtotoc, bibtotocnumbered
% DIV12, DIV 14, Standard DIV10
% DIVcalc ausrechnen lassen
% a4paper ist default bei KOMA
1,1 Anfang
```

(b) Grafisch: gvim

Abb. 6.3.: Textbasierte und grafische Editoren

Eine interessante Szene, in der es um die impliziten Wertungen geht, die mit verschiedenen Editoren verbunden sind, spielte sich in der von mir beobachteten IT-Abteilung ab. Zwei Personen sitzen zusammen, A25 bekommt von A30 die auszuführenden Schritte diktiert:

A25 soll eine Datei editieren, A30 sagt so etwas wie „vi blabla“, A25 tippt „mcedit blabla“.⁸ A30 spielt darauf an, dass A25 immer *mcedit* benutzt, nicht *vi*. Er weist darauf

⁸Man öffnet eine Datei mit einem bestimmten Editor, indem man den Namen des Editors, gefolgt vom Namen der Datei, eingibt.

hin, dass die Kollegin A31 auf einem der Server den Befehl ‚*mcedit*‘ umdefiniert hat, so dass *vi* aufgeht. A25 sagt, er hätte das wieder geändert. Mir ist die Situation wichtig, weil *vi* in meinen Augen das Image hat, der Editor der Profis zu sein. Das ist aber unausgesprochen“ (Fir 2008e).

Dieses kurze Intermezzo interpretiere ich wie folgt: A30 weiß, dass A25 nicht mit *vi* arbeitet. Anstatt ihn einfach aufzufordern, die Datei zu bearbeiten, neckt er ihn, indem er explizit den Editor benennt, den A25 seiner Ansicht nach verwenden soll, aber eben nicht verwendet. A30 erwähnt des Weiteren, dass in einer früheren Situation eine andere Mitarbeiterin A25 auch schon mit der Nutzung von *mcedit* aufgezogen hatte. Sie hatte aus Spaß auf dem Server *mcedit* auf *vi* verlinkt, was sich so auswirkte, dass beim Aufruf des Editors *mcedit* immer *vi* geöffnet wurde.

Im späteren Gespräch über diese Situation erklärten mir A30, A25 und ein weiterer Mitarbeiter, dass *mcedit* bzw. *Midnight Commander* ein freier Klon des *Norton Commanders* unter → *DOS* sei, also eine Anlehnung an ein altes *DOS*- und *MSDOS*-Programm. A25 sei der einzige unter ihnen, der diesen Editor verwende. A25 sagt, er würde *vi* nutzen, wenn er die ganzen Tastaturkürzel auswendig könnte, wenn er sie „sich reinprägeln“ würde. Er sei so immer darauf angewiesen, dass jemand sie ihm diktiere. „A27 bestätigt, dass er sich anfangs dazu zwingen musste, *vi* zu benutzen. Er habe gewusst, dass *vi* gut ist und es sich lohnt, ihn zu lernen. Vorteil sei, dass er auf jedem Linux installiert sei“ (Fir 2008e). Die stattfindende Distinktion impliziert somit, dass A25 den mit *vi* verbundenen Lernaufwand scheut, den die beiden anderen auf sich genommen haben. Diese „Anfangshürde“ wird auch an anderen Stellen erwähnt:

Und dann hab ich halt mal den *vim* benutzt, den fand ich am Anfang ’ne totale Katastrophe zu benutzen, und hab aber aus irgendeinem Grund den weiterbenutzt und hab mich da ’n bissl reingelesen und probiert. Und hab dann festgestellt, dass der für mich sehr... sehr easy und einfach zu bedienen geht. Wenn man einmal sozusagen diese Anfangshürde überwunden hat, dann kommt man schnell damit zurecht. Und dann is es halt irgendwie ’n Gewohnheitseffekt gewesen (Int 2008a).

Die Website *ubuntusers.de*, die einsteigerfreundliche Erklärungen zu sehr vielen Programmen enthält, beginnt gleich im ersten Satz mit der Information, dass der Editor zunächst abschreckend wirkt:

„Viele Benutzer schrecken vor dem sehr praktischen Editor Vim wegen seiner etwas sperrigen Bedienung zurück. Wer sich aber erst einmal mit den verschiedenen Be-

triebsmodi vertraut gemacht hat, mag auf die hilfreichen Dienste des schlanken⁹ Editors nicht mehr verzichten. (...) Zugegeben: es gibt Editoren, die wesentlich leichter zu bedienen sind (...). Man sollte sich nicht von der anfangs komplizierten Bedienung abschrecken lassen. Mit einer Handvoll Befehlen kommt man sehr gut klar (z. B. i, ESC, :w, :wq, dd).¹⁰

vim ist also ein Editor, der ausschließlich über Tastaturkürzel zu bedienen ist. Er verfügt über weitaus mehr als eine „Handvoll“ davon. Gerade dies macht ihn für Intensivnutzer_innen interessant, da er damit mehr Funktionen mitbringt als andere Editoren. Der oben Interviewte spricht von einem „Gewohnheitseffekt“, der erst erarbeitet werden muss, da die Shortcuts nicht die in anderen Programmen üblichen sind. Dieses Erlernen von *vim*, das zu einer nicht mehr reflektierten Eingabe von Shortcuts führt, stellt einen Prozess der Inkorporierung von Wissen dar, der deutlich körperbezogen ist: Wer *vim* beherrscht, dessen Hände tippen zum Speichern nicht das konventionelle *Strg+S*, sondern *:w*.¹¹ Pierre Bourdieu hat für die Internalisierung von Gesten und Körperhaltung den Begriff der körperlichen Hexis bemüht, ein Konzept, auf das sich an dieser Stelle ein kurzer Blick lohnt. Er beschreibt die körperliche Hexis als „die realisierte, *einverleibte*, zur dauerhaften Disposition, zur stabilen Art und Weise der Körperhaltung, des Redens, Gehens und damit des *Fühlens* und *Denkens* gewordene politische Mythologie“ (Bourdieu 1999: 129). Damit meint er, dass die Befolgung eigentlich willkürlicher kultureller Grundprinzipien in den Körper integriert wird (z. B. das Messer nicht in der linken Hand zu halten). Solche scheinbar kleinen Details evozierten stets den gesamten gesellschaftlichen Kontext, aus dem sie kommen; eine kleine Geste sei immer auch eine „Unterwerfung unter die bestehende Ordnung“. Wer Höflichkeitsregeln befolge, müsse die Gesamtheit einer politischen Ordnung anerkennen und stillschweigend meistern (vgl. ebd.). „Der menschliche Körper führt so gesehen immer schon das Soziale, die Gesellschaft, die Kultur mit“ (Göhlich/Zirfas 2007: 120), und zwar in seinem inkorporierten Habitus. Robert Schmidt, der Programmierende bei der Arbeit teilnehmend beobachtet hat, konstatiert ebenfalls die Körperlichkeit des Arbeitens am Rechner: „Die Arbeit am Code zeigt sich in dieser mikroanalytischen Perspektive also nicht als kognitive Analyse- und Konstruktionsleistung¹²,

⁹„Schlank“ bezieht sich hier darauf, dass das Programm keine großen Rechenkapazitäten in Anspruch nimmt, eine sehr häufig erwähnte Eigenschaft von Software. „Schlanke“ Programme können auch auf älteren Computern noch eingesetzt werden, ohne den Arbeitsprozess stark zu verlangsamen. „Schlank“ ist in dem Zusammenhang eine ästhetische Wertung.

¹⁰<http://wiki.ubuntuusers.de/Vim>, Abruf: 21.12.2010

¹¹Zur Körperlichkeit von Lernprozessen vgl. den Überblick von Göhlich/Zirfas (2007: 117-125).

¹²Der kognitive Teil des Arbeitsprozesses ist für die teilnehmende Beobachtung ja nicht sichtbar.

sondern als ein praktischer Vollzug, in dem ausgeführte und vorgestellte Körperbewegungen eine wichtige Rolle spielen“ (Schmidt 2008: 290). Die Ordnung, das Grundprinzip, das *vim*-Nutzer_innen in ihrem Aneignungsprozess des Editors habituell mit einverleiben, ist die Effizienz: Wie lässt sich eine Aufgabe möglichst schnell erledigen? Wie kann mit einem schnellen Griff ein ganzer Absatz gelöscht werden, wie alle Leerzeilen? Der Körper wird Teil dieses Prozesses, indem ihm die spezielle Bedienung dieses einen Programms eingeschrieben wird, eine Konditionierung, die über das Erlernen sonstigen körperbezogenen Wissens (wie des Zehn-Finger-Schreiben oder des quasi-automatischen Abspulens programmübergreifender Shortcuts) hinausgeht. Bei versierten Nutzer_innen eines Editors wird diese Inkorporierung sichtbar: Sie arbeiten extrem schnell und routiniert mit „dem Editor ihrer Wahl“ und passen ihn teilweise speziell auf ihre Bedürfnisse und Geschmäcker an.¹³

Das Spezialwissen über den Editor wird im gesamten Forschungsmaterial sehr oft eingebracht. In diesem Protokollausschnitt aus der LUG richtet A1 für A24 eine Firewall ein:

A1 hat seinen *vim* besonders eingerichtet, die Farben verändert: schwarzer Hintergrund, grüne Schrift, rote → *Kommentare* und gelbe und blaue Syntax-Teile. (...) A1 baut die Änderungen teilweise so schnell ein, dass ich gar nicht folgen kann bzw. die Funktionen von *vim* nicht kenne, die er verwendet. A24 sagt, er rutscht mal ein Stück und geht kurz raus (langweilig? Kann er auch nicht folgen?) (LUG 2007b).

Abgesehen von der Firewall selbst, die A1 immer wieder erläutert, kommt hier nebenbei auch sein Spezialwissen über *vim* zum Einsatz, das an keiner Stelle thematisiert wird. Es ist nicht Gegenstand des Wissenstransfers, wirkt aber differenzierend, weil die Arbeitsschritte für mich (und möglicherweise auch für A24) nicht nachvollziehbar sind. In anderen Situationen entstand eine Ablenkung vom eigentlichen Thema, weil die Benutzung von *vim* ein Problem darstellte oder jemand ein Detail in der Nutzung beobachtet hatte, das er_sie kurz eruieren wollte.

A46 startet den Mailserver neu und erhält eine Fehlermeldung, in der steht, in welcher Zeile der Konfig es einen Fehler gibt. A46 öffnet die Datei und springt direkt zu der Zeile. A39 stellt eine Zwischenfrage an A46, nämlich, wie er so schnell zu der Zeile komme. A46 und A47 erläutern, man könne einfach die Zahl eingeben. A39 will das

¹³Diese persönlichen Einstellungen werden in einer einzigen Textdatei, der *.vimrc*, gespeichert. Menschen, die viel mit dem Editor arbeiten und sich dabei sehr an ihre persönlichen Einstellungen gewöhnt haben, kopieren, wenn sie an fremden Rechnern arbeiten, ihre eigene *.vimrc* auf diesen Rechner, um in ihrer gewohnten „Arbeitsumgebung“ zu arbeiten.

gleich ausprobieren, schließt den *vi* nochmal, öffnet ihn wieder und springt zu der fraglichen Zeile (LUG 2009d).

Eigentlich beschäftigt sich die Gruppe mit der Mailserverkonfiguration. A46s routinierter Umgang mit dem Editor fällt A39 quasi nebenbei auf. Er hakt dort ein, um etwas zu lernen, was ihm selbst nützlich sein könnte, nämlich direkt zu einer fehlerhaften Zeile in einer Konfigurationsdatei springen zu können. Dies ist ein Beispiel dafür, dass neben einem gemeinsamen Lerngegenstand auch anderes Wissen transferiert wird, das auf dem Lösungsweg zum Einsatz kommt. Der praktische Sinn der Beteiligten beinhaltet, dass sie bei der Beobachtung praktischer Details in der Bedienung eines Programms nachfragen, dass sie also vorausahnen, welches Wissen sie selbst einmal gebrauchen können oder die Erfahrung haben, dass das Gesehene bei ihnen selbst länger gedauert hätte oder umständlicher gewesen wäre.

In den nächsten beiden Ausschnitten wird deutlich, dass *vim*-Nutzer_innen Probleme mit grundlegenden Funktionen wie dem Einfügen von kopiertem Text haben. Der jeweilige Kontext zeigt einmal mehr, wie themenabhängig das technische Expertenwissen ist und wie es unterschiedlich eingesetzt werden kann. Die erste Situation stammt aus der IT-Abteilung:

Beim Kopieren und Einfügen von Text in *vim* passiert, was mich schon oft genervt hat: die Ordnung, in der das Textstück vorher war, geht verloren und der eingefügte Text ist völlig unübersichtlich. Ich sage genervt: „Das ist mal wieder *vim*.“ A36 guckt rüber und sagt, ich solle `:set paste` eingeben. Tatsächlich beseitigt das das Problem. Ich frage, ob das dauerhaft gespeichert wird, er sagt, das müsse ich in eine Datei schreiben. Ich sage: „In die *.vimrc*.“ Er: „Ja.“ Er sagt, er hätte sich mit wenigem so intensiv beschäftigt, aber mit *vim* schon. Und dass er bemerkenswert finde, dass ich damit überhaupt arbeite. Ich frage leider nicht nach - immerhin arbeitet (glaub ich) die ganze IT außer A25 damit. Warum hat er das von mir nicht erwartet? (Fir 2008d).

A36 zeigt mir, was ich einstellen muss, damit das Einfügen von Text so funktioniert, wie ich es aus anderen Programmen gewohnt bin. Er betont, wie intensiv er sich mit *vim* beschäftigt hat und gibt zu erkennen, dass er von mir nicht erwartet hat, dass ich ihn verwende. Wie er zu dieser Annahme kommt, bleibt offen. Mögliche Klassifikationen sind jedoch: Der Kollege ist neu; ich sitze in der ihm bekannten Arbeitssituation als einzige Aushilfe und als einzige Frau unter männlichen, fest angestellten Softwareentwicklern. Ich bin dort die einzige ohne professionellen IT-Hintergrund. Er hatte mich bis dahin hauptsächlich

bei *einer* Tätigkeit gesehen, nämlich beim Verfassen von Software-Dokumentationen für Anwender_innen. Die verschiedenen Tätigkeiten innerhalb des Softwareentwicklungsprozesses haben Konnotationen, nicht nur innerhalb von Open-Source-Projekten. Übersetzung und Dokumentation fallen unter „die weichen Sachen“ (LUG 2009l) und werden oft gegenüber dem Programmieren abgewertet (vgl. Jung 2006: 241 f.).¹⁴ Die Situation ist damit durch Kategorien wie Arbeitsteilung, Wertung von Aufgaben, formale Qualifikation und Gender strukturiert, die sich (einzeln oder gemeinsam) darauf auswirken können, welches Wissen und welche Routinen von Personen erwartet werden.

Die zweite Szene, in der dasselbe Problem beim Einfügen von Text bemerkt wird, spielt sich in der LPI-Lerngruppe der LUG ab. Hier sitzen fünf Männer zusammen und bereiten sich auf die Zertifizierungsprüfung des → *Linux Professional Institute* vor. Das „eigentliche“ Thema des Treffens ist die Konfiguration des → *Webservers Apache*.

A2 findet ein paar Zeilen, die er in die Datei einfügen möchte. Auch er hat Probleme bei der Bedienung des *vi*. Er fügt den kopierten Text ein und jede Zeile ist um einen Tab weiter hinten eingefügt als die vorherige. Er sagt, dass es so nicht aussehen sollte und beginnt einzelne Zeichen zu löschen. Ich rate ihm, das Einfügen nochmal rückgängig zu machen und `:set paste` zu setzen. Er kennt das nicht und fragt wie. Ich diktiere ihm: „Zuerst ‚u‘, dann ‚Doppelpunkt set paste‘.“ (...) A100 fragt, wie er rückgängig machen könne, ich sage, mit ‚u‘. Dann klappt es. A2 fügt den Text ein und er erscheint so, wie er auf der Website auch ausgesehen hatte. A2 und A100 freuen sich über diesen Weg und loben den Tipp. (Ich notiere mir im Kopf die Frage, warum sie eigentlich einen Editor verwenden, der ihnen Probleme bereitet. Ob sie sich da explizit irgendwann drauf geeinigt haben? Oder ob das Konsens war, den *vi* zu benutzen?) (LUG 2009c).

Die anderen Beteiligten sind aus meiner Perspektive professionelle Systemadministratoren, von denen zumindest einige ihr Geld mit dieser Arbeit verdienen. Sie bereiten sich bereits auf das LPI-Niveau 202 vor¹⁵, zu deren Inhalten Mail- und Webserverkonfiguration gehören. Mein persönlicher Maßstab dafür ist, dass es sich um „Profis“ handelt. Dennoch sind ihnen grundlegende Schritte in *vim* nicht bekannt, wie sie z. B. einen Schritt rückgängig machen können oder eben kopierten Text aus der Zwischenablage einfügen

¹⁴Zu symbolischen Kämpfen um die Bewertung von Praktiken des Programmierens und Programmieransätze in der kommerziellen Softwareentwicklung vgl. Schmidt (2008).

¹⁵Nach Bestehen der ersten zwei Prüfungen (101 und 102) bekommen die Kandidat_innen das erste Zertifikat, nach weitere zwei Prüfungen (201 und 202) das nächste, für das dritte Zertifikat werden sechs Prüfungen nötig sein, die aber noch in der Eintwicklungsphase sind.

können. Die Themengebiete, in denen man sich Spezialwissen aneignen kann, sind so zahlreich, dass immer Wissenslücken bleiben. Um so deutlicher wird hier, wie der Arbeitskollege A36 an einem kleinen Thema eine Differenz zwischen sich und mir konstruiert, denn viel fortgeschrittenere oder professionelle Linux-Systemadministrator_innen haben dieselben oder vergleichbare Wissenslücken, je nachdem, worauf sie sich in ihren individuellen Lernprozessen konzentriert haben.

Die Frage, die ich mir in der Beobachtungssituation gestellt hatte, warum die Prüfungskandidaten sich ausgerechnet dieses Editors bedienen, beantwortet sich mit einem Blick in die LPI-Prüfungsziele, zu denen die grundlegende Beherrschung von *vim* zählt (vgl. Heinlein 2009: 123 ff.). Im Vorbereitungsbuch zur ersten Prüfung sind Beispielfragen, die verdeutlichen können, wie weit man es mit der *vim*-Expertise treiben kann:

„Sie möchten in einem Text eine Zeile kopieren und 3 Zeilen später wieder einfügen. Mit welcher Kommandofolge können Sie das erreichen?

a) ckkp, b) yy3jp, c) c3p, d) C nnn Y, e) yjjp“ (Heinlein 2009: 126)

Hier wird die Kombination mehrerer *vim*-Kommandos (Kopieren, Springen, Einfügen) abgefragt, und zwar in einer Komplexität, die ich in der tatsächlichen Praxis nie beobachtet habe. Bis auf wenige Szenen, die ausgesprochene Routine ausdrückten (z. B. A1 in der weiter oben genannten Situation), waren meistens schon grundlegende Kommandos unbekannt. Ein letztes Beispiel dafür, wie das Wissen um diverse Tastaturkürzel oder Einstellungen in der Praxis zur Herstellung von Unterschieden eingesetzt wird: In einer Vortragssituation in der LUG riefen Zuhörende dem Referenten einzelne *vim*-Shortcuts laut zu.

A10 ist erst wenige Minuten anwesend, da mischt er sich ein: Der Referent A41 hat gerade eine Datei im *vim* geöffnet, deren auskommentierte Zeilen in dunkelblau auf schwarzen Hintergrund an die Wand geworfen werden und sehr schlecht zu lesen sind. A10 wirft in den Raum: „Mal 'groß V' drücken, dann wird's gehighlightet!“ A41 versteht zuerst nicht, was gemeint ist, markiert dann aber schließlich den auskommentierten Text, so dass er sich besser vom Hintergrund abhebt. (...) A41 öffnet erneut eine Datei im *vim*, die auf dem schwarzen Hintergrund schlecht zu lesen ist. (Ich frage mich, warum überhaupt dermaßen viele Leute einen schwarzen Hintergrund im Texteditor einstellen. Dieses Problem hab ich doch schon oft gesehen, gerade mit Beamer wird es noch schwieriger.) Der Sitznachbar von A1 sagt: „Einfach :set bg=dark“ (LUG 2008a).

Auch in diesem Vortrag ist *vim* nicht das eigentliche Thema, sondern das Detailwissen über den Editor wird nebenbei aus dem Publikum eingeworfen. A41 hat sich zu seinem Vortragsthema eine Expertise erarbeitet, wird aber von zwei Zuhörern mit anderen Unzulänglichkeiten konfrontiert, dass z. B. das Publikum schlecht lesen kann, was an die Wand projiziert wird. Die beiden heben ihr Spezialwissen über Tastaturkürzel demonstrativ hervor, was eine ähnliche Wirkung hat wie ein Co-Referat, das Wissenslücken des Referenten aufdeckt oder sogar zur Schau stellt. Dies ist ein weiteres Beispiel für eine Selbstinszenierung vor Publikum, die hier wie eine Konkurrenz um kleinteiliges Detailwissen wirkt. Sie hätten auch auf das Problem hinweisen können und damit demonstrieren, dass sie auf die Problemlösungskompetenz des Referenten vertrauen.

Zum zweiten sehr verbreiteten Editor, *GNU Emacs*, habe ich keine Beobachtungen gemacht; er war in meinen Erhebungskontexten nicht in Benutzung. Wohl aber wurde teilweise auf ihn oder auf den so genannten „Editor War“ angespielt. Dieser „Krieg der Editoren“ ist eins von mehreren Beispielen für grundsätzliche Streits über die beste Software.¹⁶ Auf die Frage „*vi* oder *Emacs*“ wird immer wieder rekuriert, meistens auf ironische Art und Weise. Einige der Beobachteten grenzten sich z. B. von *Emacs* ab: „Da haben sie ein tolles Betriebssystem geschrieben, leider haben sie den Editor vergessen“ (Fir 2008e). Der Spruch zielt darauf ab, dass *Emacs* über dermaßen viele Nutzungsmöglichkeiten verfügt, dass darüber glatt die zentrale Funktion des Editierens übersehen werden könnte. Kelty geht in seiner Ethnografie sogar so weit, Streits wie den „Editor War“ als Allegorie eines Religions- oder Glaubenskrieg zu bezeichnen.¹⁷

Die Szenen, in denen es um die Verwendung eines bestimmten textbasierten Editors ging, stehen in diesem Abschnitt als Beispiele für eine grundsätzliche und prägende Distinktionspraxis von FLOSS-Communities. In dieser Praxis geht es nicht allein darum, *ob* jemand freie Software benutzt, sondern auch darum, welche Software das genau ist, da viele Programme ein bestimmtes, teilweise mit Wertungen behaftetes Image haben. Die gewählten Materialbeispiele bezogen sich hauptsächlich auf den textbasierten Editor *vim*, dem — wie auf *ubuntuusers.de* kolportiert — der Ruf vorausseilt, er sei abschreckend, weil

¹⁶Die anderen dieser meist scherzhaften Grundsatzdiskussionen betreffen die Wahl der grafischen Benutzeroberfläche (KDE oder Gnome) oder der Linux-Distribution, die Wahl von Programmiersprachen oder ob im Programmcode Einrückungen mit Leerzeichen oder Tabulatoren gesetzt werden.

¹⁷„Such stories describe a piece of cynicism: they describe a technical war of details between two pieces of software that accomplish the same thing through different means, so devotion to one or the other is seen as a kind of arbitrary theological commitment, at once reliant on a pure rationality and requiring aesthetic or political judgement“ (Kelty 2008b: 67).

schwierig zu benutzen. Die Ausschnitte haben jedoch auch gezeigt, dass selbst unter denen, die die „Anfangshürde“ bewältigt haben, die Distinktion weitergeht, nämlich in den Details der Handhabung. Im Umgang mit dem Wissen um diese Feinheiten gibt es diverse Facetten: Die Nutzer_innen beherrschen sie nicht nur in sehr verschiedenem Umfang, sondern legen auch unterschiedlich starke Betonung auf dieses Wissen. Es *kann* situativ zur Selbstverortung oder Zurschaustellung von Expertise eingesetzt werden. Zudem nutzen auch Menschen *vim*, die den Funktionsumfang des Editors bei Weitem nicht ausschöpfen. Dies deutet darauf hin, dass sie mit *vim* eine geschmackliche Entscheidung getroffen haben, dass sie verstanden haben, dass dies Prestige einbringt. Sie stellen sich nicht die Frage, welche Anforderungen ihre Software erfüllen muss, so dass ihre Entscheidung für das komplizierte Werkzeug auch einen Beigeschmack von Technikfetischismus bekommt.¹⁸ Was Bourdieu über Abgrenzung per Geschmack konstatiert (vgl. Bourdieu 1987: 111), mag in ähnlicher Form für die Distinktion über technisches Detailwissen gelten: Es geht oft eher um eine Absetzung von Nahestehenden, von unmittelbaren Konkurrent_innen als von denen, die sich gar nicht mit freier Software beschäftigen. Computernutzer_innen, die sich nicht mit *vim* auskennen, fehlt der praktische Sinn, der sie die Optimierungsmöglichkeiten oder Effizienzzugewinne dieses Editors erkennen lassen würde. Nur wer diesen Sinn hat, antizipiert, dass der Lernaufwand sich lohnt. Der Ruf von *vim*, seine ausgefeilten Konfigurations- und Bedienungsmöglichkeiten spricht also einerseits die Wahrnehmungsschemata derjenigen an, die ihn zu schätzen wissen. Andererseits wertet dieser Ruf gleichzeitig diejenigen ab, die sich nicht an diese Software herantrauen oder das Unterscheidungsvermögen dafür nicht haben.

Die Verwendung textbasierter Programme, die für Teile von FLOSS-Communities eine inkorporierte Selbstverständlichkeit ist, wirkt differenzierend gegenüber denen, die keine textbasierten Programme verwenden. Textbasiertes Arbeiten symbolisiert in vielen Szenen den Habitus von Systemadministrator_innen oder Programmierer_innen. Darüber hinaus konnte hier gezeigt werden, dass die, die viele Fragmente dieses Habitus inkorporiert haben, viele weitere symbolische Abstufungen schaffen, gegenüber anderen, die auch textbasiert arbeiten. Diese graduelle Disktinktion kann stark mit technischer Kompetenz aufgeladen und inszeniert werden. Zudem werden der Software Wertungen zugemessen, die in den Nutzungspraktiken Erwähnung finden. Dabei tragen nicht nur die

¹⁸Margolis/Fisher (2002: 49 ff.) haben gezeigt, dass eine Beschäftigung mit Computern, die Selbstzweck ist, unter Männern sehr viel verbreiteter ist, als unter Frauen. 44 Prozent der befragten Studentinnen gaben an, dass sie Zwecke oder Einsatzfelder vor Augen hätten, für die sie ein Informatikstudium aufgenommen hätten. Dies teilten nur 9 Prozent der männlichen Befragten (vgl. ebd. 53).

Betriebssysteme selbst die Images von freier oder proprietärer Software, sondern auch die unterschiedlichen Linux-Distributionen, die grafischen Desktopumgebungen und einzelne Programme werden mit Konnotationen versehen. Diese Konnotationen greifen Kriterien wie Benutzerfreundlichkeit, Effizienz, grafische Aufmachung und Umfang der Funktionalitäten auf und können situativ abgerufen werden. Auf diesen Wissensschatz greifen nicht nur diejenigen zurück, die sich inszenieren möchten, sondern auch die, die jemandem eine passende Software empfehlen möchten. Auf mehr oder weniger subtile Weise machen die Beteiligten so Unterschiede auf, die teilweise wertend sind: Wer arbeitet textbasiert? Welches genaue Programm kommt dabei zum Einsatz? Wer beherrscht ein textbasiertes Programm bis in welches Detail? Diese kleinteiligen Distinktionspraktiken zeigen auch, dass ein einheitlicher Habitus der Techniknutzung beim Einsatz textbasierter Programme nicht beobachtbar ist. (Dies wurde am Umgang mit dem Editor *vim* gezeigt, der selbst bei Systemadministrator_innen in der Praxis öfter scheitert.) Aus diesem Grund kann — ähnlich wie im fünften Kapitel — nicht von *einem* Habitus gesprochen werden, sondern höchstens von Habitusfragmenten, die die Beteiligten zu unterschiedlichen Anteilen internalisiert haben.

6.2. Ästhetik und Kreativität

In den Vorüberlegungen in Abschnitt 3.1.2 wurde bereits auf Bourdieus Analyse der „gesellschaftlichen Urteilskraft“ am Beispiel von Geschmack verwiesen. Geschmack wird dort als Grundlage des Klassifizierens anderer und der Selbstverortung begriffen (vgl. Bourdieu 1987: 104). Es wurde bereits stellenweise gezeigt, dass geschmackliche Äußerungen und ästhetische Vorstellungen in FLOSS-Communities Distinktion hervorbringen, z. B. an der Ästhetik, die verschiedenen Lösungswegen für ein Problem zugeschrieben wird. Auch das humorvolle Spielen mit technisierter Sprache spricht einen bestimmten Sinn für Ästhetik an. Mit Bourdieu ist Ästhetik keine „akademisch-philosophische Reflexion über das Schöne und deren normativ-theoretische[...] Grundlagen“ (Diaz-Bone 2002: 43). Er rekonstruiert kollektive Ästhetiken vielmehr, um soziale Klassen und damit gesellschaftliche Strukturen zu analysieren. Der Sinn für Ästhetik ist für ihn damit sozial verortet (vgl. ebd). In diesem Abschnitt werden Ästhetik und Kreativität vertieft, die in Praktiken der Techniknutzung zum Ausdruck kommen. Dabei geht es zuerst um die kreative Umnutzung von Technologien, danach um die Einschreibung von ästhetischen Vorstellungen und persönlichen Einstellungen in Technologien.

6.2.1. Kreative Umnutzung von Hard- und Software

Im Umgang mit Hardware ist sehr häufig zu beobachten, dass Community-Mitglieder ihre Notebooks mit Aufklebern verzieren, die Szene-Maskottchen oder netz- bzw. technologiepolitische Äußerungen abbilden. Dies ist kein Spezifikum dieses Feldes, dennoch können die Aufkleber als Symbole von Mitgliedschaft gelesen werden, denn es gibt sie in erster Linie auf Community-Events oder direkt bei den in den entsprechenden Projekten Aktiven. Sie bilden zumindest zu einem gewissen Grad ab, an welchen Veranstaltungen jemand teilgenommen hat bzw. wie lange jemand schon dabei ist. Die Umgestaltung von Hardware geht teilweise weit darüber hinaus. Einige bauen sich spezielle Teile in ihre Rechner ein (z. B. WLAN-Karten mit speziellen, von Linux gut unterstützten Chipsätzen), manche bauen sich aus Hardware oder für Hardware andere Gebrauchsgegenstände oder Möbel.

A44 hat den Server mitgebracht, der in einem selbst gebauten Holzgehäuse steckt, das er für sein Regal passend gebaut hat. Auch A1 ist recht beeindruckt davon (LUG 2008j).

Aus einem alten Rechner hat A105 sich selbst einen Couchtisch gebaut, mit Beinen auf Rädern. Vorne dran kleben noch die originale → *IP-Adresse* und der Rechnernamen ,bienchen'. (...) Es sind ca. sieben Computer in der ganzen Wohnung, nicht alle in Benutzung. Teils stehen Pflanzen darauf, in der Ecke vom Wohnzimmer. A105 sagt, da fehle noch ein Möbelstück (dabei finde ich, dass die Einrichtung mit den Rechnern die einzige individuelle Note an der Wohnung ist). Die Musikanlage ist im Wohnzimmer in einem Serverschrank (A10 2008).

Die hier Beobachteten haben einen Umgang mit Hardware, der über die pragmatische Computernutzung weit hinaus geht: Durch Umbauten sind ihre personalisierten Artefakte entstanden. A44s Gehäuse ist dabei eine Konstruktion, die seinem Server im Regal einen passenden Platz schafft, während A105 ausgediente Hardware zu Möbeln umfunktioniert hat. Damit zeigt er seine Verbundenheit mit dem technischen Artefakt über seine Nutzung hinaus.

Auch bei der Nutzung von Software ist zu sehen, dass viele Linux-Nutzer_innen technologische Artefakte für andere Zwecke als für die vorgesehenen umnutzen. Dies soll am Beispiel eines Rituals verdeutlicht werden, das alljährlich auf einem großen Community-Event stattfindet: Der Hacking-Contest wird von einer Firma veranstaltet, die Software im

Bereich Internetsicherheit entwickelt. Sie sponsert auch die Preise. Das Ritual wird zunächst kurz erläutert, bevor es aus der Perspektive des kreativen Umgangs mit Software beleuchtet wird.¹⁹ In Kapitel 5 wurde bereits auf das Thema Konkurrenz um Lösungswege eingegangen; der Hacking-Contest ist eine ähnliche, aber durch Spielregeln formalisierte Situation. Bereits die Ankündigung reflektiert dies:

„Kennern der IT-Security ist lange bekannt, dass Open-Source-Betriebssysteme als die sichersten Systeme gelten. Die offenen Quellen lassen im Gegensatz zu proprietären Ansätzen die Überprüfung der Sicherheitsfunktionen durch Experten zu - mögliche Schwachstellen können so schnell lokalisiert und behoben werden. Doch ein System ist weiterhin nur so gut, wie der Administrator, der es bedient und wartet ...

... Es ist daher wieder Zeit, dass sich die Besten der Besten (oder die, die sich dafür halten) untereinander messen und ihre Fähigkeiten unter Beweis stellen.²⁰

Der Ablauf des Hacking-Contest ist folgender: Einzelpersonen oder kleine Teams treten gegeneinander an. Sie messen sich im Einbauen und Ausnutzen von Sicherheitslücken, so genannten „Exploits“. Einige Teams melden sich vorab zum Hacking-Contest an, es besteht aber auch die Möglichkeit, sich vor Ort spontan zu melden. Auf einer Bühne sitzen in jeder Runde zwei Teams. Ob alle Teams gegeneinander antreten oder ob nur um die ersten drei Plätze gespielt wird, wird unterschiedlich gehandhabt. Jede solche Runde besteht aus drei fünfzehnminütigen Teilen: Die Teams erhalten Linux-Rechner mit einer ihnen vorher nicht bekannten → *Distribution*. Das Publikum kann per Beamer mitverfolgen, was die Teams machen, die sich selbst nicht zur Projektionsfläche umdrehen dürfen. Sie bauen im ersten Schritt so genannte → *Backdoors* ein, also Lücken, durch die sie auch als „unprivilegierte“ User_innen, die nur eingeschränkte Zugriffsrechte haben, root-Rechte erlangen könnten. Nach fünfzehn Minuten werden die Rechner getauscht. Jetzt begibt jedes Team sich auf die Suche nach den eingebauten Backdoors des gegnerischen Teams. Für gefundene Sicherheitslücken vergibt die Jury, die aus den Organisator_innen besteht, Punkte. Im dritten Teil werden wiederum die Rechner getauscht. Die Teams demonstrieren jetzt, welche „Exploits“ ihre Gegner_innen nicht gefunden haben und wie sie sie genau nutzen können. Auch hierfür gibt es Punkte. Auf diese Weise werden die ersten

¹⁹Zum Hacking-Contest liegt mir relativ wenig Datenmaterial vor: Die teilnehmende Beobachtung war nur vom Publikum aus möglich. Vom Publikum aus konnte weniger die Kooperation der Teams, dafür eher die Gesamtsituation beobachtet werden. Die Darstellung bezieht sich auf die Beobachtung zweier Wettbewerbe. Im Anschluss habe ich einen Teilnehmer zur Organisation seines Teams und der Vorbereitung auf den Wettbewerb befragt. Weitere Fragen konnte ich ihm per E-Mail im Nachgang stellen.

²⁰<http://www.linuxtag.org/2009/de/program/hacking-contest.html>, Hervorhebung im Original

drei Plätze ermittelt und mit teilweise attraktiven Preisen (z. B. Spielekonsolen oder Netbooks) belohnt.²¹

Die Gewinnerteams sind auf konkrete Szenarien vorbereitet. Sie sind mindestens zu zweit: Eine Person liest von einem Zettel vor, was die andere eingeben soll, die sich auf das schnelle Tippen konzentriert. Es ist laut der Regeln explizit gestattet, Notizen *auf Papier* zu Rate zu ziehen. Das Team, dessen Notizen ich eingesehen habe, hatte diese Szenarien distributionsspezifisch angelegt.²² Zunächst verbringen die Teams einige Minuten damit, ihre eigenen Spuren zu verwischen, die normalerweise auf einem Linux-Rechner dokumentiert werden (z. B. welche Befehle zuletzt ausgeführt wurden). Dann erst werden die „Backdoors“ eingebaut. Das Publikum quittiert oft die Aktionen der Teams mit Applaus oder Gelächter, versteht also, welche Ideen vorne auf der Bühne umgesetzt werden — während ich selbst nicht verstand, was dort gemacht wird. Ein Teilnehmer verdeutlicht, was Hacking für ihn bedeutet, und gibt ein Beispiel für eine solche „Backdoor“:

S. M.: Was heißt für Dich überhaupt ‚Hacking‘?

A109: Hacking bedeutet für mich, kreativ mit der Technik umzugehen. Dazu gehört z. B., Tools für einen ganz anderen Zweck zu benutzen als den eigentlich vorgesehenen. Es hat viel mit Experimentierfreude und Spieltrieb zu tun.

S. M.: Kannst Du dafür ein Beispiel nennen? Wie kann ich mir das vorstellen?

²¹Zu den genauen Regeln siehe <http://www.linuxtag.org/2009/de/program/hacking-contest.html> oder <http://www.linuxtag.org/2010/de/program/hacking-contest.html>, Abruf: 22.02.2011

²²Linux-Distributionen haben teilweise unterschiedliche Funktionsweisen. Die Gruppe hatte sich also überlegt, welche Backdoors sie wie in verschiedene Distributionen einbauen würde (vgl. Int 2009).

A109: Ein Beispiel ist das → *ssh-Banner*. Der → *ssh-Daemon* kann dem → *Client* beim Login (vor der Passwortabfrage) einen Bannertext übermitteln. (...) Das *ssh-Banner* bietet also eine Möglichkeit, Informationen an nicht eingeloggte User rauszugeben. Jetzt stellt sich die Frage: Wofür kann man diesen Mechanismus noch verwenden bzw. zweckentfremden? Man könnte z. B. Zugangsdaten (username/password) für eben diesen *ssh*-Zugang damit rausrücken, untergebracht in einem unverdächtig aussehenden Bannertext. Oder man schreibt einfach kackfrech den *private key* eines *ssh*-Key-Paares rein, um Zeit zu sparen — so haben wir es auf dem Hacking-Contest gemacht. Das Key-Paar gehört natürlich zum root-User :) Praktisch bei Key-basiertem *ssh*-Login ist, dass man das Passwort des betreffenden Users nicht kennen muss. Es reicht, im Besitz des *private key* zu sein (Int 2009).

A109 erläutert, dass für ihn der Hacking-Contest ein Spielen und Experimentieren mit den Programmen ist. Es ist ein Ausdruck von Kreativität, Programme zu zweckentfremden. Der Clou ist für ihn daran, dass mit nur den „Bordmitteln“ einer Desktop-Installation hantiert werden darf; die standardmäßig vorhandenen Programme sind also schlau einzusetzen.

Sie stellen also keine ‚real life hacking‘-Situation dar, in der man sich Zeit nehmen kann und mit C-Compiler, Debugger und Disassembler bewaffnet vorhandene Binaries auf Lücken untersucht. (...) Der Reiz beim Hacking-Contest (der eigentlich ‚Backdoor-Contest‘ heißen müsste) besteht für mich gerade in der Beschränkung auf Bordmittel (ebd.).

Die Wettbewerbssituation stellt für ihn eine Mischung dar aus Unterhaltung des Publikums, Selbstdarstellung und „sportlichem Ehrgeiz“. Den Contest findet er dann spannend, wenn mehrere gut vorbereitete Teams gegeneinander antreten.²³ Ihm ist die Teilnahme wichtiger als das Gewinnen. „Mir würde es auch ausreichen, jedes Jahr wieder ein paar neue und elegante Tricks aus dem Hut zu zaubern und dadurch nicht ganz schlecht abzuschneiden“ (Int 2009). A109 zufolge erarbeiten die Teilnehmer_innen sich durch den Hacking-Contest einen „gewissen Ruf“.²⁴

²³Er hat die Erfahrung gemacht, dass unvorbereitete Teilnehmer oder Teams den Contest nicht gewinnen.

²⁴„S. M.: ‚Was ist damit verbunden - noch was anderes als die Preise?‘ A109: ‚Ruhm und Ehre und ewiges Leben :) Einen gewissen Ruf erarbeitet man sich durch den Hacking Contest schon‘“ (ebd.).

Der Hacking-Contest kann einerseits als Wettbewerb im Sinne einer differenzierenden und ordnenden Praxis interpretiert werden. Er hat andererseits viel mit dem Anerkennen von Kreativität zu tun: Gute Tricks werden als „elegant“ wahrgenommen, die Jury vergibt nach Ermessen Zusatzpunkte für „stilistische, technologische oder strategische Äußerste Gerissenheit“.²⁵ Sie erwähnt anerkennend, welche Tricks neu waren und welche bereits in einem der Vorjahre gezeigt wurden. Das Regelwerk, das dem Hacking-Contest eine Ordnung gibt, erlaubt Experimente in einem abgesteckten Rahmen, der es z. B. verbietet, den Computer neu zu starten. Ein Team ging das Risiko ein, zum Ende seiner fünfzehn Minuten den → *X-Server*, die grafische Oberfläche, neu zu starten, was nicht durch explizite Regeln abgedeckt war. Die Jury befand nach einer kurzen Konsultation, dass dieses Vorgehen in Ordnung war. Hier antizipierten die Teilnehmer_innen, dass ihr etwas riskantes Vorgehen als regelkonform akzeptiert werden würde. Sie haben den „Sinn für das Spiel“ (Bourdieu 1998b: 145), der es sie praktisch beherrschen lässt: Ein Neustart des *X-Servers* ist eben deutlich von einem Neustart des Rechners zu unterscheiden.

Distinktion findet beim Hacking-Contest einerseits unter den Teilnehmenden statt, die sich durch ihre Kreativität (und Schnelligkeit) voneinander unterscheiden. Andererseits wirkt im Publikum eine Distinktion über das Verständnis für die vorne gezeigten Tricks. Diejenigen, die verstehen, welche Lücken ausgenutzt werden und warum das kreative Ideen sind, können mit Applaus oder Gelächter reagieren, gleichzeitig lernen sie unter Umständen etwas hinzu. Wer (wie ich) gar nicht versteht, womit die Teilnehmer_innen eigentlich spielen, kann die vorgeführte Kreativität nicht wahrnehmen. Die vorgeführten „Exploits“ werden während der Auswertung von der Jury zwar erklärt, die Erläuterungen sind jedoch schon so voraussetzungsvoll, dass sie fehlendes Hintergrundwissen nicht aufwiegen.

Als Ausdruck eines kreativen Umgangs mit freier Software kann auch die in Kapitel 4 bereits gezeigte Verwendung von Humor und → *ASCII-Art* betrachtet werden. *ASCII-Art* taucht nicht nur in E-Mail-Signaturen auf, wo z. B. das Logo von *Debian-Linux* neben der Postadresse steht. Sie wird auch in Programme eingebaut. Das Programm *cowsay* besteht nur aus *ASCII-Art*.

Die Programme *cowsay* und *cowthink* geben auf der → *Kommandozeile* Kühe mit Sprech- bzw. Denkblasen aus, die die Nutzer_innen mit Inhalt füllen können. Kühe anzuzeigen

²⁵<http://www.linuxtag.org/2010/de/program/hacking-contest.html>, Abruf: 22.02.2011. „Die Punkte verdoppeln sich jeweils, wenn nachgewiesen werden kann, dass dieser Angriff über das Netzwerk hätte ausgeführt werden können“ (ebd.).



Abb. 6.4.: Humor und Ästhetik: Kühe in ASCII-Art

ist die einzige technische Funktion dieser Programme. Es gibt eine → *man page*, in der dokumentiert ist, welche Argumente dem Programm mitgegeben werden können: „-d“ für „dead“ zeigt eine tote Kuh (vgl. Abb. 6.4b), ebenso können z. B. gierige Kühe oder Kühe unter Drogeneinfluss gezeigt werden. Hier hat sich offenbar jemand „ausgelebt“, um eine Software zu schreiben, deren Schwerpunkt auf Spaß und Kreativität, nicht etwa auf Information oder Effizienz liegt.²⁶ Dass die beiden Programme auch in offiziellen Distributionen wie *Debian* enthalten sind, zeigt darüber hinaus, dass jemand keine Mühen gescheut hat, um das Programm zu verbreiten: Das Prozedere zur Aufnahme von Software in *Debian* ist sehr formalisiert (vgl. O’Neil 2009: 133) und zieht sich oft über einen längeren Zeitraum hin.²⁷

Schließlich möchte ich auf persönliche Routinen eingehen: Individuelle Routinen und Arbeitsabläufe stellen eine persönliche Note dar. Sie können gerade deshalb von anderen als Stile wahrgenommen werden, weil sie in den Situationen der praktischen Problembehandlung bzw. Wissensvermittlung sichtbar und vergleichbar werden; die Beteiligten sitzen ja nebeneinander am Rechner. Dabei geht es nicht nur um die visuelle Gestaltung der Desktopoberfläche, sondern um Einschreibungen vom persönlichem Geschmack und Routinen in Hardware und Software. Die persönlichen Einstellungen zeigen an vielen Stellen, dass Linux-Nutzer_innen sich nicht alles vorgeben lassen: Viele bestimmen selbst, wofür sie welche Tasten einsetzen und was Programme über die Standardfunktionalitäten hinaus noch tun sollen.

Ich sehe, dass A53 an etlichen Stellen mit einem haltbaren Stift etwas auf das Gehäuse geschrieben hat. Manches davon sind offensichtlich Tastenbelegungen, über einer

²⁶Für die grafische Oberfläche gibt es ähnliche Programme, z. B. einen Fisch, der Sprüche anzeigt.

²⁷Eine der LUGs war während meiner Beobachtungen dabei, ihr eigenes Softwareprojekt bei *Debian* unterzubringen, und war mit den hohen Anforderungen konfrontiert. Dieselbe Software war von der Distribution *Fedora* schon aufgenommen worden.

→ *Funktionstaste* steht z. B. „HTOP“²⁸, wobei die Buchstaben vertikal untereinander stehen (LUG 2009g).

Der hier Beobachtete hat sich seine Funktionstasten so konfiguriert, dass er Programme, die er viel nutzt, direkt über sie aufrufen kann. Dies scheinen für ihn dauerhafte Griffe zu sein, denn er hat die Funktionen auf das Gehäuse geschrieben. Andere verändern die Software, die sie nutzen, für sich selbst. Dabei werden erstens persönliche Einstellungen vorgenommen. Im Fall von textbasierten Programmen werden die Einstellungen in Textdateien abgelegt (z. B. bei A1 auf S. 157). Bei manchen Programmen ist es möglich, dort *sehr* vieles einzustellen, so gibt es z. B. Workshops, die sich nur den Einstellungen des Editors *vim* widmen. Zweitens schreiben manche Nutzer_innen zusätzlich Skripte, die mit einem Programm immer mit ausgeführt werden und die Funktionen des Programms für sie individuell erweitern.²⁹ Dies sind nicht unbedingt Änderungen, die an die Softwareprojekte als → *Feature Requests* zurückgegeben werden, sondern oft ganz lokale, persönliche Anpassungen. „Ich hab meistens [das E-Mailprogramm] *mutt* im Einsatz mit [der Verschlüsselungssoftware] *gpg* und durch entsprechende Skripte noch halt aufgebohrt“ (Int 2008b). Die Nutzer_innen machen von ihrer Möglichkeit Gebrauch, den → *Quellcode* freier Software zu verändern oder durch eigene Skripte zu ergänzen.

Ästhetische Vorstellungen und Kreativität zeigen sich in FLOSS-Communities in Praktiken der Techniknutzung und -gestaltung. Viele personalisieren oder zweckentfremden ihre Hardware und Software, und das auf unterschiedlichen Ebenen: Sie bauen technische Änderungen in Software ein, nutzen sie für andere Zwecke als die ursprünglich indentierten, schreiben teilweise Software, deren Zweck Spaß ist. Dieser Umgang mit technischen Artefakten ist selbstbestimmt, Technologien erscheinen darin zutiefst gestaltbar. Vor allem die kreative Umnutzung von Technologien bringt in den Communities große Anerkennung ein, erfordert aber auch — wie beim Hacking-Contest — ein sehr hohes Maß an Wissen und Erfahrung. Die Möglichkeit zu einem selbstbestimmten Umgang mit Technologien ist deshalb auch sehr ungleich verteilt. Die in Hard- oder Software vergegenständlichten Äußerungen von Kreativität sind als solche wiederum nur für diejenigen wahrnehmbar, die den Sinn für diese Ästhetik teilen.

²⁸Das Programm *htop* ist ein Systemmonitor, der die laufenden Prozesse und die von ihnen verwendeten Ressourcen anzeigt.

²⁹Ein Beispiel für eine solche Erweiterung könnte z. B. sein, dass beim Starten des Programmes ein Abgleich der enthaltenen Userdaten mit einem anderen Computer stattfindet und die neuen Veränderungen beim Beenden des Programmes gleich wieder auf den anderen Computer geschickt werden.

6.2.2. Einschreibungen ästhetischer Vorstellungen in Artefakte

Während im letzten Abschnitt der kreative oder selbstbestimmte Umgang Einzelner mit ihren Artefakten im Vordergrund stand, geht es hier um die Kehrseite, nämlich um die ästhetischen Vorstellungen, die in den Standardversionen der Software enthalten sind. Auf sie stoßen Nutzer_innen, wenn sie Software neu in Betrieb nehmen.

A52 will nach einer Anleitung in der Zeitschrift *LinuxUser Pendrive-Linux* auf dem USB-Stick installieren und davon den Rechner starten. (...) Nach kurzem Hin und Her mit dem BIOS startet *Pendrive-Linux*. Als Desktop-Hintergrund ist ein Bild eingerichtet, das sie kommentiert: Sie fragt, ob sie da nicht mal einen jungen Mann bekommen könnte, was immer diese Frauen sollten. A1 antwortet darauf so etwas wie: „Ja, das hier ist wohl für Männer gemacht“ (LUG 2008k).

A52 äußert in dieser Szene Kritik beim Anblick des Standard-Hintergrundbildes. Das Bild zeigt eine Frau, die sich hinter vorgehaltener Hand spielerisch vor der Kamera zu verstecken scheint und lacht.³⁰ A52 gibt zu erkennen, dass sie sich von der Abbildung nicht angesprochen fühlt. Meiner Lesart nach verortet sie sich als heterosexuelle Frau und distanziert sich davon, dass die Technologie für heterosexuelle Männer gemacht ist, was sie scheinbar nicht zum ersten Mal beobachtet. Die neue Distribution, die sie ausprobiert, erscheint ihr durch das Hintergrundbild als vergeschlechtlicht. Sie und A1 stimmen in ihrer Interpretation überein, dass das Bild für heterosexuelle männliche Betrachter gemacht ist, dass die Kamera diesen männlichen Blick übernimmt.

Das Jonglieren mit den Konnotationen technischer Ausdrücke wurde bereits im Abschnitt 4.4 thematisiert. Das folgende Beispiel zeigt, wie vergeschlechtlichte Bedeutungen so in Software eingeschrieben werden können, dass sie zu einem verbreiteten Standard werden, der sich — anders als ein Hintergrundbild — nicht individuell ändern lässt, ohne die Funktionalität der Software zu beeinträchtigen. Die Beobachtung stammt aus einer Initiative, die → *Access Points* mit einer freien Software bestückt, um ein spezielles Funknetzwerk aufzubauen. Bei der Konfiguration von *Access Points* spielt die so genannte → *BSSID* eine Rolle. Dies ist eine Zahl, die die Funkzelle angibt. Sie wird mit einer Zahl in hexadezimaler Notierung angegeben, das bedeutet, dass neben den Zahlen 0 bis 9 die Buchstaben A bis F zur Verfügung stehen. Im Folgenden geht es um das Spiel mit einer hexidezimalen Zahl, die sich auch als Wort lesen lässt. Dieses Spiel wird als *Hexspeak* bezeichnet. Da

³⁰<http://www.pendrivelinux.com/>

sich bei diesem speziellen Projekt alle Access Points miteinander verbinden sollen, werden sie alle als *eine* Funkzelle mit derselben *BSSID* konfiguriert. Die *BSSID*, auf die das Entwicklerteam sich geeinigt hat, lautet 02:CA:FF:EE:BA:BE, „Zwei Kaffee, Baby!“. Hierin steckt eine subtile, vergeschlechtlichte Bedeutung, nämlich dass jemand (und hier wohl ein technisch versierter heterosexueller Mann) bei einer weiblichen Bedienung zwei Kaffee bestellt. Wie die Benennung zustande kam, wurde mir wie folgt beschrieben: Bei hexadezimalen Zahlen, steht nur eine begrenzte Anzahl an Zeichen zur Verfügung, aus denen eine *BSSID* entstehen sollte, die man sich leicht merken kann, da man sie ja oft braucht. Derjenige, der mir dies erklärte, machte deutlich, dass er es als *kreativen Akt* wahrnimmt, Hexspeak zu verwenden. Er trug auch ein T-Shirt mit derselben Aufschrift. Die *BSSID* ist für diejenigen sofort sichtbar, die mit einem textbasierten Programm nach verfügbaren WLAN-Netzen suchen. In einem grafischen Programm wird diese Information erst nach Betätigung zusätzlicher Schaltflächen dargestellt (vgl. Abb. 6.5). Die *BSSID* wäre nur durch eine Umkonfiguration *aller* Access Points veränderbar, damit das Netz weiterhin funktioniert. Da dieses Projekt dezentral strukturiert ist, sind die bereits vorhandenen Geräte an Tausenden von Orten verteilt und werden von zahlreichen Einzelpersonen gewartet. Eine Änderung wäre demnach nur schwer umzusetzen. Die Einschreibung wird damit ab einem gewissen Grad der Verbreitung zu einer technischen Notwendigkeit.

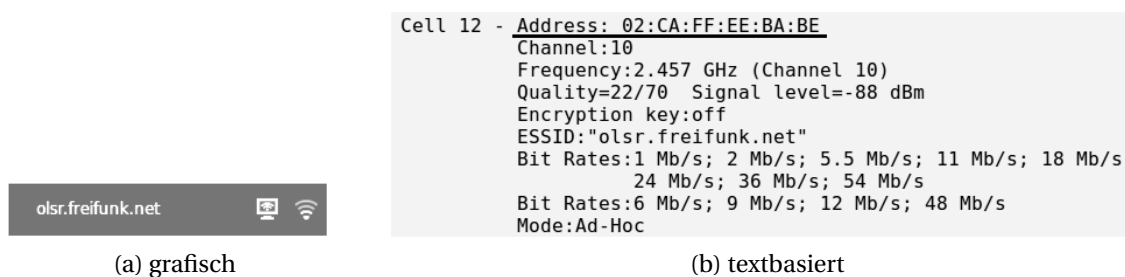


Abb. 6.5.: Suche nach WLAN-Netzen

Hier spielen die Reproduktion der Differenzkategorie Geschlecht und die stilistische Distinktion zusammen: Erstens wird eine vergeschlechtlichte Anspielung in eine Technologie eingeschrieben, die ein Bild von Kaffee kochenden Frauen und damit von geschlechtlich segregierter Arbeitsteilung evoziert. Die eingeschriebene Konnotation ist dauerhaft angelegt, eine Veränderung auf einzelnen Geräten würde sie vom Netzwerk abschneiden. Zweitens ist diese Einschreibung vor allem für diejenigen sichtbar, die die Access Points konfigurieren, und für die, die sich textbasierter Programme bedienen. Distinktion heißt hier, technische Mittel einzusetzen, die die Konnotation kaschieren. Technisch nicht so

Versierte lassen sich diese Zusatzinformation kaum anzeigen und können sie damit auch nicht kritisieren. Sie wird zwar auf T-Shirts reproduziert, meine Beobachtungen haben jedoch auch gezeigt, dass nicht alle Menschen auf die Idee kommen, Hexadezimalzahlen als Wörter zu lesen.³¹ Damit nehmen drittens nur diejenigen hier einen Akt von Kreativität überhaupt wahr, die ein Auge für Hexspeak haben und um die Begrenzung möglicher Zeichen wissen.

Eine andere Form von ästhetischen Einschreibungen von Entwicklerseite kann an der Gestaltung des → *Desktops* der Distribution *Ubuntu* nachvollzogen werden. Der Standarddesktop wurde zwischen dem ersten Release 2004 und der aktuellen Version (2011) stark verändert, was die Frage nach einer herrschenden Ästhetik bei der Gestaltung grafischer Oberflächen aufwirft. *Ubuntu* Linux ist eine Distribution, die von dem südafrikanischen, weißen Multimillionär Mark Shuttleworth ins Leben gerufen wurde. „Ubuntu“ bedeutet in etwa „Gemeinsinn“ oder „Menschlichkeit“; der Name der Distribution geht auf *Ubuntu* als afrikanische Philosophie zurück.³² Die Gestaltung des Standard-*Ubuntus* trug der südafrikanischen Herkunft des Projektes lange Rechnung: Der Standarddesktop war jahrelang von braunen Erdfarben dominiert, die mit afrikanischen Savannen assoziiert werden, der Anmeldeklang erinnert an Buschtrommeln.³³ Auf den kostenlos verschickten Installations-CDs waren Bilder von Menschen verschiedener Hautfarben, die sich an den Händen hielten. 2008 hat Shuttleworth erklärt, die Entwickler_innen sollten sich an *Apples* Betriebssystem *Mac OS X* orientieren, damit „freie Software in Bereichen wie Look & Feel, Style und Design verbessert“ werde.³⁴

In der Entwicklung von *Ubuntus* Desktopgestaltung deutet sich eine Vereinheitlichung in Richtung einer „herrschenden“ Ästhetik an, die von *Apples* proprietärer Software vorgegeben wird: Über die Jahre wurden zahlreiche Details so umgestaltet, dass sie stark an den Desktop von *Mac OS X* erinnern, wie das Aussehen des Standard-Dateimanagers, die Icons oder die abgerundeten Fensterecken. Seit dem Release 10.04 (2010) ist das Design nicht

³¹Eine Person sah ein Mindmap von mir, das unter dem Titel Geschlecht stand, und fragte, warum da eine → *MAC-Adresse* stehe. Sie erkannte damit die hexadezimale Notierung, las sie aber als Zahl, nicht als Wort.

³²Diese Philosophie beinhaltet vor allem eine respektvolle und anerkennende Grundhaltung gegenüber Menschen und die Achtung der Menschenwürde für ein friedliches Zusammenleben. Dem Teilen kommt darin eine wichtige Rolle zu (vgl. [https://secure.wikimedia.org/wikipedia/de/wiki/Ubuntu_\(Philosophie\)](https://secure.wikimedia.org/wikipedia/de/wiki/Ubuntu_(Philosophie)), Abruf: 10.12.2011).

³³vgl. für diese Assoziationen <https://secure.wikimedia.org/wikipedia/de/wiki/Ubuntu>, Abruf: 10.12.2011

³⁴<http://www.golem.de/0902/65175.html>, Abruf: 08.05.2011, siehe auch <http://www.golem.de/0807/61283.html>, Abruf: 08.05.2011

mehr braun, sondern Fensterrahmen in einem nüchternen Dunkelgrau heben sich vor einem violetten Hintergrund ab, seit 11.04 (2011) sind die Menüs der Programme im Unity-Desktop in die Leiste am oberen Bildschirmrand integriert statt in die Leiste des jeweiligen Fensters. Von der „ethnisierten“ Gestaltung ist nur noch der Anmeldeklang übrig geblieben. Das typische *Apple*-Dock, in dem die Anwendungsstarter aufgereiht sind, ist längst als freie Software nachprogrammiert; in entsprechenden Foren sind seit Jahren → *Themes* erhältlich, die das „Look & Feel“ des *Mac* auf den Linux-Desktop bringen.³⁵ In diesen Veränderungen wird deutlich, dass sich freie Software zum proprietären Softwaremarkt verhält: FLOSS-Communities und Firmen reagieren darauf, dass ein Konzern ein Produkt entwickelt hat, das nicht allein als Technik, sondern als ganzer Lifestyle vermarktet wird.³⁶ Dies sollen verdeutlichen, dass von Seiten der Hersteller_innen der Distributionen oder Programmen eine jeweilige Ästhetik in die grafischen oder textbasierten Oberflächen eingeschrieben wird, die den Nutzer_innen zunächst als Standard oder als Ausgangspunkt für die eigene Konfiguration dient. Dabei sind die Artefakte in ihrer „offiziellen“ Version (also so, wie sie von den Softwareprojekten herausgegeben werden) keineswegs neutral. Sie tragen Bedeutungen und Bezüge in sich, referenzieren subtil auf Konkurrenzprodukte oder auf Vorstellungen von Geschlechterverhältnissen oder Nutzer_innen.

Ähnliche Einschreibungsprozesse wurden bereits in anderen Zusammenhängen erforscht, wir haben es nicht mit einem linuxspezifischen Phänomen zu tun.³⁷ Madeleine Akrich, die aus der Perspektive der Akteur-Netzwerk-Theorie³⁸ schreibt, spricht von Skripten, die die Entwickler_innen in technische Artefakte einschreiben. Sie vergegenständlichen ihre Vorannahmen über die künftigen Nutzer_innen in ihnen, basierend auf Analysen oder aber unbewusst (vgl. Akrich 1992; Akrich 1995). Die feministische Technikforschung griff Akrichs Skriptekonzept auf und dachte es mit der Analysekategorie Ge-

³⁵Diese Themes wurden nicht von Ubuntu entwickelt, sondern von Einzelpersonen, die sie ins Netz gestellt haben.

³⁶vgl. z. B. <http://www.tagesanzeiger.ch/digital/computer/Apple-strahlt-mehr-Lifestyle-als-Technik-aus/story/21544303>, Abruf: 02.06.2011

³⁷Die Auswahl der im Folgenden angeführten Literatur erhebt keinen Anspruch auf Vollständigkeit. Für eine exzellente Aufarbeitung des Forschungsstandes zur Vergeschlechtlichung technischer bzw. informatischer Artefakte vgl. Corinna Bath (2009), auf die ich mich im Folgenden stark beziehe.

³⁸Die Akteur-Netzwerk-Theorie (ANT) wurde ab Mitte der 1980er Jahre von Bruno Latour und Michel Callon entwickelt. Sie hebt hervor, dass auch Gegenstände Handlungspotenziale haben. „Nicht nur der Mensch sei Akteur im gesellschaftlichen Geschehen, sondern es gebe viele Faktoren bzw. Aktanten, die die konkreten Ereignisse bzw. Operationsketten von Handlungen beeinflussten“ (Ruffing 2009: 29). „Aktanten“ ist ein aus der Semiotik entlehnter Begriff, der die Erweiterung im Vergleich zum Akteursbegriff verdeutlichen soll, mit dem ausschließlich Menschen gemeint sind (vgl. ebd.; zur Begriffsverwendung in der ANT siehe auch Akrich/Latour 1992). Die ANT denkt somit in so genannten Kollektiven aus menschlichen und nicht-menschlichen Wesen, die gemeinsame Handlungen oder Handlungsketten vollziehen.

schlecht zusammen (vgl. Rommes 2002: 17 f.). Die Autor_innen sprechen von Genderskripten, wenn Entwickler_innen von Technologien eine Auswahl treffen, die auf einer symbolischen, strukturellen oder identitären Ebene vergeschlechtlichte Zuschreibungen enthält (vgl. Bath 2009: 78 ff.). Ein Beispiel für eine Arbeit, die mit dem Konzept der Genderskripte Artefakte analysiert hat, ist Ellen van Oosts Untersuchung (vgl. van Oost 2003; vgl. auch Cockburn/Ormrod 1993; Offenberger/Nentwich 2009): Die Autorin hat zwei Modelle von Rasierapparaten auf die dort eingeschriebenen Genderskripte untersucht. Im Design der beiden Apparate, die denselben Zweck erfüllen, zeigen sich stereotype Annahmen über das Verhältnis späterer Nutzer_innen zu Technik. So war das Modell „Philishave“ in schwarz und Metall gehalten und verfügte über ein Display mit Informationen und Einstellungsmöglichkeiten. Außen waren Verweise auf das Innenleben des Geräts zu entnehmen. Beim Modell „Ladyshave“ wurde das technische Innere unsichtbar gemacht, indem keines dieser Elemente vorhanden war. Statt Schrauben war ein Klickmechanismus zum Öffnen und Schließen vorhanden (vgl. Offenberger 2010: 88).

„Die Studie zeigt, wie in das Design geschlechterstereotype Annahmen über Technikaffinität bzw. Technikferne der zukünftigen NutzerInnen eingeschrieben sind. (...) Die Tatsache, dass für dieselbe Tätigkeit je nach Geschlecht verschiedene Gerätetypen angeboten werden, verweist unmittelbar auf das System der Zweigeschlechtlichkeit als einem grundlegenden gesellschaftlichen Strukturierungsprinzip. Jenseits aller technischen Sachzwänge werden stereotype Annahmen über männliche und weibliche Kompetenzen, Eigenschaften und Aufgaben in Artefakte eingeschrieben, die somit dazu beitragen, das System der Zweigeschlechtlichkeit zu reproduzieren“ (Offenberger 2010: 88).

Den Konzepten der Skripte bzw. Genderskripte geht es einerseits um die Einschreibung von Geschlechterverhältnissen in Objekte im Prozess der Technikentwicklung. Laut Els Rommes stellen die Entwickler_innen sich oft selbst als Nutzer_innen der Artefakte vor („I-Methodology“). Da sie zum überwiegenden Teil männlich sozialisiert seien, würden dadurch „unbewußt einseitige männliche NutzerInnenbilder“ bedient, in denen „auf hegemoniale Maskulinitätsvorstellungen“ zurückgegriffen würde (Bath 2009: 79 bzgl. Rommes 2002 und Oudshoorn et al. 2004). Dies zeige sich an verwendeten Symbolen und zugeschriebenen bzw. erwarteten Kompetenzen. Andererseits bezieht das Konzept genauso die Entschlüsselung der Skripte durch die Nutzer_innen der Artefakte mit ein. Es denkt diese Entschlüsselung nicht deterministisch: Nutzer_innen könnten sie von sich weisen, sich Technologien auf andere Weise als auf die intendierte oder nahegelegte an-

eignen (vgl. Akrich 1992: 208; siehe auch Wajcman 2002: 353). Laut Bath wurde dieser Aspekt des Konzeptes oftmals missverstanden und ihm daher zu Unrecht Technikdeterminismus vorgeworfen (vgl. ebd. 83 f.).

Die feministische Techniksoziologie konzentrierte sich nicht nur auf Hardware, sondern wies auch auf der Ebene der Software eingeschriebene Annahmen über Nutzer_innen nach (vgl. z. B. den Sammelband von Zorn et al. 2007). Dabei ging es z. B. um die Gestaltung vergeschlechtlichter Avatare oder → *Softwareagenten* (vgl. Weber/Bath 2007; siehe auch die Analyse der Video- und Computerspielfigur Lara Croft bei Lübke 2004: 65 f.), um das Design von Computerspielen (vgl. John 2006) oder um vergeschlechtlichte Annahmen bei der Entwicklung von Textverarbeitungssoftware (vgl. Hofmann 1997; Hofmann 1999).

Die Konzepte der Skripte bzw. Genderskripte lassen sich für die Auswertung meiner Beobachtungen gut nutzen, da Beobachtungen zu beiden Seiten vorliegen: Einerseits wurde deutlich, dass freie Software von den Macher_innen eine bestimmte Gestalt bekommt, durch die sie sich besonders an bestimmte Nutzer_innen zu richten scheint: Software, deren Standarddesktop mit bestimmten Bildern von Frauen dekoriert ist, richtet sich an Nutzer_innen, die diesen Frauentyp attraktiv finden, und zwar in erster Linie an heterosexuelle männliche Betrachter. Software, aus der die „ethnisierten“ Gestaltungselemente herausgenommen werden, die die südafrikanische Herkunft von *Ubuntu* reflektierten, scheint (Life)Style-Ideale weißer Nutzer_innen im globalen Norden anzusprechen. Software, die — wie die → *Access Points* — eine Botschaft auf der Ebene der Konfiguration in sich tragen, richtet sich an diejenigen, die dort überhaupt hinsehen und die Botschaft entschlüsseln. Solche Einschreibungen schließen niemanden tatsächlich von der Nutzung aus, das Problem liegt vielmehr in den Anpassungsleistungen, die bestimmten Nutzer_innen implizit abverlangt werden. Rommes zeigt dies am Beispiel von Frauen, der Gedanke kann auf alle durch die Skripte nicht gemeinten Personen(-gruppen) ausgeweitet werden.

„More likely, it means that some women have to adjust more, have to work harder, in order to be able to use a certain technology in present society. A gender script analysis can thus be specified as a study of ‘mechanisms of adjustment’ or the failure to adjust between the technology and a potential user. It is about who has to ‘adjust’ more, who has to pay the price for not fitting the norm that is reproduced in the artifact“ (Rommès 2002: 19).

Andererseits spiegeln meine Beobachtungen auch die Entschlüsselung von Einschreibungen wider: Die Nutzerin, die ihr Missfallen am Hintergrundbild laut äußert, thematisiert in der LUG die Einschreibung und bekommt von einem (männlichen) Sitznachbarn eine Bestätigung für ihre Lesart. Ebenso können die Praktiken der kreativen Umnutzung von Software als „De-Description“ (Akrich 1992), als Entzifferung interpretiert werden. Die Teilnehmer_innen des Hacking-Contest wissen, zu welchem Zweck eine Software vorgesehen ist, sie testen aber aus, was sie außerdem noch alles mit ihr machen können. Sie lassen sich nicht vorgeben, wie und wozu sie verwendet werden soll. „The freedom to run the program, for any purpose“³⁹ ist auch die erste der vier Freiheiten, die freie Software laut der *Free Software Foundation* ausmachen. In den konstitutiven Prinzipien freier Software wird somit eine sehr selbstbestimmte „De-Description“ vorausgesetzt. Jede_r kann die Software benutzen, aber nicht jede_r ist vom Wissensstand her dazu in der Lage, ihre Besonderheiten für sich zu nutzen.

Zu Beginn dieses Abschnitts ging es auch um die Hardware, die von etlichen Linux-Nutzer_innen durch eigene Umbauten zu personalisierten Artefakten gemacht wird. Die Studie von Ellen van Oost lädt dazu ein, die Hardware der verbreitetsten Notebooks in FLOSS-Communities, der ThinkPads, genauer zu betrachten. Die Rechner haben seit der Einführung des ersten Modells 1992 ein sehr ähnlich bleibendes Design: Sie sind schwarz und kantig, haben den in Abschnitt 4.4 erwähnten TrackPoint, teilweise zusätzlich zum Touchpad, teilweise statt des Touchpads. Über dem Bildschirm ist eine Leuchte eingelassen, die bei Dunkelheit ein spartanisches Licht auf die Tastatur wirft. An der Unterseite haben sie diverse Schrauben, die mit Symbolen gekennzeichnet sind. Sie weisen darauf hin, an welche Hardwareteile man an den entsprechenden Stellen herankommt (Festplatte, Arbeitsspeicher, Öffnen des kompletten Gehäuses); im Internet stehen Pläne für die Modelle zum Download bereit, die zeigen, was wo ist und wie die Teile ausgetauscht werden (z. B. der Lüfter). Auch sind ThinkPads dafür bekannt, im Gegensatz zu sehr kostengünstigen Notebooks Hardware zu enthalten, die unter Linux gut unterstützt wird.

Weiter oben wurde Sherry Turkles Perspektive auf die grafische Oberfläche des *Macintosh* angeführt (vgl. Turkle 1998: 49 ff.). Analog dazu eignet sich auch die Hardware der *Apple*-Notebooks zu einem kurzen Vergleich mit den ThinkPads, um die Skripte sichtbar zu machen: Die meisten Notebooks von *Apple* sind in hellen Farben gehalten, weiß oder silber; die Ecken des Gehäuses sind abgerundet; das Touchpad gilt als ausgefeilt. Im

³⁹<http://www.gnu.org/philosophy/free-sw.html>

Deckel tragen die Rechner als Leuchte das Firmenlogo, das — frei von technischer Funktionalität — in erster Linie für gegenüber sitzende Personen sichtbar ist. Die Modelle sind unterschiedlich gestaltet bzw. wurden über die Jahre verändert. Bei dem um 2004 gängigen *iBook G4* waren die vier Schrauben zum Öffnen des Gehäuses an der Unterseite in den Ecken, die von Gummifüßchen verdeckt waren. An den Arbeitsspeicher und die WLAN-Karte kam man über das Hochklappen der Tastatur heran, die mit schraubenlosen Scharnieren eingerastet war.⁴⁰ Auch hier werden Annahmen über die Technikaffinität der Nutzer_innen in der Hardware vergegenständlicht. *Macs* scheinen sich eher an Nutzer_innen zu richten, die abgesehen von der Software auch an der Hardware wenig herumbasteln und denen das Äußerliche des Rechners wichtig ist. Dies bedeutet nicht, dass Käufer_innen von *Apple*-Produkten die in die Hard- oder Software eingeschriebenen Skripte alle hinnehmen: Das so genannte Jailbreaking des *iphone*, also das Erlangen von root-Rechten, gehört zu den Umgangsweisen, mit denen Nutzer_innen ein bestimmtes Skript ablehnen. Für ihre Kontrolle über das Gerät geben sie die Herstellergarantie auf. Demgegenüber wird den Nutzer_innen von ThinkPads nahegelegt, dass sie das Gerät öffnen können. Robust statt stylisch.

In diesem Abschnitt wurde gezeigt, dass auf verschiedenen Ebenen ästhetische Vorstellungen in Bezug auf Hard- und Software in FLOSS-Communities existieren. Mit den angeführten Arbeiten aus der feministischen Technikforschung lässt sich für diesen Abschnitt festhalten, dass technische Artefakte keine neutralen Gegenstände sind, sondern dass sie bewusst oder unbewusst Vorannahmen über Nutzer_innen enthalten. Die beobachteten Beispiele verweisen tendenziell auf sich männlich verortende, weiße und technisch versierte Nutzer, die im globalen Norden leben. Längst nicht alle Linux-Nutzer_innen entsprechen diesem Bild. Wer sich in den Gestaltungselementen nicht wiederfindet, kann sich — mit einem Extraaufwand — der für freie Software konstitutiven Freiheit bedienen, die Software zu verändern oder umzunutzen. Die Praktiken der Techniknutzung und -umgestaltung müssen jedoch als Praktiken begriffen werden, die innerhalb von FLOSS-Communities in sehr unterschiedlichem Maße verbreitet sind. Je mehr Einblicke Nutzer_innen in das Betriebssystem haben, desto souveräner können sie Anpassungen, z. B. durch eigene Skripte (im Sinne von kleinen Programmen), vornehmen oder die Umnutzungsideen anderer als kreative Ideen wahrnehmen und wertschätzen. Die Souveränität, mit der eingeschriebene Skripte abgewiesen werden können, ist ein Distinktionskriterium

⁴⁰Bei aktuellen *MacBooks* sind die Schrauben nicht mehr verdeckt. Das *iBook* war ein Modell, das im Gegensatz zum *PowerBook* für Privatanwender_innen, nicht für den „professionellen“ Einsatz konzipiert war.

in den Nutzungspraktiken. Die Aneignung von technischem Know-how und von situativer, praktischer Problemlösungskompetenz geht einher mit der Aneignung von ästhetischen Wahrnehmungskategorien. Diese erlauben es fortgeschrittenen Nutzer_innen in zunehmendem Maße „Unterscheidungsvermögen an den Tag zu legen“ (Bourdieu 1998b: 22), aber auch, sich selbst kreativ zu äußern, souverän mit Software zu jonglieren und mit ihren Äußerungen zum Vorbild für andere zu werden (vgl. analog dazu Bourdieu 1987: 104 ff.).

6.3. Die Strukturierung der Zeit bei LUG-Treffen

Zeit stellte in der bisherigen Materialauswertung in erster Linie die Zeit dar, die Community-Mitglieder in Lernprozesse investieren oder bereits investiert haben. Dies spiegelt sich in Bezug auf Zeit beim Zusammentreffen auch darin wider, dass die Beteiligten häufig nicht das gleiche Tempo im Umgang mit Computern haben. In diesem Abschnitt wird dagegen untersucht, wie die Zeitlichkeit der Praktiken von FLOSS-Communities durch die Technologien strukturiert wird. Dieser Aspekt der linuxbezogenen Praktiken kam in meinen Hypothesen in Bezug auf Differenz, Distinktion und Ausschluss nicht vor; er zeigte sich jedoch bei den Datenerhebungen als sehr bedeutsam: Bevor die gemeinsame Beschäftigung mit Linux beginnen kann, ist es meistens erforderlich, dass die Beteiligten sich eine gemeinsame Arbeitsumgebung aufbauen, Hard- und Software miteinander abstimmen und sie optimieren. Diese Vorstrukturierung der Zeit durch Technologien ist mir besonders bei Treffen aufgefallen, bei denen Gruppen sich verabredet hatten, um konkrete Aufgaben zu erledigen, z. B. um Fehler in einer Distribution zu beheben (→ *Bug Jam*) oder um sich auf die LPI-Prüfung vorzubereiten. In diesem Abschnitt werden diese Beobachtungen in Bezug auf die Forschungsfragen beleuchtet: Was bedeutet es, dass Technologien die gemeinsame Zeit strukturieren? Was hat diese Strukturierung mit konstitutiven Praktiken von FLOSS-Communities zu tun? Inwiefern ist die Strukturierung der Zeit durch Technologien für die Herstellung von Differenzen und für Distinktionsprozesse relevant?

6.3.1. „Dreieinhalb Stunden. Wahnsinn.“ — Zeitgefühle

In diesem Abschnitt werden zunächst verschiedene Praktiken aus einem ausführlichen Materialbeispiel herausgearbeitet, die dann weiter unten anhand von anderen kürzeren

Sequenzen weiter ausgeführt werden. Der erste Protokollausschnitt stammt aus einer selbstorganisierten Lerngruppe, die sich auf die LPI-Prüfung vorbereitet. Sehr viel Zeit vergeht damit, überhaupt gemeinsam arbeitsfähig zu werden. Die fünfköpfige Gruppe möchte lernen, einen → *Mailserver* zu konfigurieren und dies auf einem Testserver üben. Im Unterschied zum technischen Support in der LUG, wo Kooperation u. a. dadurch charakterisiert wird, dass *eine* Person vor dem fraglichen Computer sitzt und ihn bedient, „treffen“ sich hier alle Beteiligten per Login auf dem gemeinsamen Testserver und müssen sich einigen, wer momentan zum Agieren befugt ist. Die folgende Sequenz zeigt die Schritte, die von den Beteiligten vollzogen werden, bis sie sich dem Mailserver zuwenden. Um die Dauer bis zum Beginn der „eigentlichen“ Übung zu zeigen, wird das Protokoll bis dahin vollständig wiedergegeben.

Der Plan von A39 und A2 ist es nun, dass alle sich per → *ssh* auf dem Testserver einloggen. A39 stellt fest, dass auf dem Server noch kein *ssh* installiert ist. Er erwähnt, dass man mit → *apt-get* irgendwie auch in einem Rutsch Pakete installieren und andere deinstallieren könne. Er geht wieder aus dem Raum, zum Server. Als er wiederkommt, sagt er, jetzt lade der Server Updates herunter. Nach einem Moment geht er nochmal nachsehen. A2 erzählt, womit er sich am Vorabend beim LUG-Treffen beschäftigt hat. A48 aktualisiert seine Software mit einem grafischen Programm. Zwischendurch versuchen alle Anwesenden immer wieder, ob sie nun den Testserver über *ssh* erreichen können. Dabei stelle ich fest, dass ich sowohl bei A48 als auch bei A39 ziemlich schlecht lesen kann, was auf ihren → *Shells* steht. (...) Beide haben einen schwarzen Hintergrund mit weißer Schrift eingestellt und zwar recht kleine Schrift, vor allem A39.

A2 stellt fest: „Noch kein *ssh*.“ Er ruft A39 zu: „Hast du auf dem Ding → *screen* installiert?“ A39 hat ihn nicht verstanden und kommt herein. A2 schlägt vor, dass über *screen* einer von ihnen konfigurieren könnte und die anderen auf ihren Bildschirmen alles mitverfolgen könnten. A39 findet die Idee gut. Beide gehen raus zum Server. A46 und A47 tauschen sich über irgendein technisches Thema aus, ich wende meine Aufmerksamkeit aber A48 zu. Er sucht über seine grafische → *Paketverwaltung* nach *sshd*, dann nach *ssh*. Er scrollt mit seiner USB-Maus durch die Liste und sieht sich an, was es dort gibt. Er sieht ein grünes Häkchen bei *openssh*. Er sagt, er gehe sich noch einen Kaffee holen, steht auf und verlässt die Vereinsräume. Mir fällt auf, dass er seinen Rechner mit geöffneter → *root*-Shell vor mir stehen lässt.

A39 und A2 kommen zurück. A39 fragt, wie sie es jetzt machen wollen und A2 fordert ihn auf, sich per *ssh* einzuloggen. Ich beobachte, dass A39 nur mit zwei oder drei

Fingern tippt, nämlich mit beiden Zeigefingern. Ich glaube, dass er manchmal noch einen Mittelfinger zuhelfe nimmt. Sie haben Probleme mit der Bedienung von *screen*. Sie probieren erst, ob sie es aus dem Kopf wissen, wie eine *screen* gestartet wird und alle sich damit verbinden können. Dann recherchiert A2 dazu im Internet. Einer von ihnen hat überhaupt keine Netzwerkverbindung. Nachdem mehrere von ihnen *screen* auf dem Server gestartet haben, laufen dort etliche Sitzungen, die A39 sich anzeigen lässt. Niemand versteht, welche welche ist. A39 möchte die laufenden Sitzungen wieder schließen. Dazu startet er den Rechner neu. (Ich denke: Warum beendet er nicht die einzelnen Prozesse, sondern startet gleich den ganzen Server neu? Die Antwort ist schon klar, er weiß nicht wie, er beschäftigt sich aber nicht damit, es herauszufinden.) A39 sagt: „Das haben wir noch nie gemacht.“ Zwei oder drei der anderen erinnern sich jedoch an ein länger zurück liegendes Treffen, wo sie wohl doch schon mal so gearbeitet haben. A39 loggt sich erneut per *ssh* auf dem Server ein und fordert A2 auf, eine *screen*-Sitzung zu starten. Sie sammeln nochmal ihre Ideen, dann klappt es: A2 startet mit *screen* herum. A46 fragt, ob A48 root sei. Der antwortet: ‚Klar, das nützt mir aber nichts.‘ Ich sehe, dass A48 zwar root ist, aber auf seinem Notebook, nicht auf dem Server. (Ich wundere mich, dass er es nicht versteht oder so unaufmerksam ist.) A48 loggt sich erneut als → *User* auf dem Server ein. Er startet *screen* und sieht nun, was jemand anders eingegeben hat. (Das ist aber die User-Sitzung, die A46 und A47 zuvor gestartet hatten, nicht die, auf der jetzt alle anderen sind.)⁴¹ A39 schießt in der Zwischenzeit die Sitzung ab, mit der die vier verbunden waren. Niemand kann die Shell verlassen, bei A39 hängt sogar der ganze Rechner. Er sagt, dass er das war. Das zeige, dass er damit noch nicht vertraut sei. Er fragt, ob er den Server neu starten solle. A2 verneint. (Er startet die root-Sitzung wohl neu.) Es gibt nun immer noch zwei Sitzungen, eine als root, eine als User. Niemand steigt durch, mit welcher er nun gerade verbunden ist. A46 sagt, wenn sie so weitermachen, würde das ja lange dauern. A2 und A39 sind nun wieder drin. A48 nicht. Er ist immer noch als User auf dem Server eingeloggt. Einer fragt ihn, ob er root sei. „Ich bin root.“ Er blickt auf seinen Bildschirm. „Ich war jedenfalls root.“ Er schließt die Konsole. (Ich frage mich, ob er seinen Irrtum vor mir verstecken möchte.) A2 kommt um den Tisch herum. A48 öffnet eine neue Konsole und ruft mit der Pfeil-hoch-Taste die → *History* auf. Er will sich wieder als User mit dem Server verbinden. A2 will ihm in die Tastatur greifen und sagt: „Nee, root@...“ A48 tippt, A2 diktiert ihm nochmal die → *IP-Adresse* des Servers. A48 sagt: „Das habt ihr mir vorhin nicht gesagt.“ A39 witzelt: „Du wolltest uns von → *Mandriva* überzeu-

⁴¹Jede *screen*-Session wird mit den Rechten des Accounts gestartet, mit dem man sich eingeloggt hat. Ein → *User* hat nicht das Recht, sich mit einer *screen*-Session des Administratoraccounts root zu verbinden. A48 irrt sich hier diesbezüglich.

gen?“ (Ich denke, aha, die Distribution, die A48 nutzt, ist Mandriva. Der Spruch von A39 erlaubt es, das Nichtfunktionieren auf Mandriva zu schieben.) A48 ist jetzt mit derselben *screen*-Sitzung verbunden wie die anderen vier. A46 und A48 haben eine kleine und schnelle Interaktion über *screen*. A46 hat eingetippt ‚Mandriva‘ und wollte wohl etwas Negatives über Mandriva schreiben. A48 greift über seinen Rechner ein und korrigiert ‚Mandriva ulez‘. (Ich denke, er meint ‚Mandriva rules‘.) Beide lachen. Nun kann es losgehen, sie wenden sich dem Server zu. Das (eigentliche) Thema des Abends ist die Konfiguration eines Mailservers (LUG 2009d).

Zur Interpretation: Die Dauer dieser Vorbereitungen kommt durch verschiedene Verzögerungen zustande: Erstens stoßen die Beteiligten Prozesse in den Rechnern an, auf deren Beendigung sie dann warten müssen. Ihr praktischer Sinn gibt der Installation vorhandener Sicherheitsupdates absolute Priorität: Für das Arbeiten ist es demnach zentral, die Software aktuell zu halten. Zweitens schweifen Einzelne vom Thema ab, wenn ihnen bei den Vorbereitungen andere technische Fragen oder Möglichkeiten einfallen, die sie auch interessieren (z. B. die Bedienung von *apt-get* oder *screen*). A39 und A2 erweitern durch dieses Abschweifen die Definition des Lerngegenstandes (vgl. Holzkamp 1995: 509 ff.): Die Recherche nach einer bestimmten Option des *apt-get*-Befehls⁴² bzw. die Inbetriebnahme von *screen* für die Kooperation werden dadurch quasi nebenbei als Teile des gemeinsamen Lernprozesses definiert, auch wenn sie nicht in den LPI-Lernzielen⁴³ stehen mögen. Während A39 und A2 diese Umdefinitionen im Konsens vornehmen, machen die anderen drei Personen zwar mit, handeln die inhaltliche Änderung jedoch nicht verbal aus. Von einem expliziten Konsens der ganzen Gruppe über die Abschweifungen kann nicht gesprochen werden. Mindestens A39 und A2 teilen jedoch einen praktischen Sinn: Wenn auf dem Lösungsweg Programme zum Einsatz kommen können, die auch nützlich oder interessant erscheinen, dann sind sie bereit, sich (unter Umständen sofort) mit ihnen auseinanderzusetzen.⁴⁴

Dieser gemeinsame praktische Sinn der Beobachteten wird wiederum vor allem in der Konfrontation mit meiner Beobachter_innenperspektive deutlich: Der Einsatz von *screen* kommt mir als Abschweifen vom „eigentlichen“ Thema vor. Es wäre aus meiner Perspektive auch möglich gewesen, dass alle sich direkt vor den Server oder vor ein Notebook

⁴² „Auf dem Server soll der Mailserver *exim* deinstalliert und stattdessen *postfix* installiert werden. A2 tippt ein: ‚apt-get install‘ und A39 sagt den Befehl auch laut (obwohl alle mitlesen können). A39 hält ihn auf und ruft in einer Shell bei sich lokal die *manpage* von *apt-get* auf, um nachzusehen, wie er in *einem einzigen* Befehl sowohl *exim* deinstalliert als auch *postfix* installiert“ (LUG 2009d).

⁴³ vgl. <http://www.lpice.eu/LPIC-Programm.52.0.html?&L=4>, Abruf: 20.10.2010

⁴⁴ Vergleiche zum Abschweifen auch den folgenden Abschnitt 6.3.2.

setzen, um die Konfigurationsübungen zu sehen. Ich teile den praktischen Sinn der Beteiligten nicht, sondern mache einen Unterschied zwischen dem „eigentlichen“ Lernziel und dem Abschweifen zu einem anderen Thema, den die Involvierten nicht machen. Die Dauer des Prozesses wird zwar bemerkt⁴⁵, niemand macht jedoch den Vorschlag, das Unterfangen abubrechen, um schneller mit dem Thema Mailserver-Konfiguration beginnen zu können. Die fünf teilen im Vergleich zu mir mehr Habitus, da sie nicht nach der Notwendigkeit von *screen* für die Lernsituation fragen, sondern das Interesse an der technischen Lösung teilen, selbst wenn die Inbetriebnahme der Software länger dauert als eine nichttechnische Lösung. Möglicherweise zählt zum praktischen Sinn auch, dass die Beteiligten lieber am eigenen Computer sitzen als sich als Zuschauer_innen vor einem anderen Rechner zu versammeln, selbst wenn sie auch vor dem eigenen Rechner „nur“ zuschauen.⁴⁶

Drittens zieht das Abschweifen zum Thema *screen* zeitliche Verzögerungen nach sich, die darauf zurückzuführen sind, dass die Anwesenden mit der Software nicht vertraut sind und ausprobieren und recherchieren. Neben der Bedienung von *screen* kommt es zu Verwirrungen mit den verschiedenen Accounts (→ *User* vs. → *root*), unter denen sie auf dem Testserver einloggt sind und *screen*-Sitzungen starten. Die Einbeziehung von *screen* in die Definition des gemeinsamen Lerngegenstandes führt dazu, dass wiederum ein anderes, grundlegendes Thema, Zugriffsrechte unter Unix/Linux, aufgearbeitet werden muss, um gemeinsam weiterlernen zu können.⁴⁷ A39, der *screen* vorgeschlagen hat, und A2, der sich dem Vorschlag gerne anschließt und dazu recherchiert, teilen eine Experimentierfreude und Bereitschaft zu zusätzlichen Recherchen, die A48 in dieser Situation nicht zeigt. Während der Vorbereitungen auf dem Server holt er sich Kaffee. Er loggt sich im Gegensatz zu den anderen nicht routiniert unter dem richtigen Account auf dem Server ein, beginnt aber — anders als A2 — keine Recherche, um mögliche Ursachen für das Nichtfunktionieren zu finden. Ich notiere meine Verwunderung über ihn.⁴⁸ Damit lege ich auch hier meinen eigenen impliziten Maßstab an seinen Habitus an: Ich verlange von Menschen, die sich auf die LPIC-2-Prüfung vorbereiten, dass sie die Wahl des richtigen Accounts in-

⁴⁵ „A46 sagt, wenn sie so weitermachen, würde das ja lange dauern.“

⁴⁶ Bei der Kooperation über *screen* kann auch jeweils nur eine Person tippen. Das Programm erfordert deshalb im weiteren Verlauf des Treffens noch mehr Zeit, in der die Personen ständig aushandeln, wer wann und wie lange dazu befugt ist, die *screen* zu kontrollieren.

⁴⁷ „Er will sich wieder als User mit dem Server verbinden. A2 will ihm in die Tastatur greifen und sagt: ‚Nee, root@...‘“

⁴⁸ „Ich wundere mich, dass er es nicht versteht oder so unaufmerksam ist.“

korporiert haben und (re-)produziere damit in meinem Protokoll eine Vorstellung von allgemeingültigem Grundlagenwissen.⁴⁹

Wenn man die protokollierte Situation auf Distinktion unter den Beteiligten hin befragt, fällt auf, dass A48 nicht dieselben habituellen Dispositionen zeigt wie A2 und A39: Er aktualisiert zwar auch zu Beginn seine Software und vollzieht auf seinem Notebook nach, was A39 auf dem Testserver tut.⁵⁰ Insgesamt investiert er jedoch nicht im selben Maße Zeit und Mühe. Er lässt die anderen machen, mischt sich nicht ein, er diskutiert und recherchiert nicht mit, er stellt trotz seines Mißerfolgs mit der *ssh*- bzw. *screen*-Verbindung keine Fragen an die Runde, sondern scheint Informationen zu erwarten.⁵¹ Die mündlichen Tipps, die A48 erhält, helfen ihm nicht, sich einzuloggen, sondern „A2 kommt um den Tisch herum“, „will ihm in die Tastatur greifen“. Diese Art von Hilfe erinnert eher an die Konstellationen beim technischen Support der LUG als an das übliche Setting der fortgeschrittenen LPI-Lerngruppe.

Das zweite Materialbeispiel ist der oben erwähnte → *Bug Jam*, ein weltweit an mehreren Orten gleichzeitig stattfindender Termin, an dem vor der Fertigstellung einer neuen Version einer bestimmten → *Distribution* Fehler durch die Community behoben werden sollten. Ich habe eins der Treffen beobachtet und bin dort etwa vier Stunden geblieben, habe jedoch lediglich einen tatsächlich abgeschickten → *Bugreport* gesehen, was mir wenig vorkam.

Der Bug Jam findet an einem Samstag statt. Ein sehr pünktliches Kernteam findet sich um elf Uhr am Treffpunkt ein.⁵² Die nächsten drei Stunden vergehen mit verschiedenen Aktivitäten: Alle bauen ihre privaten Notebooks auf und sehen sich im → *Bugtracker* verschiedene bekannte Softwareprobleme an. Sie bauen neue, gespendete Festplatten in Computer ein, die an dem Treffpunkt zur Verfügung stehen, um darauf dann Tests durchzuführen. Zwei Personen packen mitgebrachte Aufkleber der Distribution aus und bieten sie den anderen an. Es gibt Probleme mit dem Beamer, die schließlich zwei von ihnen

⁴⁹Dies bringt mich auch zu der Interpretation, dass die humorvollen Äußerungen von A46 und A48 über A48s Distribution *Mandriva* sein Wissens- oder Aufmerksamkeitsdefizit überspielen. Damit wird möglicherweise auch eine Spannung bewältigt, da alle auf A48 warten müssen, der sich aber nicht aktiv um die Lösung seines Problems kümmert (vgl. Friedman/Friedman 2003: 9).

⁵⁰„Er sucht über seine grafische Paketverwaltung nach *sshd*, dann nach *ssh*. Er scrollt mit seiner USB-Maus durch die Liste und sieht sich an, was es dort gibt. Er sieht ein grünes Häkchen bei *openssh*.“

⁵¹„Das habt ihr mir vorhin nicht gesagt.“

⁵²Jemand hatte mich vorgewarnt, dass an dem Treffpunkt morgens eh noch niemand sei. Ich war darauf eingestellt, länger auf das Eintreffen der Beteiligten warten zu müssen, und war überrascht über ihre Pünktlichkeit.

lösen.⁵³ Sie wollen verschiedene „Installationsszenarien“ und die → *Live-CD* testen. Bezüglich der Live-CD stellt jemand fest, dass auf dem zentralen Server nicht die aktuellste Version angeboten wird.⁵⁴

Bei einer CD, von der sie heute eine Probe-Installation machen wollen, steht das Datum von vor 4 Tagen. A13 ist in einen Chat-Kanal eingeloggt, der den Namen der Distribution und das Wort „testing“ im Titel hat. Dort bittet er darum, die Updates von heute hochzuladen. Er sieht sich die Liste der eingeloggten Kommunikationsteilnehmer_innen an und spricht eine_n davon im Kanal gezielt an, ob er_sie das vielleicht machen könnte (LUG 2009l).

Bevor Upgrades auf die *neue* Version getestet werden können, müssen die *bereits installierten* Versionen auf den neuesten Stand gebracht werden, doch der Download dieser Updates, die nicht Gegenstand der Tests waren, läuft wegen mangelnder Bandbreite sehr langsam. Als eine Person versucht, ein Upgrade auf die neue Version nicht über das Internet, sondern von einer vorhandenen CD durchzuführen, startet doch der Download aus dem Internet, den die Person wieder abbricht.⁵⁵ Die Person holt per Chat Informationen von Dritten darüber ein, wie beim Upgrade direkt auf die CD zugegriffen werden kann, um zu erfahren, dass eine ganz andere CD dafür nötig wäre.

Die Teilnehmer des *Bug Jams* teilen ein gemeinsames Zeitgefühl: Einerseits *haben* sie die Zeit, sich einen Samstag lang für die Verbesserung einer Distribution einzusetzen. Andererseits ist ihnen im Umgang mit der investierten Zeit die Geduld gemeinsam, die mehrstündige Dauer der Vorbereitungen als selbstverständlichen Teil des Treffens in Kauf zu nehmen. Wie ich als Beobachtende schon in der LPI-Gruppe zwischen einem eigentlichen Lernziel und einem Abschweifen unterschieden habe, kam es mir auch hier so vor, als seien sie nicht mit dem Wesentlichen befasst, sondern mit langwierigen Vorbereitungen für die eigentlichen Tests. Die Beteiligten machen diesen Unterschied nicht mit derselben Deutlichkeit.

⁵³Der Beamer wird in der von mir beobachteten Zeit jedoch nicht für die gemeinsame Arbeit gebraucht.

⁵⁴Die Entwickler_innen laden ihre jüngsten Änderungen teilweise jede Nacht auf einen Server, von dem andere sie dann wiederum beziehen können.

⁵⁵„Dann kommt A13 zu A14. Sie sprechen darüber, warum der Rechner sich das Update nun nicht von der eingelegten CD holt, sondern doch aus dem Internet, was entscheidend länger dauert. Sie brechen das Update ab. A13 nimmt die CD nochmal raus, legt sie erneut ein. Sie lassen wieder die Paketverwaltung starten, doch als die Größe des Downloads aus dem Internet angezeigt wird, sagt A13 ‚verwirrend‘“ (LUG 2009l).

Die Vorbereitung der technischen Artefakte, das damit verbundene Warten, nicht funktionierende oder falsch bediente Technologien wirkten sich auch in der LUG strukturierend aus: Alte Hardware, die den Anforderungen eines modernen Linux-Desktops nicht mehr gewachsen ist, kann sehr viel Zeit kosten. In der folgenden Situation sind A1 und A54 verabredet, um auf A54s Notebook Linux zu installieren. Sie warten auf eine Reaktion des Computers, der für eine → *Live-CD* zu wenig Arbeitsspeicher hat:

A1 versucht, auf der extremen langsamen Live-CD durch ein Menü zu navigieren. Er sagt, er gebe dieser CD noch eine Viertelstunde, sonst wolle er es anders versuchen. A54 stimmt dem zu. (Ich halte es schon kaum noch aus, so lange dauert das hier.) A1: Er versuche seit fünf Minuten, über dieses Menü ein Terminal zu öffnen. (...) Plötzlich wird der Rechner leise und scheint anzuhalten. A1 und A54 beugen sich leicht vor und lauschen. A1 steht auf und sagt: „Okay, machen wir’s anders.“ Er verlässt wieder den Raum. A54 ärgert sich: „Scheiß Computer, was soll das? Mist!“ Sie stöhnt (LUG 2009h).

Nachdem verschiedene Versuche mit der Live-CD der von A54 gewünschten Distribution misslingen, installiert A1 schließlich eine andere Distribution. (Er entscheidet das selbst, ohne sich mit ihr darüber abzustimmen.) A54 ist es nicht gewöhnt, sich so lange mit ihrem Computer auseinanderzusetzen. Nachdem die Installation beendet ist und A1 ihr die wichtigsten Programme gezeigt hat, äußert sie sowohl Anstrengung als auch ein schlechtes Gewissen, seine Zeit so lange beansprucht zu haben.

A54 fragt (ich glaub zum zweiten Mal), ob er heute sowieso den Abend hier verbracht hätte. Er bejaht, dazu seien sie ja hier. A54 sagt: „Dreieinhalb Stunden. Wahnsinn.“ Sie sei so etwas nicht gewöhnt (LUG 2009h).

In vielen Praktiken der Techniknutzung kommt ein spezifisches Verhältnis zur Zeit in FLOSS-Communities zum Ausdruck, das wie alle gesehenen Habitusfragmente mehr oder weniger stark ausgeprägt ist. Dieser Umgang mit Zeit prägt die Treffen der Gruppen. Dazu gehört es erstens, eine Einschätzung von der Dauer von Arbeitsabläufen zu haben, die eingeübt werden kann, dennoch passieren Fehleinschätzungen immer wieder, unabhängig von Wissensstand und praktischer Erfahrung. Zweitens haben Nutzer_innen, die mehr Habitus verinnerlicht haben, eine höhere Akzeptanz für zeitliche Abläufe, die die Technologien ihnen „vorgeben“. Laut Judy Wajcman wohnt den Artefakten keine eigene temporäre Logik inne. Technologien brächten selbst keine Be- oder Entschleunigung hervor,

sondern vielmehr verschiedene Praktiken im Umgang mit Zeit und verschiedene Zeitgefühle, die nur im jeweiligen Kontext verstanden werden könnten (vgl. Wajcman 2008: 72). Hier in der LUG ist es Konsens, Sicherheitsupdates nicht zu verschieben, die Geduld für Downloads auch bei geringer Bandbreite aufzubringen. Andere Vorhaben werden solange zurückgestellt, bis der Rechner bereit ist. Drittens teilen sie eine Neugierde auf Neues oder auf unbekannte Software und halten die Augen danach offen. Stoßen sie im Arbeits- oder Lernprozess darauf, so antizipieren sie, ob die entdeckte Software ihnen künftig dienen kann. Falls ja, dann sind viele sofort bereit, dem Erlernen Zeit einzuräumen. Die Unterscheidung zwischen dem ursprünglichen Lerngegenstand und dem auf Weg Gelernten ist nicht so relevant wie das Lernen an sich. Die gemeinsame Definition des Lerngegenstandes wird sehr flexibel gehandhabt. Es ist für viele Konsens, dass sich die in Lernprozesse investierte Zeit lohnt: Zukünftige Kooperations- und Lernprozesse erfordern sowohl abrufbares Wissen als auch Erfahrungen, situative praktische Problemlösungskompetenz, um erfolgreich zu sein. An diesen Kompetenzen arbeiten sie, wenn sie antizipieren, dass es sich lohnt, mehr Zeit in ein Thema zu stecken als zunächst gedacht.

6.3.2. Neugierde auf Neues: Spontane Erweiterungen des Lerngegenstandes

In diesem Abschnitt soll die bereits mehrfach erwähnte, gängige Praxis aufgegriffen werden, sich spontan anderen oder weiteren Themen zu widmen als nur den ursprünglich vorgesehenen. Es wurde bereits gezeigt, dass die Beobachteten diese Praxis nicht als ein „Abschweifen“ von einem eigentlichen Gegenstand aufzufassen scheinen, sondern dass sie Experimente und Erweiterungen des Lerngegenstandes für einen unabdingbaren Teil des Lernprozesses halten.

A46 öffnet die Datei und springt direkt zu der Zeile. A39 stellt eine Zwischenfrage an A46, nämlich, wie er so schnell zu der Zeile komme. A46 und A47 erläutern, man könne einfach die Zahl eingeben. A39 will das gleich ausprobieren, schließt den *vi* nochmal, öffnet ihn wieder und springt zu der fraglichen Zeile (LUG 2009d).

In Szenen wie dieser geht es meines Erachtens darum, den Nutzen bei der Beobachtung fremder Routinen zu antizipieren: Jemand erledigt etwas besonders schnell, was einer anderen Person auffällt, die dann einhakt. Dieses Nachfragen ist tendenziell voraussetzungs-voll, da die fragende Person genug Erfahrungen haben muss, um ein Auge für die Routinen anderer und eine Einschätzung der Dauer der eigenen Routinen zu haben. Bei dieser Bereitschaft, spontan das Thema zu wechseln, habe ich abweichende Praktiken beobachtet.

Sowohl in der LUG als auch in der IT-Abteilung achteten manche darauf, bei einem Thema zu bleiben. „A44 will Fragen stellen, doch er wehrt sie ab, er könne sich nur auf eine Sache gleichzeitig konzentrieren“ (LUG 2009h). „Was mir schnell aufgefallen ist, ist dass A36 bemüht ist, am Thema zu bleiben und sich nicht zu verzetteln. Viele (...) neigen dazu, auf der Suche nach einer Lösung an etwas ganz anderem hängen zu bleiben. Er sagt dann, dass es eigentlich um das und das ging und wir da noch nicht fertig waren“ (Fir 2008d).

Der zweite häufige Anlass, Lerngegenstände spontan umzudefinieren, ist, dass die Person, die eine Frage hat, nicht die gleichen Programme verwendet wie die helfende Person. Oft haben Fragende bereits selbst versucht, mit ihrem Anliegen weiterzukommen und haben bereits ein bestimmtes Programm in Betrieb, an dem sich die Fragen orientieren. Aufgrund der Vielfalt an freien Programmen ist es nicht unwahrscheinlich, dass die helfende Person selbst eine andere Software verwendet. In der folgenden Situation wollen A7 und A8 eine Firewall erstellen. A7 hatte es bereits mit einem anderen Programm versucht, das A8 nicht kennt. A8 rät zu dem Programm, das er selbst benutzt und zu dem er Hilfestellung geben kann. Sie müssen jedoch das erste Programm und die angefangene Firewallkonfiguration zuerst löschen. Auch dazu hat A7 eine Software in Gebrauch, mit der sein Helfer nicht zurechtkommt, so dass wiederum A8 ein anderes Programm empfiehlt, das dann installiert wird.

Dann geht es um das Löschen von Konfigurationsdateien und dass die → *Paketverwaltung Adept* das nirgendwo leicht sichtbar anbietet. Der jüngere Mann recherchiert auf seinem Rechner nach einem Befehl dafür und regt sich über *Adept* auf: *Adept* findet das Firewallprogramm *guarddog* nicht. A8: „Ich traue diesem *Adept* nicht. Das ist primitiv. Die Ergebnisse sind nicht vorhanden.“ (Er meint wahrscheinlich die Suchergebnisse nach einem Programm, das verfügbar sein müsste.) Er sieht auf seinem Rechner online nach, in welchem → *Repository guarddog* liegt: *universe*. Dann sehen sie nach, ob dieses Repository auf A7s Rechner aktiviert ist. Doch, *universe* ist aktiviert. Aber die *Backports* nicht. „Ich kommentiere das hier mal wieder ein.“ A8 editiert jetzt in A7s Liste der Repositories herum. Er speichert sie und diktiert dann wieder dem älteren (A7): „*apt-get update*“. Bei Übergriffen auf die Tastatur von A7 sagt er: „Darf ich mal“ oder „mal kurz gucken“. A8 empfiehlt die Paketverwaltung *Synaptic*, die sie dann installieren. Er diktiert: „*apt-get install synaptic*“ (...) (LUG 2007d).

Die beiden verändern das anfängliche Szenario durch zwei Softwarewechsel, bevor sie sich der Konfiguration der Firewall zuwenden. In Lernprozessen von Linux ist diese Praxis häufig: Lernende bekommen von Erfahreneren bestimmte Programme empfohlen,

die immer von deren individuellen Routinen und durchlaufenen Lernprozessen abhängig sind. Die Hilfesituation kommt dann nicht durch einen Wechsel der Helfer_innen, sondern durch den Wechsel zu dieser empfohlenen Software zustande, ohne die Helfer_innen eben nicht unbedingt in einer situativen Expert_innenrolle wären. Die Zeit, die für Wechsel der Software aufgebracht wird, ist hier nötig, um eine Arbeitsgrundlage für das Helfen zu schaffen. Mehmet Göktürk and Görkem Çetin nennen das erste Zusammentreffen, das hier die Helfer_innen mit einer unbekanntem Software haben, „Out of Box Experience“ (Göktürk/Çetin 2007). Sie bemerken, dass gerade bei freier Software viele alternative Programme existierten, so dass die Wahrscheinlichkeit hoch sei, dass Nutzer_innen wieder abspringen, wenn sie nicht sofort mit einer Software zurechtkommen. Deshalb sei bei der Entwicklung besonders auf → *Usability* zu achten, damit eine Software Erfolg habe. „Furthermore, when the user has previous experience with other similar products, his/her expectations, existing factual and procedural knowledge affect initial usability experience to a great extent“ (vgl. ebd. 774). Die beobachteten Situationen, in denen Helfer_innen sich nicht auf die Software einlassen, die die fragende Person mitbringt, sprechen für die Gültigkeit dieser Thesen. Beim eigenen Erlernen einer Software bringen die Personen — wie am Beispiel *vim* gezeigt — dagegen sehr viel Geduld dafür auf, Einstiegshürden zu nehmen, auch wenn die Software nicht in erster Linie auf *Usability* ausgelegt ist.

6.4. Empirische Zwischenergebnisse

Die Analyse verschiedener Praktiken der Techniknutzung bezog sich auf drei Komplexe: auf die Nutzung textbasierter Programme, auf den kreativen Umgang mit Hard- und Software und auf das besondere Zeitgefühl, das Linux-Nutzer_innen teilweise zeigen.

Eine der wichtigen Distinktionslinien im Umgang mit freier Software ist die Verwendung textbasierter Programme. Dabei grenzen sich die Beobachteten teilweise explizit von Nutzer_innen grafischer Programme, aber auch von Nutzer_innen anderer textbasierter Programme ab. Sie rekurrieren auf Wertungen, die bestimmten Programmen zugeschrieben werden. Am Beispiel des Texteditors *vim* wurde deutlich, wie bestimmten Details Bedeutung beigemessen werden kann, z. B. *welche* Software jemand verwendet und wie effizient die Person sie einsetzt, ob sie den Aufwand auf sich genommen hat, sich besonders tief in die Konfigurations- und Bedienungsmöglichkeiten des Editors einzuarbeiten. Einerseits spielt Effizienz hierbei eine große Rolle: Durch Shortcuts üben Linux-Nutzer_innen sich

im effizienten Arbeiten, sie eignen sich Routinen an, die immer auch körperlich eingeübt werden, und schätzen genau die Effizienz, die bei dieser Aneignung im Zusammenspiel mit der Software entsteht. Praktiken der individuellen Umgestaltung von Software haben oft, aber nicht immer etwas mit Effizienzsteigerung zu tun: Während die Personalisierung von Software durch eigenes, ergänzendes Programmieren Arbeitsschritte automatisiert und beschleunigt, sind ästhetische Akte in Form von eigener Software Beispiele für einen oft humorvollen, souveränen Umgang mit Technologien jenseits von Effizienzgedanken. Auch der Habitus, durch den die Beteiligten sich für diesen oder jenen Editor entscheiden, lässt sich nicht allein mit der möglichen Effizienz oder Konfigurierbarkeit eines Programmes erklären. Manche zeigen eine große Faszination an der Konfigurierbarkeit *selbst*. Zudem spielen auch die verbreiteten Images der Programme eine Rolle und ob die Nutzer_innen einen Sinn dafür haben, dieses Image als relevant zu erkennen. Im Fall von *vim* wägen sie beispielsweise ab, ob sie sich auf den hohen anfänglichen Lernaufwand einlassen möchten. Wenn sie den entsprechenden praktischen Sinn haben, erkennen sie, dass der Editor deshalb so verbreitet ist, weil die meisten anderen angeben, dass es sich für sie gelohnt hat, den Umgang zu erlernen, *und* sie erkennen, dass sie damit ein prestigeträchtiges Werkzeug erlernen. Die Verwendung eines solchen Programmes stellt, wie gezeigt, auch einen Bezug zu denen her, die sich nicht an komplex zu bedienende Software heranwagen, das Unterscheidungsvermögen nicht haben oder die Zeit nicht investieren. Sie ist deshalb eine der präsentesten distinguierenden Praktiken der Softwarenutzung unter Linux. Kreative Zweckentfremdung von Software, „Hacking“, wird in den Communities mit einem besonders hohen Maß an Anerkennung belohnt, was diese persönlichen Umnutzungsideen als Kür der Softwarenutzung erscheinen lässt. Genau diese kreativen Äußerungen sind auch für weniger erfahrene Community-Mitglieder oder Neulinge besonders schwer zu entziffern bzw. überhaupt als findige Tricks wahrnehmbar.

Die später individuell anpassbare Software wurde an einigen Beispielen auf die von den Herstellerteams gesetzten ästhetischen Vorstellungen hin überprüft. Die Standardeinstellungen, die Software, z. B. durch Bebilderung, Klänge und Farbgebung mitgegeben werden, zeigen, dass Software keineswegs neutral ist, sondern dass sie bereits eingeschriebene Bedeutungen und Zuschreibungen enthalten kann. So wurden subtile Bezüge auf den proprietären Softwaremarkt und auf Geschlechterverhältnisse gezeigt. Je nachdem, auf welchen Ebenen sie in die Software eingestellt werden, lassen sie sich mehr oder weniger leicht identifizieren oder verändern.

In den Arbeitspraktiken der Linux-Nutzer_innen wurde deutlich, welche spezifische Ge-

duld und welches Zeitgefühl viele von ihnen bei der Softwarenutzung aufbringen. Je mehr Erfahrung sie haben, desto mehr sind sie bereit, sich auf die Dauer einzulassen, die das Erlernen und Bedienen von Software teilweise erfordert. Sie geben der Technologie die nötige Zeit für Updates, sie nehmen sich (auch spontan) die nötige Zeit für Neuentdeckungen und Ausprobieren, *wenn* sie davon ausgehen, dass sie dabei etwas Interessantes oder Nützliches lernen. Sie nehmen dabei in Kauf, dass die Inbetriebnahme einer Software oft länger dauern kann als eine nichttechnische Lösung, so groß ist der Anreiz, Neues zu erproben. Neulinge nehmen die Dauer eines Prozesses bis zu einer Problemlösung oft als sehr langwierig wahr. Das Erlernen dieses Zeitgefühls ist ein zentrales Habitusfragment, das zur Sozialisation in der Szene gehört. Es ist eine Voraussetzung dafür, dass die Community-Mitglieder sich technologischen Herausforderungen stellen und ist eng mit der situativen, praktischen Problemlösungskompetenz verbunden.

6.5. Theoretische Reflexion: Technikverständnisse in FLOSS-Communities

In den Praktiken, mit denen Linux-User_innen Technologien nutzen und anpassen, zeigen sich bestimmte Technikverständnisse, die ich vor dem Hintergrund techniksoziologischer Ansätze verdeutlichen werde.

In der Techniksoziologie gibt es laut Nina Degele zwei Zugänge zur Analyse von Technik: Einerseits wird untersucht, wie Technik menschliches Verhalten determiniert. Dies sei bis in die 1970er Jahre das vorherrschende Forschungsinteresse gewesen. Andererseits habe sich ab den 1980er Jahren der Fokus der Analysen zur Genese von Technik verschoben, die nun als gesellschaftliches Produkt gedacht wird, das Sozialität mitprägt. Mit diesem Zugang sind z. B. die oben erwähnten Vorannahmen über Nutzer_innen untersucht worden, die ihre Spuren in den Artefakten hinterlassen, oder die Umgangs- und Aneignungsweisen von Nutzer_innen (vgl. Degele 2002: 7 f.). Im Folgenden beziehe ich mich zuvorderst auf Autor_innen, die ebenfalls mit den Ansätzen Bourdieus arbeiten.⁵⁶ Ihre Argumentationen stellen die Anwendbarkeit der Bourdieu'schen Begriffe in techniksoziologischen Arbeiten in Frage. In diesem Abschnitt greife ich zwei Stränge auf: Erstens beziehe ich mich auf Ansätze, die hinterfragen, ob Technologien (reinen) Objektstatus haben. Sie beschäftigen

⁵⁶Für breiter angelegte techniksoziologische Einführungen vgl. z. B. Degele 2002; Schulz-Schaeffer 1999: 409 ff.

sich damit, ob ihnen eigene Handlungen zugeschrieben werden können und welche Anteile sie am menschlichen Handeln haben. Zweitens geht es darum, ob der Habitus außer Kraft gesetzt wird, wenn Personen Artefakte bedienen, die nach expliziten Regeln funktionieren. Die theoretischen Positionen nutze ich dazu, die Spezifika meiner empirischen Ergebnisse zu Praktiken der Techniknutzung und den dahinter stehenden Technikverständnissen herauszuarbeiten.

Zusammenfassend wurde in diesem Kapitel gezeigt, dass die Praktiken der Techniknutzung von Linux-Nutzer_innen sehr divers sind und dass sie auf heterogene Technikverständnisse hinweisen. Ihre Bandbreite lässt sich wie die verschiedenen Habitus höchstens fragmentarisch in einem imaginierten Kontinuum denken. Am einen Ende dieses Kontinuums wären versiertere Nutzer_innen, deren Verhältnis zu den technischen Artefakten durch verschiedene Elemente charakterisiert werden kann: Sie schauen hinter die „Kulisse“ der grafischen Oberfläche, sie wissen, wie viele Zugewinne textbasierte Programme beim Arbeiten bringen können und begreifen das Betriebssystem in seiner Struktur. In effizient eingeübten, auch körperlichen Praktiken werden sie eins mit ihrem Programm, rufen Tastaturkürzel und Wissen über Konfigurationsmöglichkeiten ab. Diese Effizienz verbinden sie teilweise mit selbst programmierten Skripten. Sie lassen sich souverän auf die Zeit ein, die Linux ihnen für Lernprozesse, Experimente und Wartung abverlangt. Sie begreifen die Hardware, vor allem aber die Software als gestaltbar und bestimmen die Abläufe sehr detailgenau. Am anderen Ende dieses Kontinuums wären Nutzer_innen, deren Verhältnis zu den Rechnern tendenziell eher so aussieht: Sie arbeiten ausschließlich mit grafischen Programmen, können oder wollen sich darüber hinaus nicht mit Strukturen des Systems beschäftigen oder selbst Software umgestalten. Sie investieren ihre Zeit nicht in das Erlernen bestimmter, prestigeträchtiger Programme und haben nicht unbedingt ein Auge für kleine Handgriffe, die sie sich bei anderen in der Praxis abgucken. Sie lassen das Betriebssystem weitaus näher an den mitgelieferten Standards, was die Ausstattung mit Programmen angeht.

Vor dem Hintergrund dieser communityinternen Heterogenität eignen sich nun die Linux-Nutzer_innen, die über viele oder besonders ausgeprägte Habitusfragmente verfügen, dazu aufzuzeigen, welche spezifischen Züge die Techniknutzung von Linux-User_innen ausmachen können. Im Abgleich mit den Theorien möchte ich vier Komponenten darlegen, die Teil der Technikverständnisse von Linux-User_innen sind.

1. Je mehr der beobachteten Habitusanteile Linux-Nutzer_innen inkorporiert haben, desto

weniger sehen sie Artefakte als autonom handelnde oder entscheidende Maschinen, weil sie in ihrer Praxis erfahren, wie tiefgehend sie selbst in die Software eingreifen können.

Frank Hillebrandt und Ingo Schulz-Schaeffer haben sich mit der Frage auseinandergesetzt, wie Pierre Bourdieus Konzepte für techniksoziologische Untersuchungen genutzt werden können. Sie fragen, ob man vergegenständlichte technische Artefakte (in meinem Fall die Hard- und Software) untersuchen kann, indem man auf Bourdieus Vorstellung von objektivierter Geschichte zurückgreift. Unter objektivierter Geschichte versteht Bourdieu „die im Laufe der Zeit in den Dingen (Maschinen, Gebäuden, Monumenten, Büchern, Theorien, Sitten, dem Recht usw.) akkumulierte Geschichte“ (Bourdieu 1997b: 28). Er selbst hat sich kaum mit dem Thema Technik beschäftigt. Betrachtet man Technik als Teil der objektivierten Geschichte kann sie einerseits ein „Distinktionsmittel zur Positionierung der Akteure im sozialen Raum“ darstellen (Schulz-Schaeffer 2004b: 47; vgl. analog dazu Bourdieu 1998b: 17 ff.). Technische Artefakte seien „Teile der objektivierten Außenwelt des Menschen“ (Hillebrandt 2004: 35) und damit Objekte. Dennoch verfügten sie andererseits über Eigenschaften, die sie „eigene Handlungen“ ausführen ließen, so dass sie nicht „lückenlos als kausale Wirkungszusammenhänge beschrieben werden können“ (ebd. 36). An einem Beispiel aus meinem Feld kann der Gedankengang wie folgt nachvollzogen werden: Wenn Informationen über das Internet übertragen werden sollen, ermitteln → Router den Empfänger und stellen die Informationen zu. Wenn auf dem Weg ein Router ausfällt, ermitteln die benachbarten Router automatisch einen neuen Weg über einen anderen Router. Router stellen also fest, dass sie diesen Weg berechnen müssen, wenn sie keine Antwort erhalten.

Dieser Gedanke geht auf die Akteur-Netzwerk-Theorie (ANT) von Michel Callon und Bruno Latour zurück, zu der hier kurz ausgeholt werden muss. Die ANT hat die Trennung von menschlichen Subjekten und technischen Objekten radikal hinterfragt. Latour hat empirisch gezeigt, wie stark Artefakte an Handlungen beteiligt sind und grenzt sich mit seiner Theorie sowohl von technik- als auch von sozialdeterministischen und sozialkonstruktivistischen Perspektiven ab (vgl. Belliger/Krieger 2006: 20 ff.):

„Die beiden Mythen vom neutralen Werkzeug unter vollständiger menschlicher Kontrolle und von autonomem Geschick der Technik ohne jede Chance menschlicher Beherrschbarkeit sind symmetrisch. Doch meistens liegt eine dritte Möglichkeit näher: die Schöpfung eines dritten Ziels, das keinem der beiden ursprünglichen Handlungsprogramme mehr entspricht (...). Diese Unbestimmtheit der Ziele habe ich (...) eine Übersetzung genannt. (...) Unter Übersetzung verstehe ich eine Verschiebung,

(...) es ist die Schöpfung einer Verbindung, die vorher nicht da war und die beiden ursprünglichen Elemente oder Agenten in bestimmtem Maße modifiziert“ (Latour 2002: 217 f.).

Latour geht davon aus, dass Menschen, Dinge und Natur in Netzwerken miteinander verbunden sind (vgl. Ruffing 2009: 9): Er spricht von „Kollektiven aus menschlichen und nicht-menschlichen Wesen“ (ebd. 211), die gemeinsam handeln. Welche besondere Qualität er diesen Kollektiven beimisst, macht er an verschiedenen Beispielen deutlich: Seiner Ansicht nach ist ein Mensch ohne Schusswaffe ein anderer als ein Mensch mit Schusswaffe. Doch auch eine Schusswaffe verändere sich, wenn sie auf jemanden gerichtet würde.⁵⁷ Aus dem Zusammentreffen beider entstehe ein neuer „Hybrid-Akteur“ (vgl. Latour 2002: 218). Sein Anliegen ist es, die genaue Zusammensetzung von Handlungen durch diese Perspektive sichtbar zu machen: Wer ist an einer Handlung wie beteiligt? Der Autor geht so weit, die Verantwortung für Handlungen unter den an einem solchen Kollektiv beteiligten Akteuren zu verteilen (vgl. ebd. 219), indem er nachzeichnet, wie „Handlungsprogramme“ an Artefakte *delegiert* werden: Er beschreibt Betonschwellen zur Kontrolle von Fahrverhalten oder den viel zitierten Berliner Schlüssel⁵⁸ als Aktanten, zu denen (Teil-)Handlungen und Verantwortung verschoben werden. Ein entsprechendes Beispiel aus der Systemadministration ist → *Monitoring*-Software: Systemadministrator_innen konfigurieren sich ihre individuell angepassten → *Server* mit der Software, die sie brauchen und mit Software, die automatisiert überwacht, ob die nötigen Dienste erreichbar sind. Ist dies nicht der Fall, benachrichtigt die → *Monitoring*-Software den Menschen, z. B. per E-Mail oder per SMS. Derart ausgestattete Systemadministrator_innen haben wichtige Tätigkeiten auf ziemlich verlässliche Weise delegiert und können sich anderem widmen. Sie und ihre Monitoring-Systeme bilden Hybrid-Akteure, die anders aufgestellt sind, als Administrator_innen, die ständig von Hand überprüfen müssten, ob einzelne Server oder Dienste laufen (vgl. auch die Übertragung der ANT auf FLOSS-Projekte von Morner/Lanzara 2007).

Welche genaue Rolle Technik spielt, ist laut Latour schwer auszumachen: Jede einmal

⁵⁷ „Mit der Waffe in der Hand bist du jemand anderes, und auch die Waffe in deiner Hand ist nicht mehr dieselbe. (...) Der gute Bürger wird zum Schurken, der Gangster zum Killer, der stumme Revolver zu einer abgefeuerten Waffe, (...) das Sportgerät zum Tötungsinstrument“ (Latour 2002: 218).

⁵⁸ Latour (1996) hat eine bestimmte Schlüsselform untersucht, die Hausbewohner_innen durch ihre Machart zwang, die Haustür in den Nachtstunden abzuschließen. Der Schlüssel hatte zwei Bärte und konnte nach dem Aufschließen der Tür nur abgezogen werden, wenn er durch das Schloss geschoben wurde und die Tür von innen wieder verschlossen wurde (Für eine Zusammenfassung siehe auch Ruffing 2009: 31 f.).

funktionierende delegierte Handlung werde auf Dauer unsichtbar, bis ein technisches Problem auftritt. Dieses Verschwinden der funktionierenden Artefakte aus der Wahrnehmung nennt der Autor *Blackboxing*, sie werden zu einer Selbstverständlichkeit, die die Nutzer_innen für die Funktionsweise eines Artefaktes blind mache. Erst, wenn ein Artefakt nicht mehr wie vorgesehen funktioniere, werde wieder enthüllt, in wie viele Einzelteile es zerlegbar ist und was diese Einzelteile zum Funktionieren beitragen (vgl. Latour 2002: 222 ff.; siehe auch Belliger/Krieger 2006: 43 f.). Latour möchte genau diese Details sichtbar machen, das *Blackboxing* nicht hinnehmen.

Die Akteur-Netzwerk-Theorie ist oft kritisiert worden, vor allem weil sie Dingen Handlungen zuschreibt (vgl. die Zusammenfassung der Kritik in Ruffing 2009: 113 ff.). Auch Hillebrandt merkt kritisch an, dass Latour mit der Behauptung einer Symmetrie von menschlichen Akteuren und nicht-menschlichen Aktanten die Eigenmächtigkeit technischer Artefakte überspitze. Dennoch regten Artefakte durch eingebaute Handlungsaufforderungen Praxis an.⁵⁹ Artefakte seien dann nicht lediglich neutrale Bestandteile von Situationen, in denen Praxisformen entstehen, sondern brächten diese Situationen mit hervor (vgl. Hillebrandt 2004: 37 f.). Hillebrandt wirft die Frage auf, wie eigenmächtig technische Artefakte an sozialer Praxis- und Strukturbildung beteiligt sind (vgl. ebd. 33). Dabei trifft er eine Unterscheidung zwischen trivialen Artefakten (z. B. einem Mixer), deren Handlungsprogramme eindeutig determiniert seien, weil sie nur begrenzte Handlungen vorsehen, und nicht-trivialen Techniken (z. B. Software), die sich — wie im Router-Beispiel — selbst verwalteten. Bei Pierre Bourdieu entfalte die objektivierte Geschichte keine Eigendynamik, sondern sie sei immer auf Akteur_innen und ihre Habitus angewiesen um zu wirken. Nicht-triviale technische Artefakte entfalten dagegen Eigendynamiken. Um ihre Praxisrelevanz zu untersuchen, hält Hillebrandt es für nötig, den Bourdieu'schen Begriff von objektivierter Geschichte weiterzuentwickeln: Die Möglichkeit, dass technische Artefakte aktiv werden und Praxis generieren, müsse von der Soziologie mitgedacht werden. Dazu sei es nötig, die „Selbstorganisation“ von Artefakten⁶⁰ adäquat zu beschreiben und ihre Wirkungen auf Praxisformen herauszuarbeiten (vgl. ebd. 36). Nicht nur Akteur_innen hätten durch ihren Habitus, der inkorporierte Geschichte ist, bestimmte „Handlungsdispositionen“, sondern auch softwaretechnische Artefakte. Der Autor plädiert für eine klare Differenzierung bei der Untersuchung von praxisgenerierenden Effekten technischer Arte-

⁵⁹Eine solche Handlungsaufforderung könnte ein Knopf sein, der quasi dazu auffordert, gedrückt zu werden.

⁶⁰„Die Wahl, welche Handlungsoption genutzt wird, bleibt den technischen Artefakten überlassen“ (Hillebrandt 2004: 41).

fakte: Einerseits seien Habitus zu untersuchen, die durch Inkorporierung bestimmte technikbezogene Praxisformen hervorbringen. Andererseits würde ein (zu bestimmender) Anteil von Praxisformen von den Artefakten produziert. Ohne diese Ergänzung verliere der Habitus seine Erklärungskraft für die Genese und Reproduktion von Praxis, so dass die Techniksoziologie sich darauf einlassen solle, in Artefakten implementierte Handlungsdispositionen mitzudenken (vgl. ebd. 40 f.).

Die Perspektiven von Hillebrandt und Latour helfen, um Technikverständnisse von FLOSS-Communities zu konkretisieren. Die Vorstellung, das Computer sich selbst organisieren, ist gerade bei versierteren Linux-Nutzer_innen *nicht* zu sehen. Je mehr Einblicke sie in die Funktionsweise des Betriebssystems haben, desto weniger drücken ihre Praktiken aus, dass sie glauben, Software entfalte Eigendynamiken.

Dies lässt sich am bereits angeführten Beispiel von Routern zeigen: Linux-Nutzer_innen, die verstehen, wie die Übermittlung von Daten durch das Internet funktioniert, sehen die Berechnung eines alternativen Weges für die Informationen nicht als autonome Entscheidung eines Routers. Sie wissen, dass die Software so geschrieben ist, dass der Router durch den Versand von Testpaketen seine „Nachbarn“ ermitteln kann, anhand von Antworten die Geschwindigkeit der Verbindung berechnen kann und so die kürzesten oder schnellsten Pfade für Routen berechnet und sie regelmäßig aktualisiert. Sie wissen auch, wie sie in die → *Routingtabellen* eingreifen können, wie sie also selbst definieren können, welche Informationen welchen Weg nehmen sollen. Für FLOSS-Communities kann zwar nicht von einem einheitlichen Technikverständnis gesprochen werden. Es gibt jedoch viele Akteur_innen, die — in oft langjährigen Lernprozessen — ein Kapital akkumuliert haben, das sich genau dadurch auszeichnet, dass sie die Regelzusammenhänge, auf denen die Artefakte basieren, kennen. Darüber hinaus verfügen sie über ein spezifisches soziales Kapital, das über die gemeinsame inhaltliche Beschäftigung mit freier Software akkumuliert wurde. Daraus ergibt sich ihre Möglichkeit, die Software weiterzuentwickeln. Wenn sie darüber staunen, was Software alles automatisiert (und nicht autonom) erledigen kann, dann vor dem Hintergrund, dass Menschen sie so programmiert haben; in einem solchen Staunen steckt Anerkennung für die Softwareentwickler_innen, nicht Ehrfurcht vor einer autonom handelnden Technologie. Akteur_innen in FLOSS-Communities setzen — je nach ihrem individuellen Kapital — ihr Regelwissen über Hard- und Software produktiv ein, um technische Probleme zu beheben, indem sie Software in ihrer Konstruktionsweise verändern. Dadurch sind viele von ihnen in der Lage, die Regeln zu verändern, die Technik zu *beherrschen*. Hillebrandts Vorstellung von Handlungsdispositionen technischer Artefakte

mag demnach eher für die nicht so versierten Linux-Nutzer_innen gelten. Sie greift genau da *nicht*, wo viel linuxspezifisches Kapital angehäuft wurde.

Inwiefern lässt sich Latours Gedanke der Hybrid-Akteure auf mein Forschungsfeld anwenden? Kann in FLOSS-Communities von Kollektiven aus menschlichen und nicht-menschlichen Wesen, von Hybrid-Akteuren mit einer eigenen Qualität gesprochen werden? Die technischen Artefakte lassen sich nicht wegdenken — sie sind konstitutiver Bestandteil von Community-Treffen. Weder sind LUG-Treffen (abgesehen von Grillparties oder Weihnachtsfeiern) ohne Linuxrechner vorstellbar, noch Community-Events, die softwaregestützt organisiert werden und auf denen die Technologien im inhaltlichen Mittelpunkt stehen. Mehr noch, FLOSS-Communities konstituieren sich erst durch Software. Alle sozialen Beziehungen sind über das gemeinsame technische Interesse vermittelt. Sie werden hier nicht als *ausschließlich* technisch bezeichnet; die gemeinsame Philosophie hinter freier Software, die Freundschaften, die entstanden sind, haben jedoch alle den Bezugspunkt Linux. Durch die gemeinsame Nutzung von freien Technologien für die Kooperation tragen technische Artefakte einen großen Anteil an den zwischenmenschlichen Beziehungen, sie stabilisieren sie: Ohne → *Jabber*, → *IRC* oder → *screen* wären bestimmte Praktiken unmöglich, die die Interaktion und Kooperation der Beteiligten zu einem Teil ausmachen. Christopher Kelty nennt FLOSS-Communities aus diesem Grund rekursive Öffentlichkeiten: Sie entwickeln die Technologien für Kooperation und Kommunikation, über die sie sich als Gruppen konstituieren, über die und mittels derer sie kommunizieren (vgl. Kelty 2008b: 3). Dies gilt in besonderem Maße für verteilte, online kooperierende Gruppen, ist aber auch offline in der LUG beobachtbar.

Zudem tragen Technologien selbst dadurch, wie sie gemacht wurden, dazu bei, Arbeitsweisen und Zeit zu strukturieren, ganz grundsätzlich, aber auch für Linux-Nutzer_innen. Ein Linuxrechner überprüft automatisch, ob es Software-Updates gibt und fordert die Nutzer_innen dazu auf, sie zu installieren. → *Monitoring*-Software schlägt automatisch Alarm, wenn etwas nicht stimmt und menschliche Eingriffe erforderlich sind. Bei der Automatisierung von Abläufen kann hier von Delegation im Sinne Latours gesprochen werden. Was erfolgreich delegiert wurde, unterliegt auch dem *Blackboxing*, das Latour beschreibt: Alles, was „läuft“, erfordert keine Aufmerksamkeit und wurde unter Umständen nach der Implementation in seinen Details zum Teil vergessen. Erst, wenn etwas nicht funktioniert, geht es darum, die Black Box zu öffnen, was — wie in Kapitel 5 gezeigt — mit hoher situativer praktischer Problemlösungskompetenz um so schneller gelingt. Diejenigen, die das spezifische Zeitgefühl haben, die Geduld aufbringen, sich auf Prozesse mit

manchmal unvorsehbarer Dauer einzulassen, gestehen dem Arbeitsprozess eine eigene Dauer zu. Darin schätzen sie nicht nur ihre eigenen Kompetenzen ab, sondern immer auch die Zeitlichkeit, die den technischen Artefakten eingeschrieben ist: Bandbreitenbeschränkungen, vorhandene Rechenkapazitäten, eingeblendete Informationen etc. Sie gestatten den Technologien somit zu einem gewissen Grad einen eigenen Anteil an der Interaktion, wenn sie wissen, dass sie momentan nicht eingreifen können. Sie wird auf die Langsamkeit eines alten Rechners Rücksicht genommen, nicht jedoch auf die Ineffizienz einer Software. Sie wissen jedoch, dass die Artefakte nicht autonom handeln, sondern so konstruiert wurden und grundsätzlich veränderbar sind. Die Verantwortung für das, was an die Technologien delegiert wird, verbleibt jedoch bei menschlichen Akteur_innen.

2. Software kann als Distinktionsmittel innerhalb von FLOSS-Communities vor allem dann eingesetzt werden, wenn sie wegen ihrer ausgefeilten technischen Funktionalitäten ein besonderes Prestige zugeschrieben bekommt und Nutzer_innen souverän mit diesen Funktionalitäten umgehen können.

Auch Ingo Schulz-Schaeffer problematisiert Bourdieus Konzepte von objektivierter Geschichte und Habitus im Zusammenhang techniksoziologischer Arbeiten. Er unterscheidet zwischen zwei verschiedenen Perspektiven, dem Routinenaspekt und dem Ressourcenaspekt⁶¹ von Technik. Forscher_innen, die den Routinenaspekt untersuchen, interessieren sich dafür, wie technische Artefakte als Distinktionsmittel in Routinen eingebunden sind. Sie untersuchten die praktischen Schemata, die Akteur_innen dazu disponieren, „das technische Regelwissen habitusabhängig in je bestimmter Weise auszudeuten und anzuwenden“ (Schulz-Schaeffer 2004b: 64).⁶² Bourdieus Ansatz eigne sich vor allem für die Untersuchung dieses „Routinenaspekts“ von Technik: Distinguierende Praktiken unter Rückgriff auf technische Artefakte beruhten in erster Linie auf zusätzlichen Bedeutungszuschreibungen an Artefakte, die über ihre technische Funktionalität hinausgingen und eine ästhetische Stilisierung erlaubten. Ob sich Artefakte als Distinktionsmittel oder gar als Statussymbole eignen oder nicht, hängt laut Schulz-Schaeffer nicht primär mit ihren technischen Eigenschaften zusammen, sondern mit ihren Kosten und ihrer öffentlichen Wahrnehmbarkeit (vgl. ebd. 50).

Was kann Schulz-Schaeffers Blickwinkel zur Analyse von Techniknutzungspraktiken in FLOSS-Communities beitragen? Die Ergebnisse der Empirie bestätigen, dass der „Rou-

⁶¹Der Ressourcenaspekt wird weiter unten ausgeführt.

⁶²vgl. für eine ausführlichere Darstellung der Dualität von Ressourcen und Routinen Schulz-Schaeffer (1999).

tinenaspekt“ von Technik in dieser Form beobachtbar ist: Artefakten werden Konnotationen zugeschrieben, über die Software ein Prestige erhält, das in der Nutzung bestimmter ästhetisch stilisierter Programme aktualisiert und öffentlich wahrnehmbar wird. Software und Softwarenutzung werden als Distinktionsmittel eingesetzt. Anders als bei Schulz-Schaeffer haben diese Bedeutungszuschreibungen jedoch sehr viel mit der technischen Funktionalität der Software zu tun. Es sind in der Regel „mächtige“ Programme, die zu Statussymbolen gemacht werden. Sie entsprechen verschiedenen Kriterien, wie z. B. Effizienz, Erfordernissen an Rechenkapazitäten, Möglichkeiten, remote (also textbasiert oder webbasiert) zu arbeiten, → *Usability*, Schnittstellen für andere Software. In FLOSS-Communities liegt hier der Clou: Nicht Kosten oder Prestige machen ein „Statussymbol“ aus, sondern technische Kriterien führen zu einem Prestige, das in erster Linie diejenigen wahrnehmen können, die den entsprechenden praktischen Sinn dafür inkorporiert haben.

3. Eine komplexe Technologie wie ein Betriebssystem kann als zuverlässige Ressource pro-spektiven Handelns nur unter Einsatz eines Habitus eingesetzt werden, der bereits zu weiten Teilen inkorporiert wurde.

Laut Schulz-Schaeffer ist es neben der Untersuchung des Routinenaspekts von Technik ein spezifisch techniksoziologisches Erkenntnisinteresse, die soziale Bedeutung von Technik *in ihrer Eigenschaft als Technik* zu untersuchen: Das Besondere an technischen Artefakten sei nämlich, dass sie von Akteur_innen gezielt eingesetzt würden, um immer die gleichen Effekte hervorzurufen. Technik basiere auf Vorhersehbarkeit. Aus diesem Grund nennt er technische Artefakte „Ressourcen pro-spektiven Handelns“. Solche Ressourcen müssten auf expliziten, regelgeleiteten Ereigniszusammenhängen basieren um zu gewährleisten, dass zukünftige Wirkungen zuverlässig und wiederholbar eintreten (vgl. ebd. 57 ff.). In Bezug auf den Begriff der objektivierten Geschichte liegt für Schulz-Schaeffer genau hier der Unterschied zwischen anderen kulturellen Gütern und technischen Artefakten: Andere kulturelle Artefakte (z. B. Verwandtschafts- oder Geschlechterbeziehungen) ließen sich als historisch gewachsen verstehen, sich „als verfestigte Formen vorgängiger Praktiken erklären“ (ebd. 54), was für Technik aufgrund ihrer Konstruiertheit nach ausdrücklichen Regeln nicht erschöpfend gelte.

„Als Ressourcen pro-spektiven Handelns sind Techniken künstlich eingerichtete Ablaufzusammenhänge, mit deren Hilfe sich hinreichend zuverlässig und wiederholbar bestimmte Effekte erzielen lassen (...). Es ist der Aspekt der expliziten Vorherseh-

barkeit bestimmter Ereignisketten, durch den Techniken zu Ressourcen prospektiven Handelns werden können“ (Schulz-Schaeffer 2004b: 54).

Der Habitus lasse die Akteur_innen seine Genese und Geschichte vergessen. Er folge keinen expliziten Regeln, so dass habituelle Praktiken und die vom Habitus hervorgebrachten Objektivierungen kontingent, aber nicht zuverlässig produzierbar seien. Wie technische Artefakte welche Wirkungen hervorrufen, sei dagegen bewusst konstruiert und bekannt. Man müsse nur die expliziten Gebrauchsanweisungen befolgen, um Technik als Ressource vorausschauenden Handelns einzusetzen. Schulz-Schaeffer spricht aus diesem Grund von einer „radikale[n] Entwertung der praktischen Könnerschaft des Habitus“ (ebd. 61). Alle Akteur_innen hätten prinzipiell die gleiche Möglichkeit, Technik als Ressource vorausschauenden Handelns einzusetzen. Diese egalisierende Wirkung von Technik entwerte differentielles kulturelles Kapital, da eine Entkopplung der Regeln technischen Funktionierens von den Gebrauchsregeln stattfinde.⁶³ Schulz-Schaeffer resümiert, dass Bourdieus Begriffe sich nicht eignen, um Technik als Ressource vorausschauenden Handelns zu untersuchen. Er erinnert jedoch daran, dass sich der Ressourcenaspekt nicht getrennt vom habitusgeleiteten Routinenaspekt analysieren lässt: Empirische Arbeiten müssten beide Aspekte einbeziehen und untersuchen, in welchem Umfang egalisierende Wirkungen von Technik beobachtbar sind und in welchem Umfang sie durch bestimmte technikbezogene Habitus unterlaufen werden (vgl. ebd. 61 ff.).⁶⁴

Schulz-Schaeffers Position hilft dabei, den Nutzen Bourdieus für techniksoziologische Arbeiten differenzierter zu sehen. Inhaltlich ziehe ich seinen Ressourcenaspekt als Kontrastfolie heran, da meine Erhebungen etwas anderes zeigen: Die Habitusfragmente von Linux-Nutzer_innen werden *nicht* dadurch entwertet, dass offen dokumentiert ist, wie freie Software als Ressource prospektiven Handelns eingesetzt werden kann: Jeder kann zwar nachlesen, wie sie zu bedienen ist, sogar wie sie programmiert ist. Ein Betriebssystem ist allerdings so komplex, dass genau diese Wissensaneignung ein Prozess ist, der Jahre

⁶³ „Man muss nicht über Jahre und Jahrzehnte in die Denk- und Handlungsschemata eines bestimmten Habitus einsozialisiert und eingeübt sein, um über dieses Wissen zu verfügen. Man braucht nur die mündlich oder schriftlich ausgedrückten Gebrauchsanweisungen zu befolgen. (...) Dort, wo die gemeinsamen Dispositionen die Praxis strukturieren, ist es der differentielle Besitz solcher praktischer Könnerschaft, der den Inhabern eines anderen Habitus verschlossen bleibt. Wo dagegen nur explizit verfügbares Wissen erforderlich ist, verliert dieser Effekt der einverlebten Geschichte seine Wirkung“ (Schulz-Schaeffer 1999: 61).

⁶⁴ Er bemerkt auch, dass unter Bedingungen ungleicher (ökonomischer oder kultureller) Kapitalverteilung das egalitäre Potenzial von Technik meist theoretisch bleibt. Forscher_innen müssten deshalb auch andere Faktoren einbeziehen, die egalisierende Wirkungen von Technik unterlaufen, die jedoch nicht Gegenstand seines Artikels sind.

erfordert. Wie am Beispiel von situativer praktischer Problemlösungskompetenz gezeigt wurde, ist auch nicht nur explizit verfügbares Wissen nötig, um mit auftretenden Schwierigkeiten umzugehen, obwohl das Betriebssystem nach klaren Regeln aufgebaut ist. In FLOSS-Communities entscheiden einige zentrale und differenzierend wirkende Habitusfragmente und die darüber akkumulierten Kapitalformen darüber, ob Nutzer_innen die Software als Ressource einsetzen können, um kalkulierbare, zuverlässige Wirkungen zu erzielen. Dazu zählt die Problemlösungskompetenz, die langjährige Erfahrung erfordert, genau so wie die Fähigkeit zum textbasierten Arbeiten, die zwar in Handbüchern dokumentiert ist, aber heute nicht standardmäßig von Computernutzer_innen erlernt wird. Die „praktische Könnerschaft des Habitus“ wird keineswegs entwertet, sondern genau solche Habitusfragmente machen Differenzen aus.

Schulz-Schaeffer arbeitet mit drei Prämissen, die ich vor dem Hintergrund des Beobachteten nicht teilen kann: Er setzt erstens voraus, *dass* Technik-Nutzer_innen Handbücher lesen. Linux-Nutzer_innen, vor allem die weniger versierten, versuchen jedoch zunächst, eine Software intuitiv zu bedienen. Erst, wenn sie an Grenzen stoßen, ziehen sie Hilfe hinzu. Die individuellen Grenzen liegen an unterschiedlichen Orten. Zweitens setzt er voraus, dass Handbuchlektüre dazu führt, dass bei der Bedienung eines Artefaktes die erwünschte und voraussagbare Wirkung eintritt. Dies trifft auf die Nutzung von freier Software ebenfalls nur bedingt zu: Zum einen ist nicht jede freie Software fehlerfrei und in ähnlichem Umfang dokumentiert. Zum anderen hängen Programme als Mosaikstückchen eines Betriebssystems meist voneinander ab. Eine Dokumentation deckt jedoch selten die nötigen anderen Voraussetzung mit ab.⁶⁵ Drittens bezieht sich Schulz-Schaeffer auf Nutzer_innen, die von der Technikentwicklung abgekoppelt sind. In diesem Kapitel wurde jedoch deutlich, dass viele Linux-Nutzer_innen Software aktiv gestalten. Die Trennschärfe zwischen Entwickler_innen und Nutzer_innen ist nicht gegeben. Die Entwicklung oder Modifikation von Software, auch kleinere Eingriffe in das Betriebssystem, sind konstitutive Bestandteile der Techniknutzung in FLOSS-Communities. Genau die Habitusfragmente, die versierte Linux-Nutzer_innen ausmachen, sind somit diejenigen, die Schulz-Schaeffer nicht berücksichtigt, denn sie erfordern es *gerade*, über Jahre einsozialisiert zu sein, um den Habitus zu erwerben, der es erlaubt, souverän mit den Funktionsregeln des Betriebssystems umzugehen. Die regelbasierte Konstruktion der Software macht den Ha-

⁶⁵Ein Beispiel: Eine Dokumentation zu einem Programm für Website-Statistiken enthält eine Installationsanleitung und Beschreibungen der Konfigurationsmöglichkeiten. Das Wissen über die Konfiguration eines Webservers ist jedoch Voraussetzung für die Nutzung, ein Wissen, das anderswo eingeholt werden muss.

bitus in spezifischer Weise besonders erforderlich. Linux ist damit ein Beispiel dafür, dass Technologien nicht per se eine egalisierende Wirkung ausüben.

4. In Software eingeschriebene Skripte werden den Nutzer_innen bewusst, wenn sie einen Sinn für die spezielle Einschreibung haben: So erkennen viele Nutzer_innen den Einfluss des „Look & Feel“ von Apple-Produkten auf Ubuntu. Fortgeschrittene sehen, wie sie durch kreatives Umfunktionieren von Software von intendierten Nutzungspraktiken abweichen können. Demgegenüber ist das Problembewusstsein für Genderskripte vor allem unter der sich männlich definierenden Mehrheit nicht sehr ausgeprägt.

Im Zusammenhang mit eingeschriebenen ästhetischen Vorstellungen in Artefakte kam eine weitere Dimension in die Betrachtung von Technik hinzu, die in Abschnitt 6.2.2 bereits theoretisch diskutiert wurde: In die Software werden außer nüchternen Funktionsregeln auch andere Bedeutungen eingeschrieben. Diese Ebene wurde von den bisher angeführten Techniksoziolog_innen nicht untersucht, sondern von der feministischen Forschung in den Blick gerückt. Auf dieser Ebene können technische Artefakte sehr wohl mit Bourdieus Begriff von objektivierter Geschichte untersucht werden. Das Konzept der Skripte (Akrich 1992) bzw. der Genderskripte (Rommes 2002) habe ich herangezogen, um solche Einschreibungen zu zeigen und theoretisch zu reflektieren. Die Autor_innen sehen technische Artefakte einerseits als eine Fläche, auf der Botschaften verewigt werden. In meinem Material waren dies drei Beispiele: Erstens nehmen die Entwickler_innen von *Pendrive*-Linux heterosexuelle, männliche Computernutzer als ihre Zielgruppe an, was sich in dem Hintergrundbild äußert. Zweitens spielen die Entwickler_innen der freien Routingsoftware für Access Points auf eine gesellschaftliche Arbeitsteilung an, in der Frauen Reproduktionsarbeit leisten, während Männer technische Geräte entwickeln und konfigurieren. Drittens haben die Entwickler_innen von *Ubuntu*-Linux die ästhetische Gestaltung des Standarddesktops an Modevorstellungen ausgerichtet, die von einem proprietären Konkurrenten gesetzt werden. Dabei haben sie die vorherigen ästhetischen Anspielungen auf Südafrika fast komplett aus der Software herausgenommen. Diese Skripte stehen in klarer Beziehung zu gesellschaftlichen Geschlechterverhältnissen und -stereotypen sowie zu westlichen Modevorstellungen und Marktmacht durch Stilsetzung — Verhältnisse, die in den Artefakten als objektivierter Geschichte reproduziert werden.

Andererseits sehen die Autor_innen die Entschlüsselung von Skripten durch Nutzer_innen als Teil ihres Konzeptes an: Die Nutzer_innen sind ihnen nicht ausgeliefert, ih-

re Wahrnehmung der Artefakte wird nicht unbedingt von Skripten bestimmt. Da gerade in FLOSS-Communities gerade die fortgeschritteneren Akteur_innen die Artefakte eben nicht als eigenständig handelnde Einheiten betrachten, haben sie auch ein besonderes Auge für die Machart der Software. Sie erkennen Skripte und positionieren sich zu ihnen. Vor dem Hintergrund, dass ich nicht sehr viele Materialbeispiele zur Entschlüsselung von Skripten habe, fiel mir vor allem die deutliche Reaktion der Person auf, die sich von einem Genderskript klar distanzierte, von dem sie sich betroffen sah, nämlich von dem Hintergrundbild. Für den Gesprächspartner wurde das Skript erst durch ihre Entschlüsselung zum Thema. Dies und das Genderskript aus dem Access Point-Beispiel, das mir als besonders kreative Idee erklärt wurde, führen mich zu der Interpretation, dass für Genderskripte keine hohe Sensibilität in FLOSS-Communities verbreitet ist.

7. Vertrauen und Sinn für Datenschutz

In der vorliegenden Arbeit wird jede Praxis von FLOSS-Communities daraufhin befragt, welche Differenzierungen und Hierarchisierungen sie (re-)produziert. In diesem Kapitel richte ich den Blick jedoch auf Praktiken, die auf einer *sehr grundsätzlichen* Ebene ordnenden, formalisierenden und/oder hierarchisierenden Charakter haben. Diese Praktiken werden anhand von Beobachtungen in zwei Bereichen aufgerollt: Erstens geht es um technische Zugänge zu Computern (7.1), die bei der Kleingruppenarbeit in LUGs und in der von mir beobachteten IT-Abteilung wiederkehrende inhaltliche Themen darstellen. Über Zugriffsrechte und Passwörter werden Ordnungen und Hierarchien in Softwarekonfigurationen abgebildet. Sie entscheiden, wer was machen darf und wer dies kontrolliert. Diese Praktiken spielen sich im Bereich Systemadministration ab. Zweitens untersuche ich Key signing-Parties als Ritual auf Community-Events (7.2), bei denen es um verschiedene Formen von Vertrauen innerhalb der FLOSS-Communities geht.

7.1. Technische Sicherheit in der Praxis

In einer LUG übergeben Hilfesuchende ihre Computer vertrauensvoll den Helfer_innen. Alle gehen dort davon aus, dass diese ihren Zugriff auf fremde Rechner nicht missbrauchen. Die Beobachtungen beim Hacking-Contest haben dagegen gezeigt, dass viele Anwesende ein Verständnis davon haben, wie man in Computer einbrechen kann. In diesem Abschnitt geht es aus einer anderen Perspektive um das Thema technische Sicherheit, nämlich um den praktischen Umgang mit Passwörtern und Zugriffsrechten. Beides war in der teilnehmenden Beobachtung sehr präsent; oft kam es zu Verwirrungen mit Logins oder den damit verbundenen Zugriffsrechten. Darüber hinaus unterschieden sich die Praktiken in der Linux User Group stark von denen im professionellen Umfeld der beobachteten IT-Abteilung. Meine Beobachtungen zu Zugriffsrechten und Passwörtern können als Beispiele für Praktiken im Bereich der technischen Sicherheit gelesen werden.

Sie sind einerseits durch das Betriebssystem stark vorstrukturiert, andererseits werden sie sehr unterschiedlich gehandhabt.

Vor dem Einstieg ins Material ist eine kurze Einführung nötig, worum es bei Zugriffsrechten und Userverwaltung unter Linux geht. Das Betriebssystem ist schon immer auf einen Betrieb ausgelegt, in dem sich mehrere Nutzer_innen, auch gleichzeitig, am selben Rechner anmelden können. Diese Möglichkeit zieht es nach sich, dass für alle Dateien auf dem Rechner festgelegt wird, welche Nutzer_innen sie lesen, ändern oder (im Falle von Programmen) ausführen dürfen, sonst könnten die Beteiligten die Daten anderer lesen oder löschen oder das Funktionieren des Betriebssystems selbst beeinträchtigen. Jede Datei ist aus diesem Grund mit Zugriffsrechten versehen, die besagen, welcher Person und welcher Gruppe sie gehört, was die Person und die Gruppe damit machen können, und was andere Nutzer_innen dürfen, die weder Besitzer_innen der Datei, noch Mitglieder der besitzenden Gruppe sind. Auf jedem Rechner gibt es einen *root*-Account, der die Rechte zur Administration des Computers innehat. Das bedeutet, als *root* verfügt man über sämtliche Zugriffsrechte und kann auch den so genannten „unprivilegierten“ Nutzer_innen Rechte zuteilen. Diese Unterscheidung ist einer der Gründe, aus denen Linux und Unix im Vergleich zu Windows als sicherere Betriebssysteme gelten.¹ Neben dem *root*-Account können mit dem Befehl *sudo* auch andere angemeldete Nutzer_innen unter Umständen bestimmte Befehle mit *root*-Rechten ausführen (s.u.). Dies hat vor allem durch die Distribution *Ubuntu* Verbreitung gefunden.

7.1.1. Umgangsweisen mit Passwörtern

Insgesamt ist es in FLOSS-Communities eine Konvention, die Passwörter anderer nicht wissen zu wollen. Sie gelten als zu respektierende Geheimnisse der jeweiligen Accountinhaber_innen. Um diese Privatsphäre zu wahren, sind die Programme so gemacht, dass Passwörter nur in Form von Sternchen angezeigt werden, in der → *Shell* ist in der Regel überhaupt nichts zu sehen, so dass auch die Anzahl der Zeichen des Passwortes nicht sichtbar wird. In den Hilfesituationen in der LUG sind Passwordeingaben sehr häufig nötig, wenn die Besitzer_innen sich einloggen oder für Administrationstätigkeiten in den *root*-Account wechseln. In diesen Situationen, wo also ein_e Helfer_in daneben sitzt, wird häufig deutlich, wie stark die Passwörter als Geheimnisse respektiert werden.

¹Unter Linux wurden von vorn herein Dateisysteme verwendet, die Dateizugriffsrechte unterstützten, was bei den unter Windows gängigen FAT-Dateisystemen nicht der Fall war.

Community-Mitglieder erwehren sich des Eindrucks, dass sie die Eingabe eines Passwortes auf der Tastatur mitverfolgen: Diejenigen, die potenziell ein fremdes Passwort lesen könnten, verhindern dies selbst, indem sie sich diskret wegdrehen, wenn jemand anders gerade ein Passwort eingibt:

Während er spricht, erscheint in seiner Shell die Passwort-Abfrage seines gpg-Schlüssels. Ich gehe kurz zu meiner Kaffeetasse hinüber, damit er sich bei der Eingabe nicht beobachtet fühlt. (Kommentar: Ich habe schon beobachtet, dass Menschen, die neben einer Person sitzen, die gerade ein Passwort eingibt, sich demonstrativ abwenden, um zu zeigen, dass sie das Passwort nicht mitlesen.) (LUG 2009l)

A52 hat ein kleines Befehlsfenster geöffnet, doch als sie ihr root-Passwort dort eingeben möchte, ist der Fokus [auf dem falschen Fenster].² A3 sieht, dass dort die ersten Zeichen ihres Passwortes erscheinen, hält seine Hand davor und macht sie darauf aufmerksam (LUG 2009j).

Während viele derart genau darauf achten, keine fremden Passwörter zu erfahren, gehen in der LUG andere mit ihren eigenen Passwörtern sehr großzügig um. Der junge Mann in der folgenden Szene benutzt ein extrem altes Notebook und gab an, Spaß daran zu haben, ihn noch in Betrieb zu halten. Auf dem Gehäuse waren an verschiedenen Stellen Einträge mit einem Stift vermerkt.

Ich frage A53, ob ich seinen Rechner fotografieren dürfe. Er zögert und sagt, sein Passwort stünde darauf. Ich biete ihm an, es zuzuhalten, daraufhin stellt er seine Mate-Flasche auf die Stelle. (Ich denke: Ich glaub es nicht, wer schreibt sich denn sein Passwort auf das Gehäuse und will dann verhindern, dass es jemand liest?) (LUG 2009g).

Meine Überraschung bezog sich hier auf die Widersprüchlichkeit von A53 im Umgang mit seinem Passwort: Es war deutlich sichtbar, sollte aber nicht fotografiert werden. In der User Group waren solche Widersprüche häufiger zu beobachten. A7 stellt seinem Helfer A8 in einer Situation sehr viele Fragen zur Sicherheit seines Rechners. Es ging zuvor um verschiedene Programme, die mit der Absicherung des Computers zu tun hatten und sich auf Angriffe über das Netzwerk bezogen. A7 macht sich Sorgen um die Sicherheit seines Rechners.

²A52 möchte das Passwort eigentlich in ein anderes Fenster eingeben, das in dem Moment nicht das Fenster im Vordergrund ist, das auf die Tastatureingabe reagiert. Sie tippt es versehentlich in ein anderes Fenster bzw. Programm ein, wo jedoch der eingegebene Text für den Betrachter sichtbar ist.

A7 sagt, wer sich mit Sicherheit beschäftige, rutsche da immer tiefer rein. Man glaube dann, man sei verletzlich. (...) Etwas später: A7: „Man muss sich damit abfinden, dass Sicherheit ein Verzicht auf Bequemlichkeit ist.“ (Kommentar: Spätestens ab hier merke ich, dass A8 sich über die Paranoia von A7 amüsiert, sie übertrieben findet.) A8: „Och, ich finde das hier bequem und fühle mich auch sicher“ (LUG 2007c).

Ein Passwort-Manager fragt nach einem Passwort. A7 öffnet eine Datei auf dem → *Desktop*, dort steht das Passwort im Klartext und für uns sichtbar, er kopiert das Passwort heraus und fügt es in dem Passwort-Manager ein. A8 sagt nichts. Ich frage, ob ich das richtig gesehen habe, dass das Passwort in einer Datei auf dem Desktop liegt. Ja. Er hätte das heute erst eingerichtet. (Kommentar: Es klingt so, als ob das so nicht bleiben soll. A8 amüsiert sich auch. Ich wundere mich ein bisschen, dass er nicht vor mir etwas gesagt hat.) (LUG 2007c)

Ich lege im Protokoll die Interpretation nieder, dass A7 paranoid ist, sich also zu große Sorgen um seine Sicherheit macht, und dass sein Helfer A8 die Sorge nicht teilt. Offenbar habe ich erwartet, dass er als Helfer das Speichern eines Passwortes in einer Datei auf der Arbeitsoberfläche anspricht und habe mich eingemischt, als er es nicht tat. Dies sagt über meinen eigenen Habitus etwas aus, nämlich dass er es verbietet, Passwörter in unverschlüsselte Dateien oder auf das Gehäuse des Computers zu schreiben. Gleichzeitig ist die Widersprüchlichkeit auffällig: A53 weiß, dass sein Passwort öffentlich lesbar ist, A7 und A8 beschäftigen sich über einen längeren Zeitraum mit technischer Sicherheit, doch in beiden Fällen unterläuft ihre Praxis ihr theoretisches Wissen oder Interesse.

Schließlich ist mir in der LUG aufgefallen, dass die Anwesenden ihre Rechner unbeaufsichtigt an ihrem Sitzplatz stehen lassen oder ihre User- und root-Passwörter den Helfer_innen laut sagen. Während mir A44 seinen Rechner *mit* allen Daten anvertraut, hat A61 (wie weiter oben bereits erwähnt) ein Notebook allein zum Üben und Experimentieren mit Linux. Das root-Passwort dieses Rechners hat er diversen Personen in der LUG gesagt.

Ich will mir kurz bei A44, als er rauchen geht, etwas aus dem Internet herunterladen. A44 schiebt mir seinen Rechner rüber und bemerkt, dass eine root-Shell offen sei, mit der ich jetzt mache könne, was ich wolle (LUG 2009h).

A6 steht auf und sagt, er gehe kurz raus, wenn ich wollte, könnte ich mit seinem Rechner spielen. (...) Er geht und ich sitze mit dem Rechner, dem T61, da. Ich gucke mich

darauf um und stelle fest, dass überhaupt keine Daten darauf sind, nur ein gespeichertes Passwort für einen DSL-Zugang ist in der Passwörter-Verwaltung. Sonst ist kein E-Mail-Programm eingerichtet, keine Daten außer dem Hintergrundbild (einem dunklen Bild mit der Aufschrift Ubuntu 9.04) im persönlichen Ordner. Als er später zurückkommt, spreche ich ihn darauf an, er sagt, die Daten habe er auf einem USB-Stick. Er erinnert mich daran, dass ich auch sein root-passwort wisse. Ich erinnere mich tatsächlich (LUG 2009e).

Es scheint selbstverständlich zu sein, dass niemand der Umsitzenden sich der Rechner bemächtigen würde. Hier gibt es also — wie insgesamt im LUG-Zusammenhang — ein Vertrauen darauf, dass die Anwesenden ihr Wissen über einen der Rechner nicht missbrauchen, um sich Zugang zu verschaffen.

Vor dem Hintergrund meiner Beobachtungen in der IT-Abteilung sind diese Praktiken bemerkenswert: In der IT-Abteilung durften Passwörter ausschließlich in einem dafür vorgesehenen Ordner notiert werden, der im Safe der Firma eingeschlossen war. In der alltäglichen Praxis wurden sie selten ausgesprochen, eigentlich nur beim Einarbeiten neuer Mitarbeiter_innen. Sonst wurden nur Bestandteile laut genannt, mittels derer die Kolleg_innen sich zur Erinnerung auf die Sprünge halfen. In diesem Umfeld hat sich mein Habitus im Umgang mit Passwörtern und diesbezüglichen Sicherheitsbedenken geformt, der sich in den Beobachtungsprotokollen aus der LUG widerspiegelt. Die Handhabung von Passwörtern im professionellen IT-Umfeld steht in einem Arbeitskontext von großer Verantwortung: Die Firma ist für den Schutz aller Daten verantwortlich, die sie erhebt, speichert und verarbeitet. Ein Blick in die IT-Grundschutzkataloge des Bundesamtes für Sicherheit in der Informationstechnik (BSI)³ zeigt, welche umfangreichen Maßnahmen dies erfordert. Dort werden der IT-Administration detaillierte Tipps zur Handhabung von Passwörtern, zum Zurücksetzen und zum Hinterlegen von Passwörtern nahegelegt.⁴ Das ausdrückliche Herumerzählen eines root-Passwortes, wie A6 es in der Linux User Group betreibt, zeigt demgegenüber, dass im LUG-Kontext und auf einem datenlosen Rechner nichts auf dem Spiel steht.

Der Schutz von Accounts mit Passwörtern ist eine der grundlegenden Ordnungen des Betriebssystems. Die Praktiken im Umgang mit Passwörtern sind breit gefächert; sie ziehen

³Das BSI ist eine Bundesbehörde, die beim Bundesministerium des Innern angesiedelt ist. Die Grundschutzkataloge sind rechtlich nicht bindend. Sie empfehlen jedoch Maßnahmen für einen „normalen“ Schutzbedarf in der Informationstechnik und gelten als ‚State of the Art‘.

⁴Vgl. <https://www.bsi.bund.de/ContentBSI/grundschutz/kataloge/m/m02/m02011.html>, Abruf: 06.07.2011.

sich von strengem Schutz von Passwörtern in der Firma über großen Respekt vor der Privatsphäre anderer bis hin zum Bekanntgeben eigener root-Passwörter in der Linux User Group. Personen, die ihre Passwörter nicht sicher aufbewahren, wie derjenige, der es auf den Computer schrieb, sind sich theoretisch dessen bewusst, dass dies nicht „sicher“ ist. Selbst in dem Gespräch zwischen A7 und A8, das explizit den Schutz des Rechners vor unbefugten Eingriffen zum Thema hat, gehen die Beteiligten jedoch über die Speicherpraxis bezüglich des Passwortes hinweg, die es versierten Angreifer_innen sehr leicht machen würde, an das Passwort zu kommen. Auch hier gehen die beobachteten Praktiken weit auseinander. Theoretisches Wissen oder Interesse ziehen nicht in jedem Fall eine konsequente Anwendung nach sich. Die Beobachteten verfügen in vielen Fällen über ein theoretisches Wissen, dem sie in ihren konkreten Praktiken nicht entsprechen.

7.1.2. „Ich bin root, ich darf das!“ — Zugriffsrechte

Klar geregelte Zugriffsrechte sind — wie beschrieben — fester Bestandteil von unixartigen Betriebssystemen. Bei meinen Beobachtungen gab es immer wieder Szenen, in denen Zugriffsrechte Gegenstand von Lernprozessen waren, in denen sie Fehlerquellen oder Teil von Problemlösungen waren. Zugriffsrechte unterliegen jedoch auch der Diskussion und persönlichen Aushandlung: Eine Situation aus der LPI-Lerngruppe zeigt, dass es neben den technischen Zugriffsrechten auch Absprachen über Zuständigkeiten gibt, die von den technischen Möglichkeiten abweichen:

A39 fragt, ob es Probleme geben könnte, wenn sie jetzt im Netz der LUG einen zweiten Mailserver einrichten und starten. Sie sprechen darüber, ob sie mal nachsehen, wie der bereits existierende Server konfiguriert ist, aber A2 lehnt das ab: „Nein, das ist A1s Bereich.“ Sie hätten Absprachen, wer für welche Bereiche zuständig sei. Er würde da auf keinen Fall rangehen (LUG 2009d).

A2 stellt hier das sozial Ausgehandelte über seine technischen Zugriffsrechte: A1 betreut den Mailserver der LUG und A2 möchte ihn — obwohl er könnte — nicht antasten, auch nicht, um sich nur die Konfiguration anzusehen.

Um Absprachen über Zugriffsrechte ging es auch in der beobachteten IT-Abteilung: Während meiner Zeit dort ging es erstens darum zu definieren, welche Mitarbeiter_innen welche Zugriffsrechte auf welche Teile der IT-Infrastruktur haben sollten. Zweitens sollte dieses Konzept von Rollen und Rechten in einer zentralen Software verwaltet werden.

(Ein Beispiel: Buchhalter_innen sollten Buchungen elektronisch durchführen, aber keine Kundendaten löschen und nicht die Mailingliste der Geschäftsführung lesen dürfen. Wenn ein_e neue_r Buchhalter_in die Arbeit antritt, sollte eine zentrale Software mit dieser Stelleninformation automatisiert die entsprechenden Rechte vergeben.) Die erste zu bewältigende Aufgabe ist keine technische, sondern eine organisatorische und soziale: Letztlich werden Arbeitsteilung und Unternehmenshierarchien mit IT-Instrumenten abgebildet und fortgeschrieben. Die Entscheidung über das „Rechte und Rollen“-Konzept lag bei der Geschäftsführung. Die Notwendigkeit zur Umsetzung eines solchen Konzeptes wurde von der IT-Abteilung an sie herangetragen, denn auch hierfür gibt es Richtlinien zum IT-Grundschutz vom BSI. Sie orientieren sich am Grundsatz, jede_r solle so viele Zugriffsrechte wie nötig und so wenige wie möglich erhalten.⁵ Die IT-Abteilung trug die nötigen Informationen, auf deren Basis die Geschäftsführung über die Rollen und Rechte hätte entscheiden können, zusammen, doch die kam jedoch zu keiner Entscheidung. Die Umsetzung hat demnach nicht stattgefunden, so dass die Zugriffsrechte im Vergleich zu den BSI-Vorgaben großzügig ausfielen.⁶

Die Rechte und Mitarbeiterrollen wurden in einer Software zentral verwaltet, einem so genannten Verzeichnisdienst, in diesem Fall die freie Software *OpenLDAP*. Ein Verzeichnisdienst stellt für ein Computernetzwerk zentral Informationen dar: Dies können eine Benutzerverwaltung oder zentrale Adressbücher sein, aber auch Konfigurationen für Software (vgl. Feilner/Kames 2009). Die Accounts der Mitarbeiter_innen wurden dort angelegt, ihre groben Zugriffsrechte auf gemeinsam genutzte Server oder Software konfiguriert, Telefonnummern zugewiesen bzw. Telefone konfiguriert uvm.⁷ *OpenLDAP* organisiert die Daten in einer hierarchischen Baumstruktur, in der z. B. Land, Firma, Organisationseinheiten, Mitarbeiter_innen, Telefone oder Computer mit diversen Daten gespeichert wer-

⁵vgl. <https://www.bsi.bund.de/ContentBSI/grundschutz/kataloge/m/m02/m02007.html> und <https://www.bsi.bund.de/ContentBSI/grundschutz/kataloge/m/m02/m02008.html>, Abruf: 06.07.2011

⁶Jörg Strübing hat das Verhältnis zu Softwareentwickler_innen zu anderen Berufsgruppen innerhalb von Unternehmen untersucht. Er zeigt, wie die unterschiedlichen Aufgaben und Relevanzen (vor allem an den Beispielen Marketing und Management) zu Konflikten mit Softwareentwickler_innen führen, und führt die Schwierigkeit der Vermittlung zwischen verschiedenen Arbeitslogiken und Inhalten zur Erklärung ins Feld (vgl. Strübing 1992: 103 ff., 151 f.).

⁷Ein Beispiel kann die Vereinfachung durch Zentralisierung zeigen: Ohne *OpenLDAP* hätten für jede_n Mitarbeiter_in Accounts an verschiedenen Stellen angelegt werden müssen, z. B. einer für Zugang zur ERP-Software, einer für die Groupware, einer für das Intranet, die Telefonnummer müssten von Hand in der Telefonanlage eintragen werden, etc. Je mehr Software an das *OpenLDAP* angebunden ist, desto schneller lassen sich Personen hinzufügen, Accounts verändern oder löschen.

den können (vgl. Schwaberow 2001).⁸ Durch die effiziente Zentralisierung von Konfigurationen in *OpenLDAP* werden somit soziale und organisatorische Hierarchien und Arbeitsteilungen in einer Software reproduziert, die auch in sich hierarchisch angelegt ist. In dieser Hierarchie stehen die organisatorischen Einheiten „Personen“ und „Server“ interessanterweise auf gleicher Stufe, der Organisation oder Firma untergeordnet. Das Denken, das hinter einer solchen Abbildung von gesellschaftlichen Strukturen in Software steht, wird in der theoretischen Reflexion dieses Kapitels als formalisierendes Denken begriffen, das auf der Herstellung von Unterschieden und ihrer Hierarchisierung beruht.

Zugriffsrechte wurden hier als eine grundlegende Voraussetzung untersucht, die in Linux eingeschrieben ist und die die Nutzungspraxis stark prägt. Dem Thema wird — ähnlich wie Passwörtern — in der professionellen Systemadministration eine sehr große Bedeutung beigemessen. Auf dem Spiel stehen die Daten von Kund_innen und Mitarbeiter_innen, die verarbeitet werden und die zu schützen sind. In den administratorischen Tätigkeiten, aber auch in den Empfehlungen des BSI kommt ein grundsätzliches Misstrauen gegenüber Computernutzer_innen zum Ausdruck, dessen Grund reale Datendiebstähle sind. Dieses Misstrauen muss, wie am Beispiel der Firma gezeigt, nicht unbedingt auf Seiten derjenigen bestehen, die die Software nutzen. Angesichts der hohen Anforderungen an technische Maßnahmen in der Firma wurde deutlich, wie schwer Themen wie Zugriffsrechte und Datenschutz umzusetzen sind, da sie den Verantwortlichen und allen anderen Mitarbeiter_innen der Firma erst vermittelt werden müssen. Die Software enthält die technischen Einstellungsmöglichkeiten, deren Einsatz auszuhandeln ist. Er steht in einem Spannungsfeld von Datenschutzrecht, Verantwortung und Arbeitsalltag. In diesem Fall hielt die Software Möglichkeiten vor, die in der Genauigkeit von den Entscheidungsträger_innen nie durchdacht wurden. Dadurch kommt es auch im professionellen Umfeld zu einer widersprüchlichen Handhabung von Passwörtern und Zugriffsrechten, nicht nur im privaten Umfeld der LUG.

7.2. Die Formalisierung von Vertrauen beim Keysigning

Nachdem im letzten Abschnitt Praktiken im Umgang mit technischen Vorgaben untersucht wurden, beleuchte ich hier ein soziales Ritual, das ein fester Bestandteil aller größeren Linuxveranstaltungen ist.

⁸Siehe zur Veranschaulichung auch die Abbildung der Baumstruktur unter <http://www.openldap.org/doc/admin24/intro.html>, Abruf: 17.11.2013.

„Zu jedem Linux-Event gehört eine Keysigning-Party nach klassischem Vorbild.“⁹

Keysigning-Parties stehen im Zusammenhang mit digitaler Verschlüsselung und Authentifikation. Da bei der Online-Kommunikation zunächst keine Sicherheit darüber besteht, ob „online- und offline-Identität“ der Kommunikationspartner_innen übereinstimmen (Brinkmann/Seifert 2001: 31), wurde Keysigning erfunden. Es soll sicherstellen, dass die digitalen Schlüssel, mit denen Personen sich im Internet authentifizieren, auch wirklich diesen realen Personen gehören. Die Kommunikationspartner_innen generieren sich mit der freien Software *GnuPG* Schlüsselpaare, einen *public key* (Abb. 7.1), den sie an andere zum Verschlüsseln von Daten verteilen, und einen *private key*, den sie zum Entschlüsseln behalten. Dieses Prinzip heißt *Public Key Infrastructure (PKI)*. Es ist mit diesen Schlüsseln nicht nur möglich, verschlüsselt zu kommunizieren, sondern auch Dateien digital eindeutig zu unterschreiben (auch ohne Keysigning). Da jede_r sich selbst Schlüssel auf beliebige Namen generieren und diese im Netz verteilen kann, wurde ein Verfahren geschaffen, das die Verbindung zu Personen überprüft.

„Using PKIs for authentication allows a message that is not encrypted to be authenticated to a degree of mathematical certainty that is impossible for humans to calculate, and allows the recipient to be certain that the message is both unchanged, and was signed only by the person in possession of the key with which it was signed“ (Kelty 2005c: 136 f.).

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.10 (GNU/Linux)

mQINBEwIzagBEADa2YWMQJ5BkqU4hoirQJx75wXtVIm/M91THDAqMdBgzsvG2JPO
QGBE7jaroUna2Wyg4HSVQYFq9f1WNokLPRqoLdB1+WLn0+dUNLQkq3DxzAgQ0ksV
t5Jqr+L6PD7AnyQdo0oLH5yw6jLZHjg57r0pVYM19JZ3Ij+34Bi7Y1VbbsC76zLN
f05FqDjdK88DcPukPZ6WycbKvvh2sgLYv6yIrYH2xf49YfNU9S0scdAFeb9ZJ+g1
JL70z+henBaZN80e9bDnPg3s388f5n0Zo70gRfcTFhOK92a8RZ8UD3uj8iRk6luX
gZaDTtZ0n2pv7Nzhs4wUj89MAp0Th0dIITkL4J3kDYcKgI62DZgR+v13sAU05xLI
GvLZRu2rV75H7jGyXhC8zCw7ruc2EJf0wjho6n0Hjg0kJoUA0X1qmQao873bu4Y9
W9AU=dDeT
-----END PGP PUBLIC KEY BLOCK-----
```

Abb. 7.1.: GnuPG-Schlüssel

Die Echtheit eines Schlüssels wird durch digitale Signaturen anderer Personen nach einem persönlichen Treffen bescheinigt. Dieses Treffen ist die Keysigning-Party. Dahinter steht die Idee, dass nicht alle Menschen, die verschlüsselt miteinander kommunizieren möchten, sich persönlich treffen können, um sich die Echtheit der Keys bestätigen zu lassen. Andere, die z. B. in der Nähe leben, können die Identität der Person jedoch überprü-

⁹ Ankündigung auf der Website des Brandenburger Linux-Infotages: http://www.blit.org/2008/zeitplan/zemisch_keysigning.shtml, Abruf: 27.05.2010

fen und dem Schlüssel eine Signatur beifügen. So kann ein Netz entstehen, nach dem Prinzip: A und B überprüfen gegenseitig ihre Schlüssel, B und C auch. Dann müssten sich A und C gegenseitig darauf verlassen können, dass ihre Schlüssel „echt“ sind. Dieses Netz aus digitalen Signaturen bildet das so genannte *Web of Trust*. Im Sprachgebrauch der Community „vertraut“ man sich nach dem gegenseitigen Signieren der Keys. Je mehr Menschen bzw. Schlüssel daran teilnehmen, desto höher ist die Wahrscheinlichkeit, dass ich jemandem vertraue, der den Schlüssel der Person signiert hat, mit der ich kommunizieren möchte.

In Softwareprojekten wird auf diese *GnuPG*-Keys zurückgegriffen, um Daten den einzelnen Softwareentwickler_innen eindeutig zuordnen zu können, z. B. bei *Debian*- und *Ubuntu*-Linux. Matthieu O’Neil setzt sich unter anderem damit auseinander, wie Projektleiter von Onlineprojekten die administrative Macht über das *Debian*-Projekt auf viele Menschen verteilen und es dabei schaffen, die Qualität der gemeinsamen Produkte zu erhalten. Diejenigen, die Software auf die Server einer Distribution hochladen können, könnten damit potenziell großen Schaden anrichten.¹⁰ Das *Debian*-Projekt sichert sich dagegen ab, indem es keine anonym eingereichten Beiträge zulässt: Nur autorisierte Person können Software hochladen und die Authentifizierung erfolgt über digitale Signaturen (vgl. O’Neil 2009: 136; zu *Ubuntu* siehe Lloyd 2007: 92 ff.). Bei *Debian* wird *public key cryptography* seit 1994 zur Authentifizierung von Mitgliedern verwendet. Seit 2000 sind signierte Schlüssel eine Bedingung dafür, Projektmitglied bei *Debian* zu werden (vgl. O’Mahony/Ferraro 2007: 1095). Neben anderen Zugriffsbeschränkungen wird also der Code selbst mit den Keys der Entwickler_innen unterschrieben, so dass er eindeutig zurechenbar ist.

„Online communities are routinely described as having fluid boundaries and shifting members and identities. In contrast, members of FLOSS projects who are developing software that is hosted on protected servers connected to the Internet must maintain a distinct and trusted identity, which will enable them to gain access to these protected resources“ (O’Neil 2009: 136).

Bei Keysigning-Parties versammeln sich viele Personen an einem Ort, die sich bestätigen, dass die Schlüssel tatsächlich ihre sind, und die sich mit amtlichen Dokumenten als sie

¹⁰Zum Vergleich: In den Sozialwissenschaften käme dem Schutz eines gemeinsam verfassten Artikels, der auf einem vom Internet aus erreichbaren Server abgelegt wird, eine ähnliche Bedeutung zu. Den inhaltlich Verantwortlichen ist wichtig, dass nicht beliebige Personen Zugriff auf den unfertigen Artikel bekommen oder darin herumschreiben. Bei der Entwicklung freier Software ist zudem die Wahrscheinlichkeit sehr groß, dass sich die an einem Projekt Beteiligten noch nie zuvor persönlich getroffen haben.

selbst ausweisen. Die Teilnehmenden melden sich vor einer Keysigning-Party mit ihrem digitalen Schlüssel an, indem sie ihn an den die Organisator_in der Party schicken. Er_sie erstellt eine Liste der Teilnehmenden, die alle sich vor der Party herunterladen und ausdrucken. Dort sind die Schlüssel in verkürzter Form aufgelistet: Die Key-ID, das Erstellungsdatum, der → *Fingerprint* statt einer Darstellung des gesamten Keys, die Namen und E-Mail-Adressen, für die die Schlüssel verwendet werden können. Darüber hinaus ist Platz um zu markieren, ob die Richtigkeit des Fingerprints vom Schlüsselinhaber bestätigt wurde und ob man die Ausweispapiere der Person gesehen hat (Abb. 7.2). Vor der Keysigning-Party sind die Teilnehmer_innen zudem dazu aufgerufen, eine oder mehrere Prüfsummen dieser Datei, der Teilnehmer_innenliste, zu berechnen und aufzuschreiben. Welche das sein sollen, ist in der Regel auf der Website des Community-Events angegeben:

„Berechne die MD5-, SHA1- oder SHA256-Hashsumme der Liste. Dies kannst du mit dem Programm *md5sum*, *sha1sum* bzw. *sha256sum* aus den *GNU Coreutils* tun. Weiterhin kannst du auch *GnuPG* (*gpg --print-mds \$DATEI*) oder ein anderes Programm dazu nutzen. Den errechneten Wert trägst du in das entsprechende Feld der Liste ein.“¹¹

```
-----
MD5 Checksum: _____ [ ]

SHA1 Checksum: _____ [ ]

-----
001 [✓] Fingerprint OK [✓] ID-Card OK [ ] [ ]
pub 1024D/AAE6022E 2004-12-07
Key fingerprint = 7A39 2F67 8CAE 262E 64FD 8A10 2F95 1508 AAE6 022E
uid [redacted]@n[redacted].darmstadt.de>
uid [redacted] <k[redacted]@l[redacted].com>
Signatures:[00550] Keys signed:[000645] MSD:[3.7760] Rank:[000047]
-----
002 [ ] Fingerprint OK [ ] ID-Card OK [ ] [ ]
pub 1024D/53A2E663 2004-12-03
Key fingerprint = 0439 B571 B738 4C72 4C6B 0562 F77D A547 53A2 E663
uid [redacted] <k[redacted]s@d[redacted].de>
Signatures:[00077] Keys signed:[000070] MSD:[4.3647] Rank:[001654]
-----
```

Abb. 7.2.: Teilnehmer_innenliste einer Keysigning-Party

Beim Zusammentreffen wird zuerst überprüft, ob alle Personen die gleiche Liste vor sich haben, indem jemand eine, zwei oder drei Prüfsummen der Datei mit den Teilnehmer_innen laut vorliest. Danach werden die Namen der Angemeldeten aufgerufen: Wer

¹¹<http://chemnitzer.linux-tage.de/2008/service/pgp.html>, Abruf: 27.05.2010

erschienen ist, bestätigt seine_ihre Anwesenheit und dass der Fingerprint auf der Liste mit dem des eigenen Schlüssels übereinstimmt.

Während bis dahin alle in einem Raum sitzen, stellen sie sich dann in zwei langen Reihen gegenüber auf. Die Teilnehmer_innen in der einen Reihe zeigen ihre Personalausweise oder Pässe vor, die in der anderen Reihe rücken nach und nach an ihnen vorbei, vergleichen ihre Gegenüber mit ihren Ausweisen und haken sie auf ihrer Liste ab. Dann haben sie die nötigen Informationen: Sie wissen, dass die Personen vor ihnen wirklich diejenigen sind, die sie im Internet zu sein vorgeben. Und sie haben von allen die Bestätigung, dass der Schlüssel ihnen zuzurechnen ist.

Das Signieren der Keys erfolgt später am eigenen Rechner, ist also nicht Teil der Keysigning-Party. Dabei laden die Beteiligten sich alle Schlüssel von einem Keyserver herunter und signieren sie mit dem Programm *GnuPG* mit ihrem eigenen Key. Das „Vertrauen“, das man einem Schlüssel aussprechen kann, wird von *GnuPG* beziffert: Auf einer Skala von eins bis vier ist abstufbar, wie stark das Vertrauen ist. Sie schicken den signierten Schlüssel dann per E-Mail an den Inhaber zurück. Dessen Key hat nun durch die Signaturen neue Informationen erhalten. Wenn er den aktualisierten Schlüssel wieder mit dem Keyserver abgleicht, sind die Signaturen der anderen dort öffentlich sichtbar (Abb. 7.3).

```
pub 4096R/950F2582 2010-06-04
uid Silke Meyer <silke05@...>
sig sig 53D94014 2010-06-04 _____ Silke Meyer
sig sig 55F2BCCB 2010-06-04 _____ Rolf
sig sig3 950F2582 2010-06-10 _____ [selfsig]
sig sig3 AB55B312 2010-06-11 _____
sig sig3 AAE6022E 2010-06-11 _____ Karlheinz
sig sig CB537B60 2010-06-18 _____ Martin
```

Abb. 7.3.: Ansicht eines signierten Schlüssels auf einem öffentlichen Keyserver

„Wenn man bei der Analyse eines Rituals wirklich den Ethnozentrismus des Beobachters vermeiden will (...), muß man (...) dieses praktische Verstehen verstehen. Es bewirkt, daß wir angesichts eines Ritus, dessen Grund uns uns schleierhaft ist, wenigstens überhaupt verstehen, daß ein solcher vorliegt und was ihn von der Interpretation trennt, die sich nur erlauben kann, wer eine Position außerhalb der Praxis bezieht“ (Bourdieu 1999: 38).

Keysigning-Parties werden im Anschluss an Bourdieu hier als Ritual begriffen, dessen innere Logik es zu verstehen gilt. Die kollektive Definition seiner spezifischen Funktionen für FLOSS-Communities soll nachvollzogen werden (vgl. Bourdieu 1999: 177 f.). Keysigning ist eine communityspezifische Praxisform, die von einem bestimmten Habitus

strukturiert wird, in einem Feld mit einer besonderen Kapitalverteilung stattfindet. Ziel dieses Abschnitts ist es, die (möglichen) objektiven Strukturen, die mentalen Dispositionen und die Herstellung von Differenz, die in diesem Ritual hervorscheinen, zu rekonstruieren. Dabei interessiert mich erstens, welche Unterschiede beim Keysigning sozial relevant gemacht werden. Zweitens suche ich nach den Wahrnehmungsmustern, die die Beteiligten inkorporiert haben müssen, um beim Keysigning diese Unterschiede machen zu können. Mit Bourdieu haben Keysigner_innen einen spezifischen Sinn für das Spiel, eine *illusio*, sie glauben an bestimmte Regeln und Strukturen, die herausgearbeitet werden sollen. Mit Thomas Scheffer ist Keysigning als Ritual dann „ethnografisch durchdrungen“, wenn das Kopfschütteln aufhört, wenn man zeigen kann, dass das Ritual innerhalb der Community sinnvoll und wichtig ist.¹² In den folgenden Abschnitten wird zunächst versucht, die innere Logik des Keysigning nachzuvollziehen. In einem zweiten Schritt trage ich meine Forschungsfragen an das Ritual heran.

7.2.1. Exkurs: Kämpfe um den Zugang zu Verschlüsselungstechnologien

Ein kurzer Exkurs in den historischen Wissensbestand von FLOSS-Communities zur Geschichte der Kryptografie seit den 1970er Jahren kann helfen, die Bedeutung von Keysigning-Parties zu verstehen.¹³ Dabei ist Verschlüsselung im Allgemeinen nicht mit Keysigning im Besonderen zu verwechseln. Der Einsatz, um den es bei Verschlüsselung im Allgemeinen geht, ist der Schutz elektronischer Daten vor unerwünschtem Zugriff, sei es durch staatliche Behörden, durch Telekommunikationsunternehmen oder Systemadministrator_innen. Keysigning-Parties drehen sich speziell um die Frage der Identität der realen Personen, mit denen im Internet kommuniziert wird.

Bereits bevor es das Internet gab, waren Computer und die auf ihnen hinterlegten Daten leicht überwachbare und überwachte Räume. Frühe Privacy-Aktivist_innen formulierten netzpolitische und datenschutzrechtliche Forderungen und schrieben (teilweise frei zugängliche) Software, um sich selbst die gewünschte informationelle Selbstbestimmung bei der elektronischen Kommunikation zu schaffen. Die Forschung und Entwicklung im Bereich Verschlüsselungstechnologien stand über Jahrzehnte unter der Geheimhaltung von Regierungen und Nachrichtendiensten und wurde erst von Aktivist_innen

¹²AK Politische Ethnografie am Institut für Europäische Ethnologie der HU Berlin, Mai 2010

¹³Für ausführlichere Darstellungen siehe Levy (2001), Bendrath (2001) und Singh (1999).

für die Öffentlichkeit zugänglich gemacht. Diese Programmierenden waren der Überzeugung, „dass die Privatsphäre der Menschen ein extrem hohes Gut ist, das nicht von der Gnade von Regierungen oder Konzernen abhängen darf, sondern durch die Menschen selbst geschützt werden muss“ (Bendrath 2001).

Im Kalten Krieg betrieben fast nur Regierungen, Militärs und Geheimdienste Forschung zu Verschlüsselung. In den USA hatte der Nachrichtendienst *National Security Agency* (NSA) ein Quasi-Monopol auf kryptografische Technologien: Wer zum Thema Kryptografie arbeitete, war bei der US-Regierung beschäftigt oder bei einer von der Regierung kontrollierten Organisation. Ebenso galten alle anderen, die sich starker Verschlüsselungstechnologien bedienten, als verdächtig. Starke Verschlüsselungstechnologien unterlagen in den USA — analog zu Waffen — einem Exportverbot (vgl. Levy 1993; vgl. auch Bendrath 2001). Ab den 1970er Jahren, mit dem Ausbau des ARPAnets, gab es immer größeren Bedarf an einer zivilen Nutzung kryptografischer Technologien. Das Interesse wuchs damit unter akademischen IT-Forscher_innen. Volker Grassmuck sieht in allen Ländern die gleiche Konfliktlinie: Grundrechte, die eine unangetastete Kommunikation und den Schutz der Privatsphäre versprechen, kollidieren mit „Sicherheitsinteressen“ von Staaten (vgl. Grassmuck 2004: 368).

1975 entwickelten drei Informatiker von verschiedenen US-amerikanischen Universitäten, Whitfield Diffie, Martin Hellman und Ralph Merkle, die Idee der public-key-Kryptografie. Bei Recherchen stießen sie darauf, dass fast alle Informationen zu dem Thema als geheim eingestuft waren und nur der NSA und einem begrenzten Kreis von Forscher_innen zugänglich waren. Das zentrale, bis dahin ungelöste Problem der verschlüsselten Online-Kommunikation war die Übermittlung der Schlüssel zum Öffnen von Nachrichten über das Netz, in dem sie leicht abgefangen werden können. „The difficulty with this scheme was getting the key from one party to another — if you sent it over an insecure channel, what’s to stop someone from intercepting it and using it to decode all subsequent messages?“ (Levy 1993). Die grundlegend neue Idee an der public-key-Kryptografie war, dass sie auf *zwei* Schlüsseln beruht, von denen einer im Netz verteilt werden kann, ohne Daten zu kompromittieren. Eine solche dezentrale Lösung erlaubt es jeder Person, den Schlüssel zu den eigenen verschlüsselten Daten zu haben. Darüber hinaus können die *public keys* an zentralen Speicherorten aufgehoben werden, um E-Mailverschlüsselung in größerem Maßstab zu ermöglichen.¹⁴ 1977 setzen drei Wissenschaftler am MIT die

¹⁴1997 reklamierte die britische Regierung die Erfindung der public key cryptography für sich bzw. für den britischen Nachrichtendienst *Government Communications Headquarters* (GCHQ). Die dort seit den

Idee der drei Forscher in Verschlüsselungsalgorithmen um; ihr Produkt hieß *RSA* und galt als sichere Verschlüsselungstechnologie. Die Forschenden waren gleichzeitig Repressionen der *NSA* ausgesetzt, die nachdrücklich vor Verstößen gegen das Waffenrecht warnte: Kryptografische Software war als Waffe in der *US Munitions List* aufgeführt. 1979 ließ der Direktor der *NSA*, Bobby Inman verlautbaren: „[N]on-governmental cryptologic activity and publication (...) poses clear risks to the national security“ (zitiert bei Levy 1993). Die frühen Privacy-Aktivist_innen schufen nicht nur die Werkzeuge selbst, sondern mussten auch stark für ihre Verbreitung kämpfen, da starke Verschlüsselungssoftware in den USA bis zum Jahr 2000 Exportbeschränkungen unterlag.¹⁵ Diese Technologie-Exportbeschränkungen gab es in vielen Staaten, um dem damaligen Ostblock den Zugang zu modernen Technologien zu erschweren. Von 1949 bis 1994 stimmten sich die 17 Mitgliedsstaaten des *Koordinationsausschuss für mehrseitige Ausfuhrkontrollen*¹⁶ über Exportboykotts ab. Seit 1995/96 ist stattdessen das *Wassenaar-Abkommen für Exportkontrollen von konventionellen Waffen und doppelverwendungsfähigen Gütern und Technologien*¹⁷ in Kraft. Freie kryptografische Software ist laut diesem Abkommen von Ausfuhrbeschränkungen ausgenommen (vgl. Grassmuck 2004: 371).

Das erste asymmetrische Verschlüsselungsverfahren *RSA* wurde 1983 patentiert und von der Firma *RSA Data Security* vertrieben. Kritiker_innen fanden jedoch, dass das Konzept zu relevant sei, um es einer Regierung oder einer einzelnen, wenn auch gutmeinenden Firma zu überlassen, so dass die Beschäftigung mit Alternativen weiterging. In den 80er Jahren fanden Computer große Verbreitung im geschäftlichen und privaten Bereich, wodurch das Thema Verschlüsselung an Wichtigkeit gewann. Auch Diskurse über elektronische Überwachung und die Nutzung von Verschlüsselungstechnologien für kriminelle Zwecke kamen auf und wurden von Teilen der akademischen Kryptograf_innen mitgetragen (Levy 1993: vgl.).¹⁸

1960er Jahren betriebene Forschung und Entwicklung stand jedoch unter Geheimhaltung, so dass das US-amerikanische Trio zuerst damit an die Öffentlichkeit ging (vgl. Singh 1999: 279 ff.).

¹⁵US-Softwarekonzerne mussten beispielsweise zwei Varianten von → *Webbrowsern* mit unterschiedlich starker → *SSL-Verschlüsselung* anbieten: Die Browser für das Inland unterstützten eine sichere 128-Bit-Verschlüsselung, in die Browser für den Exportmarkt war nur eine knackbare 40-Bit-Verschlüsselung eingebaut (vgl. Wölbart 2008).

¹⁶Die englische Bezeichnung lautet *Coordinating Committee on Multilateral Export Controls (CoCom)*.

¹⁷Das Abkommen heißt auf englisch *The Wassenaar Arrangement on Export Controls for Conventional Arms and Dual-Use Goods and Technologies*. „Dual Use“ bezieht sich auf Technologien, die sowohl militärisch als auch zivil genutzt werden können.

¹⁸Bert-Jaap Koops (1999) untersucht Fragen des Gesetzesvollzugs bei der Nutzung von Verschlüsselungstechnologien für kriminelle Zwecke unter Berücksichtigung der Tatsache, dass dieselben Technologien auch legal (und legitim) eingesetzt werden.

Der zuvor in der Anti-Atom-Bewegung aktive Phil Zimmermann verfolgte die Auseinandersetzung und arbeitete ab Mitte der 80er Jahre zum Thema public-key-Kryptografie (vgl. Levy 1993). Er argumentierte für verschlüsselte Kommunikation, indem er elektronisch übermittelte Nachrichten mit Briefen verglich: Briefumschläge statt Postkarten zu verwenden erzeuge keinen Verdacht, weil es eine verbreitete Praxis sei. Wenn ebenso viele Menschen E-Mails verschlüsseln würden, würden auch die Verfechter_innen der Verschlüsselung nicht mehr verdächtig erscheinen. Diese Analogie zu Briefumschlägen wird bis heute in Argumentationen für verschlüsselte Kommunikation hergestellt. 1991 veröffentlichte Zimmermann das Programm *PGP (Pretty Good Privacy)* kostenlos im Internet. Es fand sofort große Verbreitung, zog jedoch auch Rechtsstreits nach sich, weil Zimmermann das patentrechtlich geschützte RSA verwendet hatte¹⁹ und weil die Software über das Internet sofort die USA verließ und damit das Exportverbot umging. Das FBI ermittelte ab 1993 gegen Zimmermann, der sich 1995 damit behelf, in Kooperation mit dem *MIT* den kompletten Quellcode als Buch zu publizieren und so legal zu exportieren (vgl. Bendrath 2001). Im selben Jahr bekam er Hilfe von der 1990 gegründeten Bürgerrechtsorganisation *Electronic Frontier Foundation (EFF)*. Die *EFF* unterstützte eine Klage gegen die US-Regierung: Das Exportverbot sei verfassungswidrig, weil es die Publikationsfreiheit einschränke.²⁰

„Der Konflikt zwischen der US-Regierung und Phil Zimmermann kann in seiner Bedeutung kaum überschätzt werden, denn er sorgte für eine wichtige Verschiebung in der Frage, wessen Sicherheit die Sicherheitspolitik zu schützen hat. Die Position [des] Außenministeriums, der NSA und des FBI war, dass die Verfügbarkeit starker Verschlüsselung im Ausland eine potenzielle Gefahr für die nationale Sicherheit des USA darstellen würde. Zimmermann und PGP sowie seine Unterstützer unter den amerikanischen Bürgerrechtlern hielten ein ganz anderes Verständnis von Sicherheit dagegen: Zu sichern sei nicht die Nation, sondern die Privatsphäre jedes einzelnen Bürgers. Anstelle des Staates stand so plötzlich das Individuum im Mittelpunkt der Überlegungen“ (Bendrath 2001).

1997 gab Zimmermann die Spezifikationen (die Produktbeschreibungen) von *PGP* gegenüber der *Internet Engineering Task Force (IETF)*²¹ frei, so dass auf der Basis von *PGP* ein

¹⁹Die Androhung von Klagen war hier gegenstandslos, weil der Arbeitgeber der RSA-Entwickler, das *MIT*, der formelle Besitzer des Patents war. Im Jahr 2000 ist das Patent ausgelaufen.

²⁰Die Argumentation, dass Code unter die freie Meinungsäußerung falle, wurde in Abschnitt 4.3 bereits erläutert.

²¹Die IETF (<http://www.ietf.org/>) ist ein Gremium unter dem formellen Dach der NGO *Internet Society (ISOC)*. Dort werden technische Standards der Internetinfrastruktur verhandelt.

offener Verschlüsselungsstandard, *OpenPGP*, entwickelt werden konnte. Eine der Implementierungen dieses offenen Standards ist der *GNU Privacy Guard (GnuPG)*, dessen erste Version noch 1997 erschien (vgl. Bendrath 2001; vgl. auch Grassmuck 2004: 375 f.). *GnuPG* wurde von Beginn an als „freie“ Software entwickelt, die nur patentfreie Verschlüsselungsalgorithmen verwendet. Der Quellcode ist transparent. Dies ist aus folgendem Grund relevant: Nur so ist überprüfbar, ob die Software keine Fehler oder absichtlich eingebauten Sicherheitsprobleme mit sich bringt. „Das Vertrauen in die Sicherheit eines Produktes ist nicht abhängig von dem Vertrauen, das man einer Person entgegenbringt“ (Bendrath 2001; vgl. auch Grassmuck 2004: 370).

Eine Bewegung, die die Verbreitung von Verschlüsselungstechnologien betrieben hat, ist die Cypherpunk-Bewegung, die sich in den späten 80er Jahren formierte. Die Cypherpunks sind ein informeller Zusammenhang von an Kryptographie und Online-Anonymität Interessierten. Zunächst kommunizierten sie über eine Mailingliste, ab November 1992 gab es auch (nicht-virtuelle) Treffen. Der Begriff Cypherpunk wurde von einer der wenigen damals in diesem Bereich aktiven Frauen, Jude Milhon, geprägt. Nach dem ersten Treffen veröffentlichte Timothy C. May das *Crypto Anarchist Manifesto*, das technologische Entwicklungen und staatliche Interventionen zueinander in Beziehung setzt: Kommunikations- und Verschlüsselungstechnologien würden die Möglichkeit schaffen, komplett anonym miteinander zu kommunizieren. Es sei technisch möglich, im Netz zu interagieren, ohne auffindbar zu sein. Das werde nicht nur ein Einschnitt in staatliche Regulierungsmöglichkeiten sein, sondern auch wirtschaftliche und soziale Beziehungen umwälzen.

„The State will of course try to slow or halt the spread of this technology, citing national security concerns, use of the technology by drug dealers and tax evaders, and fears of societal disintegration. Many of these concerns will be valid (...). Various criminal and foreign elements will be active users of CryptoNet. But this will not halt the spread of crypto anarchy“ (May 1992).

Wie auch aus dem 1993 veröffentlichten Text *A Cypherpunk's Manifesto* von Eric Hughes hervorgeht, halten Cypherpunks die Privatsphäre im elektronischen Zeitalter für eine Voraussetzung für eine offene Gesellschaft. Ohne es so zu bezeichnen, machen sie sich für das Recht auf informationelle Selbstbestimmung stark: „Privacy is the power to selectively reveal oneself to the world“ (Hughes 1993). Um zu gewährleisten, dass bei Online-Transaktionen nur die persönlichen Informationen preisgegeben werden, die im jeweili-

gen Zusammenhang nötig sind, seien Technologien zur Anonymisierung und Verschlüsselung von Daten unabdingbar. Wenn Menschen im Internet standardmäßig anonym auftreten, seien digitale Signaturen nötig, um gegebenenfalls die eigene Identität nachweisen zu können. Cypherpunks trauten weder staatlichen Institutionen, noch Firmen oder anderen Organisationen zu, die Privatsphäre von Bürger_innen oder Kund_innen ausreichend zu schützen. Mit der Motivation, sich selbst und alle Interessierten zu schützen, machten sie sich daran, Verschlüsselungs- und Anonymisierungssoftware zu schreiben, deren Quellcode sie frei zur Verfügung stellten (vgl. ebd.). Ein bekanntes Beispiel sind so genannte Remailer, die gewährleisten, dass E-Mails nicht zurückverfolgbar sind. Während die Cypherpunk-Bewegung an Anonymisierungssoftware arbeitete, um die Zurückverfolgbarkeit von Datenspuren zu verhindern, geht es beim Keysigning gerade darum, elektronische Daten eindeutig realen Personen zuordnen zu können.

Das *Web of Trust*, das sich über Keysigning bildet, ist ein Gegenentwurf zu zentralisierten Zertifizierungsstellen, *Certification Authorities (CAs)*, die gegen Geld Zertifikate ausstellen. Das selbstorganisierte, dezentrale und unkommerzielle *Web of Trust* stellt dem eine Community-Lösung entgegen (vgl. O'Neil 2009: 136).²² Laut Christopher Kelty ist aus Cypherpunk-Perspektive auch eine CA nicht vertrauenswürdig: „a completely distributed web-of-trust with mutually signed keys (...) is the only way around the necessity of trusting a stranger (CA) to guarantee an exchange“ (Kelty 2005c: 140).²³

Die politischen Kämpfe um die öffentliche bzw. zivile Verfügbarkeit starker Verschlüsselungstechnologien gehören zum historischen Wissensbestand von FLOSS-Communities: Es ist eine Errungenschaft, dass Verschlüsselungssoftware allen Interessierten, nicht nur der staatlichen, militärischen oder geheimdienstlichen Nutzung, offensteht. Die Grundlage des Konzeptes ist, dass jede Person den Schlüssel zu den eigenen Daten besitzt. Das favorisierte Programm *GnuPG* wird als freie Software entwickelt, damit keine unbemerkten Hintertüren eingebaut werden können. Die Schlüssel, mit denen hantiert wird, werden bewusst von der eigenen Szene, nicht von einer Firma, überprüft. Mit diesem Hintergrund komme ich zu den Keysigning-Parties.

²²Auf demselben Prinzip basiert auch die unkommerzielle Zertifizierungsstelle CACert, mit der sich alle Interessierten Zertifikate für verschlüsselte Websites oder eine anderes Verfahren zur E-Mailverschlüsselung (S/Mime) erstellen können.

²³Dass dieses Misstrauen gerechtfertigt ist, zeigt sich, wenn *Certification Authorities* das Vertrauen verlieren: Das jüngste Beispiel dafür ist die niederländische CA *DigiNotar*. Das Unternehmen gab im September 2011 bekannt, dass Unbekannte in die Computer eingedrungen waren und gefälschte Zertifikate erstellt hatten. Mit diesen Zertifikaten wurde die vermeintlich verschlüsselte Online-Kommunikation iranischer Bürger_innen überwacht (vgl. die tageszeitung, 02.09.2011).

7.2.2. Keysigning-Parties

Im ersten Teil der Keysigning-Party sitzen die Teilnehmer_innen in der Regel in einem Seminarraum beim Community-Event.²⁴ Die Sitzordnung ist frontal zum Organisator hin orientiert. Zuerst wird überprüft, ob alle die gleichen Prüfsummen für die Liste berechnet haben und wer der Angemeldeten anwesend ist. Es ist unterschiedlich, wie viele Erläuterungen es zum Verfahren oder zur Bedeutung von Keysigning gibt. Die Prüfsummen werden meistens vorgelesen, teilweise aber auch auf Zettel oder Wandtafeln geschrieben.

A56 begrüßt die Anwesenden mit einem „Willkommen“. Er gehe davon aus, dass alle ihre Keys hochgeladen hätten, angemeldet seien und einen hash berechnet hätten, oder besser zwei, noch besser drei (einige kichern). Ich schaue auf die Teilnehmer_innen-Liste meines Nachbarn. Tatsächlich ist dort Platz, um drei Prüfsummen einzutragen: *md5*, *sha1*, *sha256*. A56 fragt, wer noch nie an einer KSP teilgenommen hat, es melden sich zwei oder drei. Insgesamt sind weniger als 20 Personen da. Er beginnt zu erklären, was das *Web of Trust* ist. (...) Er resümiert nochmal die Schritte der KSP und bittet dann, die Listen zu nehmen und die Prüfsummen zu vergleichen. Er fange nun an, die Hashes zu verlesen. Mein Nachbar stellt die Frage, ob das dazu sei, sicherzustellen, dass alle die gleiche Liste haben? „Ja.“ A21 schlägt vor, die Prüfsummen an die Tafel zu schreiben. Ein Mitorganisator, A58, wendet ein: „Da nimmt man den Reiz. Es geht um's Murmeln.“²⁵ Man beschließt also, die Summen doch vorzulesen. A58 wird aufgefordert zu beginnen, er sagt, er hätte nur die *md5*-Summe berechnet. A56 bietet ihm an, dann die anderen Summen von seinem Blatt abzulesen, „wenn du mir vertraust“. „Nein“, sagt A58, Gelächter. Er liest die *md5*-Summe vor. Die anderen schauen auf die Zettel vor sich auf den Tischen. Manche verfolgen die Ziffern mit dem Stift oder dem Finger mit. Einer macht sich bei verschiedenen Teilen der langen Prüfsumme kleine Häkchen, wenn sie bis dahin übereinstimmte (KSP 2008c).

Der Vergleich von einer, zwei oder drei Prüfsummen der Datei mit der Teilnehmer_innenliste zeigt eine erste Annahme, die von vielen Anwesenden geteilt wird: Wenn eine Textdatei für alle zum Download bereitgestellt und ausgedruckt mitgebracht wird, dann kann *nicht* selbstverständlich davon ausgegangen werden, dass alle Listen identisch

²⁴Die Beobachtungen basieren auf vier Keysigning-Parties in den Jahren 2007 und 2008, die zwischen 20 und 80 Teilnehmer_innen hatten. An der ersten habe ich teilgenommen und später ein Gedächtnisprotokoll erstellt. Die folgenden Keysigning-Parties habe ich als Protokollantin verfolgt. Bei der dritten war ein Co-Beobachter bei mir, der ein eigenes Protokoll angefertigt hat.

²⁵Dies ist eine Anspielung auf ein Zitat, das häufig auf den Websites der Keysigning-Parties zu lesen ist: „Keysigning Party. Kultiges Zusammensitzen und gemeinsames Murmeln magischer Zahlen.“

sind. Mit den Prüfsummen wurde in das Ritual Keysigning-Party ein Mechanismus aufgenommen, der die Integrität der Liste vor der technischen Möglichkeit zur Manipulation eines Downloads schützen soll. Eine solche „man in the middle attack“ gehört zu den häufigsten „Angriffsszenarien“ im Internet.²⁶

Wenn über die Prüfsummen eine Einigkeit darüber hergestellt wurde, dass alle die gleichen Informationen über die Angemeldeten vor sich haben, werden alle aufgerufen und bestätigen, dass sie da sind und dass die Informationen über ihren Key korrekt sind. Die Keys von Personen, die nicht erschienen sind, werden von der Liste gestrichen.²⁷

Als ich meine Beobachtung beginne, fängt der Versammlungsleiter gerade an, eine Liste durchzugehen. Er nennt dabei immer wieder Nummern in aufsteigender Reihenfolge und dazu Namen. Die Genannten antworten und die Anwesenden machen sich dazu Notizen. Ich zähle die Anwesenden: 4 Frauen sind darunter, inklusive Silke. Nach mir kommen noch zwei oder drei Männer dazu. Ein Mann, der neben mir am Rand steht, fragt nach, ob Nummer 9 nun „okay“ sei und sagt anschließend „I thought it is in English“. Der Versammlungsleiter schaut den Sprechenden kurz an und antwortet dann: „No problem. I switch to English now.“ Als die Situation auf diese Weise offenbar geklärt ist, wenden sich in einem Moment der wahrnehmbaren Koordination viele sitzende Teilnehmer gleichzeitig wieder ihren Blöcken zu. (...) Ich sehe, dass die Teilnehmer neben mir ebenfalls Listen vor sich haben, auf denen sie sich Zeichen machen. Mir fällt auf, dass der Versammlungsleiter manchmal eine Nummer nennt und dazu einen Vornamen und einen Namen, manchmal aber auch nur den Vornamen. Bei einigen, die er nur mit Vornamen nennt, scheint er diese auch bereits anzuvisieren, bevor diese sich zu Wort melden. Die nur mit Vornamen Genannten scheinen dem Versammlungsleiter also persönlich bekannt zu sein. Manchen der nur beim Vornamen Genannten wird auch von anderen zugewinkt oder auf eine andere Art Erkennen signalisiert. (...) Es wurde ein Frauenname genannt. Einige TN richteten daraufhin den Blick nicht zum Versammlungsleiter, sondern in Richtung der Frau, die ihre Anwesenheit bestätigte. Mir fällt ein, dass es auch noch andere Ausnahmen von dieser Praxis gab, zum Beispiel Momente allgemeiner Heiterkeit, zum Beispiel als ein offenbar sehr vielen bekannter Mann noch nach mir den Raum betrat und zufällig auch gerade aufgerufen war (KSP 2008d).

²⁶Dabei klinkt sich jemand in die Verbindung zwischen Computern ein und fängt den Datenverkehr ab.

²⁷Bei einer beobachteten Keysigning-Party fiel dieser erste Teil komplett weg: „Am Raum, in dem die KSP stattfinden sollte, hängt ein Zettel, dass sie in den Kellergang verlegt worden sei“ (KSP 2008a). Dort hängen die Prüfsummen an einem Flipchart, die Überprüfung der tatsächlichen Anwesenheit lag bei den Teilnehmer_innen, während der Überprüfung der Ausweise. Es laufen also nicht alle Keysigning-Parties genau gleich ab.

Bei der Überprüfung der Anwesenheit werden auch bekannte Namen und Gesichter identifiziert: Community-Mitglieder begrüßen sich mit Gestik und Mimik während sie Notizen über die teilnehmenden Keys machen. Auch werden gleichzeitig Unbekannte identifiziert, die beispielsweise durch ihre Unkenntnis der deutschen Sprache selbst auf sich aufmerksam machen müssen. In der oben beobachteten Szene sind offenbar so wenig Frauen zugegen, dass eine vielen unbekannte Frau beim Aufrufen ihres Names bemerkt wird. Bei der Bestätigung der Anwesenheit wird die Gruppe implizit nach verschiedenen Kriterien geordnet: Bekanntheitsgrad innerhalb der Szene, Sprache, Herkunft und Geschlecht werden von den Teilnehmenden wahrgenommen und mit Reaktionen quittiert.

Im zweiten Teil der Party stellen sich alle auf. (Der Eindruck, dass es ein zweiter Teil ist, entstand sowohl bei mir als auch bei meinem Co-Beobachter durch die plötzliche Bewegung im Raum, nachdem zuvor alle ruhig und frontal angeordnet saßen.) Je nach den Räumlichkeiten verlassen manchmal alle den Seminarraum und gehen in einen langen Gang, um zwei Reihen zu bilden, sortiert nach der Nummer auf der Liste.²⁸

Viele scheinen zu wissen, was von ihnen erwartet wird. Alle Anwesenden stellen sich nun in zwei gegenüberstehenden Ketten auf. Dabei werden Nummern genannt. Ich deute dies so, dass die Anwesenden sich nach Nummern sortiert aufstellen sollen (KSP 2008d).

A56 sagt, es gebe nun verschiedene Methoden, er würde vorschlagen, dass alle sich draußen im Gang in zwei Reihen aufstellen. Alle verlassen den Raum und er erklärt draußen weiter, wie das vor sich gehen soll. Er ruft nochmal die Leute der Reihe nach auf und weist ihnen Plätze zu. A58 schlägt vor, dass eine Reihe doch auch reichen würde, an der man dann entlanggehen könnte. Es gibt eine Diskussion darüber, was schneller geht. Schließlich bleiben sie dabei, zwei Reihen zu bilden. (KSP 2008c).

A64 fängt an zu rufen: „Aufeinander zugehen! Das muss zusammenpassen! Anschließen [zu den weiter entfernt Stehenden]! Damit wir hier mal fertig werden! Das ist dein Partner. Immer einen weiter.“ (...) A68 und ein anderer stehen uns gegenüber, sie passen von der Nummer her jedoch hier nicht hin. A64: „Teilt euch richtig ein. Weg hier.“ A68 fragt: „Ach, hier gehts nach Nummern?“ A64: „Ja.“ (KSP 2008b)

²⁸Eine Keysigning-Party, die ich vor meiner teilnehmenden Beobachtung sah, lief ganz anders ab: Die Teilnehmenden blieben im Seminarraum sitzen. Jeweils eine Person ging nach vorne und legte ihren Ausweis auf einen Overheadprojektor, so dass alle ihn gleichzeitig sehen konnten und den Menschen ansprechen konnten.



Abb. 7.4.: Keysigning-Party 2007

Der Moment des Sortierens der Menschen nach Nummern ist ein kurzer koordinatori-scher Aufwand, in den sich der Organisator oder auch andere Teilnehmende (wie A64) mehr oder weniger stark einmischen: Die Anwesenden sollen ihre Körper im Raum an-ordnen und nach einem Muster aufeinander abstimmen (vgl. Dinkelaker 2010: 187 f.). Diejenigen, die noch nicht wissen, nach welchem Muster sie sich anordnen sollen, stören den reibungslosen Ablauf: Sie fallen durch ihre Unwissenheit auf und werden zurecht-gewiesen. Die jeweiligen Räumlichkeiten formen die Koordination der Körper mit: Fin-det die Identitätsüberprüfung in einem langen Kellergang statt, so stellt sich eine Reihe genau entlang der Wände auf. Abb. 7.4 zeigt ein sehr großes Foyer, in dem die Teilneh-mer_innen sich dennoch parallel zu Wand, Teppichmuster und Deckenverkleidung auf-reihen. Die Aufstellung ist von dem Wunsch geprägt, dass die Keysigning-Party möglichst schnell durchgeführt werden soll. Sie wird so gewählt, dass die Überprüfung möglichst rationalisiert ablaufen kann.

A61: „Es ist eben optimiert, weil, ich hab so andere Keysigning-Parties kennen gelernt und mitgemacht und in 'ner anderen Form, wo ich das Gefühl hab, es dauert einfach viel zu lange. Wenn du eine große Anzahl von Menschen zu signieren und zu prüfen hast, bei an die 200 Leuten, da lässt die Konzentration schon mal nach. (...) Wenn es genau so gemacht wird, dieses Verfahren, geht es am schnellsten“ (Int 2008b).

In diesem ersten Teil der Keysigning-Party werden die Anwesenden durch verschiedene Praktiken geordnet: Anhand der Liste ordnen sie sich Namen und Keys zu, anhand der Nummerierung auf der Liste stellen sie sich auf. Gleichzeitig findet eine implizite Ord-

nung nach ihrem Vorwissen und ihrer Sprache statt, denn diejenigen, die das Ritual nicht kennen oder die Sprache nicht verstehen, fallen auf.

Bei der Überprüfung der Identitäten der Anwesenden geht eine Menschenreihe langsam an der anderen Reihe entlang. Die einen halten ihre Ausweispapiere vor sich. Viele bestätigen zusätzlich, dass ihr Fingerprint korrekt auf der Liste steht; manche sagen es, andere halten neben ihren Papieren einen handgeschriebenen oder ausgedruckten Zettel, auf dem ihre Nummer und eine Information wie „FP OK“ (für „Fingerprint stimmt überein“) geschrieben steht. Zwischen vielen Gegenübern gibt es kurze Wortwechsel, die über die Bestätigung dieser Informationen hinausgehen:

A63 begrüßt jede_n mit ‚Hi‘ oder ‚Hallo‘ und nennt ihre Nummer. A70 führt mit seinem Nachbarn kleine Seitengespräche. Ich wechsele die Seite und stehe den beiden jetzt gegenüber. A70 hat zwei Nummern²⁹, sagen wir mal die 13 und die 14. Er erinnert seine Gegenüber oft daran, dass er noch eine zweite Nummer hat. Es werden Witze über die Personalausweise gemacht. Eine Frau schaut sich das Passfoto von A63 ganz genau an, woraufhin A63 sagt: „So’n richtiges Verbrecherfoto.“ Ein Mann sagt zu A63 (über ihren Personalausweis): „Der is’ ja ganz neu.“ In der Gruppe um die beiden fällt der Begriff „Fahndungsfoto“, es wird über die neuen Personalausweise gefrotzelt KSP (2008d).

Eine Frau begegnet A70. Sie hält einen blauen und einen grünen Ausweis hoch und sagt, als A70 stockt: „Sind beide gültig. Sind einfach zwei Staaten“ (KSP 2008d).

Später am Stand [auf dem Community-Event] steht der Mann, der sich auf der Keysigning-Party mit dem Schweizer Führerschein ausgewiesen hatte. Er erzählt, dass einer da war, der den Führerschein als Ausweisdokument abgelehnt habe (KSP 2008b).

A67, der Niederländer, der sich am Anfang wegen der Sprache der Veranstaltung gemeldet hatte, steht vor A64 in der Reihe. Ein Gegenüber und A64 witzeln herum, weil sie wohl nicht sicher feststellen können, ob der Ausweis gültig ist. A64 sagt: „Self-made in color printer.“

A10 ruft: „Seit zwei Jahren abgelaufen!“ Sein Gegenüber lacht. A10 notiert sich einen dicken Kreis um den Namen auf seiner Liste und sagt: „Hier ist ein Fragezeichen“ (KSP 2008c).

²⁹Er hat zwei separate Keys zur Keysigning-Party angemeldet.

Dann kommt ein Mann mit österreichischem Personalausweis dran und dann ein Niederländer. A70 fragt sehr genau nach, auch persönliche Dinge, wobei er den Niederländer ein paar mal auf Deutsch nach seinem Wohnort fragt, bis er begreift, dass dieser nur Englisch versteht (KSP 2008d).

Etwas weiter in der Kette begegnen A63 und A70 einem Mann, dessen Nachname aus im Deutschen unüblichen Kombinationen von Konsonanten bestand. Ich höre ihn noch zu A70 sagen: „Aus Bosnien.“ A63 fragt: „Wie spricht man diesen Namen aus?“ Der Mann spricht ihn vor, A63 spricht nach und sagt: „Cool“ (KSP 2008d).

Einer (mit deutschem Personalausweis) kommt uns entgegen. A64 fragt, woher der Name komme, aus Tschechien? „Nein, aus Bosnien.“ A64: „Schön, dass du hier bist. Es sind viel zu wenig Leute aus Bosnien hier“ (KSP 2008b).

Eine Frau kommt an A21 vorbei und sagt: „Du kommst ja demnächst zum Vortrag zu uns. Ich organisiere da mit“ (KSP 2008a).

A64 nochmal: „Ich sehe drei Frauen in der ganzen Reihe und A10 bleibt stehen.“ Er ruft laut in Richtung A10: „A10, es wird behauptet, du bleibst immer bei den Frauen kleben.“ A10: „Dem widerspreche ich nicht“ (KSP 2008b).

A115: „Du müsstest einen Satz auf Schwäbisch sagen.“ (...) Der A116 links von mir fragt: „Warum ist dir das so wichtig?“ A115: „Ich mach meine eigene Identitätsüberprüfung“ (KSP 2008b).

Ein älterer Mann fällt mir auf, der der Person, die an ihm vorbeigeht, nicht ins Gesicht sieht. Ich beobachte ihn einen Moment: Bei den nächsten vier Personen, blickt er auf den Ausweis, dann auf seine Liste. Er scheint den Namen zu vergleichen. Die Personen sieht er nicht an. Die fünfte Person ist ein Bekannter, den er ansieht und begrüßt. Dann kommt die Nr. 85 an ihm vorbei, er blickt auf die Liste, macht eine kurze Bemerkung und hakt ab (KSP 2008a).

Die Beobachtungen zeigen, dass die amtlichen Dokumente Aufhänger für Zwischengespräche sind. Manche machen Witze über die neuen deutschen Personalausweise und äußern damit eine implizite politische Kritik an der Erfassung biometrischer Daten für die Dokumente („Verbrecherfoto“, „Fahndungsfoto“). Die Interaktionen zeigen auch implizite Annahmen der Teilnehmer_innen, z. B., dass ein Ausweisdokument die einzige Möglichkeit zur Identitätsfeststellung ist, dass eine Person genau eine Staatsangehörigkeit hat, dass Führerscheine (in Deutschland) keine anerkannten amtlichen Ausweispapiere sind

und dass abgelaufene Dokumente die Identität einer Person nicht sicher nachweisen. Zudem wird die Tatsache aufgegriffen, dass nicht alle die Echtheit eines ausländischen Dokumentes anhand von spezifischen Merkmalen erkennen können.

Das Ritual lässt den Einzelnen Raum für die Entscheidung, ob sie die Keys signieren oder nicht. Dazu gehört auch die individuelle Entscheidung, die vorgelegten Dokumente der anderen Anwesenden zu akzeptieren oder abzulehnen. Die Organisatoren zweier Keysigning-Parties explizieren auf Nachfrage hin ihren Umgang mit abgelaufenen Ausweisen:

A21: Es muss halt jeder Teilnehmer für sich entscheiden, ob er diesem Ausweisdokument vertraut oder nicht vertraut. Und beim Führerschein tendiere ich eher noch dazu, dann zu unterschreiben, wobei das halt schon äußerst grenzwertig ist, aber alle anderen Dokumente, die sind dann für mich einfach wertlos. Und was halt auch immer wieder passiert, diesmal ist es nur einmal passiert, dass die Ausweise abgelaufen sind. Und dann ist es halt auch so'ne Frage. Also hier, der war jetzt irgendwie drei, vier Tage abgelaufen, da denk ich, hab ich kein Problem, aber wenn n paar Monate das abgelaufen ist...

S. M.: Was ist dann dein Problem damit?

A21: Einfach das ist kein gültiges Dokument, das ist so'n prinzipielles Problem. Sicherlich, der Ausweis würde noch passen, aber es ist halt irgendwo schwer, so'ne absolute Grenze zu ziehen. Und meine Grenze ist dann, zu sagen, wenn der Ausweis abgelaufen ist, dann is' halt Pech, er hat kein gültiges Dokument gehabt. (kurze Pause) Das mag sich radikal anhören,... aber irgendwo muss man halt 'ne Grenze finden (Int 2008a).

S. M.: Und wenn du Leute jetzt schon aus 'nem Projekt kennst, ist es denkbar, sich zu signieren, ohne den Personalausweis anzugucken?

A61: Manchmal möchte man das tun, wenn man jemanden schon jahrelang kennt, ja? Aber es ist eben so'ne Vorgabe, dass du wirklich prüfst, ist es wirklich derjenige, für den er sich ausgibt? Und diese Prüfung des Personalausweises, auch die Gültigkeit des Personalausweises ist für mich schon wichtig. Und ich habe auch schon es abgelehnt, den Schlüssel zu signieren, wenn der Ausweis abgelaufen war. Oder dass mal einer kommt, der 'n Führerschein oder irgend so'n, das lehn ich dann auch ab. Ich mach die Prüfung schon sehr genau, (...) ich guck mir den Ausweis an: Ist er gültig? Passt das Lichtbild zu demjenigen? Die Abfrage des Fingerprints, das mach ich sehr genau“ (Int 2008b).

Die Regeln des Rituals erfordern es, eine *genaue* Prüfung der Identität des Gegenüber vorzunehmen. Dazu muss ein gültiger Ausweis vorgelegt werden. Die Einzelnen handhaben die Dokumente, die von dieser Norm abweichen unterschiedlich und haben offenbar auch unterschiedlich viel Verständnis dafür, dass Menschen anderer Herkunft andere Dokumente für selbstverständlich halten.

7.2.3. Das Web of Trust

Das *Web of Trust* ist das Netz, das sich aus den Signaturen an den Schlüsseln ergibt: Wer hat wessen Schlüssel signiert, wie gut ist ein Schlüssel dadurch mit anderen Schlüsseln verlinkt? Keysigner_innen schenken sich mit der Signatur ein Vertrauen, das es ihnen ermöglicht, an die Echtheit der Schlüssel Dritter und Vierter zu glauben, wenn sie sehen können, dass deren Schlüssel auch von anderen überprüft wurden, im Idealfall von Personen, denen sie selbst vertrauen. Manche Organisatoren von Parties veranschaulichen das bei der Party gewachsene Netz zwischen den Schlüsseln der Teilnehmenden im Nachhinein auf den Websites der Community-Events (vgl. Abb. 7.5 auf S. 235). Der Aufbau des *Web of Trust* selbst ist ein wichtiger Grund für die Teilnahme am Keysigning. Nach meiner ersten Teilnahme an einer Keysigning-Party habe ich notiert:

Ich bekomme keine zufriedenstellende Erklärung dafür, warum Leute immer wieder auf Keysigning-Parties gehen. Warum reichen nicht die Signaturen von einer Party, um den eigenen Schlüssel glaubhaft zu machen? (Nach 10 Tagen habe ich immerhin von 28 Menschen meinen signierten Schlüssel zugeschickt bekommen. Wenn das online ersichtlich ist, ist mein Schlüssel dann nicht vertrauenswürdig?) Mein Informant sagt dazu spontan: „Man geht da immer wieder hin, damit das *Web of Trust* ausgebaut wird und immer neue Knoten dazukommen“ (KSP 2007).

S. M.: Was schätzt du, auf wie vielen Keysigning-Parties du schon warst?

A61: Uuu. (7 sek. Pause) Etwa 5, 6 Keysigning-Parties pro Jahr... Und dann das ganze jeweils die letzten acht Jahre, also irgendwas zwischen 30 und 40 bestimmt schon.

S. M.: Was ist dir daran wichtig? Und warum gehst du so oft hin?

A61: Mir ist halt wichtig, dass dieses *Web of Trust* gestärkt wird, ja? Und dann is es halt 'ne Voraussetzung, dass du an diesen Keysigning-Parties teilnimmst, ja? Und du lernst andere Menschen eben kennen, ja? Also am liebsten hab ich's natürlich, wenn ich die auch so aus Projekten schon kenn, ja? Und dann hast du mit den Leuten schon gearbeitet, du weißt, wie sie arbeiten, wie sie als Mensch sind, soweit man das eben

beurteilen kann. Und dann auf der Keysigning-Party sagt man, mensch, ich vertrau dir, deinem Schlüssel. Das ist so'n Anfang für ne, ja vielleicht Mail-Partnerschaft (Int 2008b).

S.M.: Wie würdest du erklären, warum das *Web of Trust* so wichtig ist? Oder was ist daran das Vertrauen?

A21: Also, es ist ja zunächst erstmal so, einen Schlüssel kann ich ja mit 'nem beliebigen Namen erzeugen. Also, man suche nur nach Bill Gates bei den Keyservern. Da findet man irgendwie, ich weiß nich, x Schlüssel. (...) Das heißt, wenn ich jetzt auf die Idee käme, Bill Gates eine verschlüsselte E-Mail zu schreiben, wüsst' ich halt nicht, wer jetzt eventuell, falls er denn überhaupt 'nen Schlüssel hätte, (...) die richtige Person ist, der richtige Schlüssel zu der Person. Und wenn man halt so'n *Web of Trust* einmal aufgebaut hat, das heißt, (...) ich vertrau halt 'ner Gruppe von Personen. Und die Gruppe von Personen vertraut vielleicht wieder anderen Personen, dann kann ich halt über so Pfade halt dann einfach versuchen rauszufinden, okay, der Schlüssel gehört wirklich der Person, weil mein Freund vertraut seinem Freund usw. (Int 2008a).

A61: (...) neue Leute, da probier ich, mich auch wirklich vorzustellen, und dann wart ich auch auf Fragen, die sie haben. Zum Beispiel auf meinem Ausweisfoto hab ich da 'n Bart, der ist jetzt diesmal weg. Und dann kommen da schon mal so Fragen. Da versuch ich höflich drauf zu reagieren, weil es auch da so'n Moment, so, der erste Eindruck, ja? Wenn du da so schlurhaft hinkommst, wie genau traut er dir zu oder ja, was denkt der von dir, wie genau du dich in diesem Sicherheitsumfeld, im *Web of Trust* bewegst. Ja, wenn du da einfach so hinkommst, kein' Bock, und so, stell ich mich vor, er sieht mich, meinen Ausweis, ich beantworte Fragen, höflich, dann hoff' ich, er kriegt auch 'n Gefühl, wie ernst mir die Sache ist, ja? Das ist mir sehr wichtig. Es gibt also andere Menschen, die sagen, mir isses egal, die kommen zu spät und stören, ja. Aber mir isses wichtig, auch nach außen zu zeigen, das is 'ne wichtige Veranstaltung, du hast Fragen, ich werd sie dir beantworten, du brauchst meine Dokumente, die zeig ich (Int 2008b).

A61: Egal, in welcher Form du Vertrauen 'ner anderen Person oder Institution gegenüber hast, das geht immer nur über Kontrolle. Du kannst nicht einfach sagen, blindlings, dieser Person vertraue ich. Unterbewusst denk ich immer, du brauchst so 'ne Rückmeldung, ob das investierte Vertrauen auch wirklich so getragen wird, so weiterbringt und es wird immer in irgend'ner Form 'ne Kontrolle geben. (...) Und da passiert wirklich ganz offensichtlich, mit Ausweis, wie an der Grenze, du wirst gemustert und so, aber das Vertrauen funktioniert halt so. Und gerade für das *Web of Trust*, das seh ich so, es muss genauso gemacht werden (Int 2008b).

A61: Und es ist für mich so mein Vertrauensnetzwerk. Ich hab da meine Freunde, die Leute, die kennen mich, meine Telefonnummern sind bekannt, wo man mich treffen kann, in welchen → *IRC*-Kanälen und und und, in welchem → *Repository* von mir Dateien liegen, die sich wer abrufen kann, das ist bekannt. Das sind schon, kann man sagen, recht intime Daten, ja? Das sind auch Entwicklungen, die vielleicht nur halbfertig sind. Auch 'n → *Git*-Repository liegt da rum. Und das sind Freunde, denen ich vertraue. Und die gehen mit diesen sensiblen Daten auch so um, wie ich mit deren Daten umgehen würde. Und das ist mir super wichtig, ja? (...) aber eben Daten, E-Mail verschlüsseln, signieren und so weiter, und so fort, Repositories und Dateisysteme verschlüsselt. Und da haben eben Freunde drauf Zugriff. Und wir haben unser Vertrauen gegenseitig eben bestätigt - hm, und das ist für mich wichtig. Enorm wichtig. (...) Wenn jemand von mir 'n Key kriegt und weiß, wie er an meine Repositories kommt, da kommen wirklich nur Leute dran, denen ich 100 Prozent vertraue, ja? (...) Und andere eben nicht. Das ist *mein* (betont) Netzwerk (Int 2008b).

Die Interviews mit zwei Organisatoren von Keysigning-Parties zeigen, wie facettenreich das Ritual für sie ist: Beide nehmen daran teil, um das *Web of Trust* auszubauen. A21 beschreibt, dass er einer Personengruppe „vertraut“, in der jede Person wiederum die Schlüssel anderer Personen signiert hat. Dies helfe ihm bei der Orientierung auf Keyservern dabei, herauszufinden, ob ein Schlüssel wirklich zu der Person gehört, mit der er kommunizieren möchte.³⁰ A61 verbindet mit dem *Web of Trust* weitaus mehr: Beim persönlichen Zusammentreffen mit Gleichgesinnten, die er möglicherweise bereits aus der Online-Kommunikation kennt, macht er sich einen Eindruck der Personen und möchte selbst einen bestimmten Eindruck hinterlassen. Er spricht das Knüpfen sozialer Netzwerke für Kooperationen an, z. B. für gemeinsame Softwareentwicklung über das Internet. Das *Web of Trust* ist für ihn ein „Sicherheitsumfeld“, in dem es darum geht, öffentlich zu zeigen, dass man sich der Sensibilität digitaler Daten bewusst ist. Er legt deshalb Wert auf höfliches Auftreten und die Demonstration präziser Identitätsüberprüfungen, distanziert sich von „schlurhaftem“ Auftreten. An bestimmten Orten im Internet hat er Daten hinterlegt, die er nur einem bestimmten vertrauenswürdigen Personenkreis zugänglich macht, mit denen er Keys signiert hat. Von diesen Personen erwartet er ein ähnliches Verständnis für den Umgang mit seinen Daten. A61 möchte demnach nicht nur dem *Web of Trust* selbst als Authentifizierungsmechanismus vertrauen, sondern auch den Einzelpersonen,

³⁰Er bezieht sich beispielhaft auf den *Microsoft*-Gründer Bill Gates, der in FLOSS-Communities zur Symbolfigur proprietärer Software geworden ist. Viele manchen sich offenbar einen Spaß daraus, Keys auf seinen Namen zu generieren.

die es ausmachen. Dieses Vertrauen kann für ihn nur durch Kontrolle der Ausweispapiere hergestellt werden. Gültige amtliche Dokumente sind die wichtigste Voraussetzung dafür.

Das *Web of Trust* bzw. das beim Keysigning geschenkte Vertrauen beinhaltet noch weitere Aspekte: Die Abstufung von Vertrauen in der Software *GnuPG*, die Rangliste der am besten verlinkten Keys und Gegenseitigkeit.

Erstens bietet die Software *GnuPG*, mit der die Keys signiert werden, die Möglichkeit, Vertrauen abgestuft zu schenken (vgl. Abb. 7.1 auf S. 234). Der Grad des Vertrauen ist in Form von entsprechend angereicherten Daten objektivierbar. Einerseits sieht *GnuPG* vier „certificate check level“ vor, die online sichtbar sind. Beim Signieren kann man auf einer Skala von null bis drei angeben, wie genau man die Identitätsprüfung durchgeführt hat. Andererseits bietet die Software die Möglichkeit, sich persönliche, nicht online sichtbare, Informationen zum „trust level“ zu hinterlegen und so festzuhalten, wie stark man jemandem vertraut. In der Dokumentation des Programmes *GnuPG* steht dazu, Vertrauen sei in der Praxis subjektiv. Wenn jemand zwar einen Key signiere, aber wisse, dass die Person, der der Schlüssel gehört, Schlüssel nicht genau überprüfe, so müsse das in der Signatur abbildbar sein. Dazu gibt es die Vertrauensstufe „kein Vertrauen“. „Marginales“ Vertrauen bedeutet, dass der / die Schlüsselinhabern die Implikationen von Keysigning versteht und Schlüssel vor dem Signieren gründlich überprüft. „Volles“ Vertrauen wird so beschrieben, dass die Person ein exzellentes Verständnis von Keysigning hat und ihre Signatur an einem Schlüssel so viel wert ist wie die eigene.³¹ Auch wenn sie heterogene Vergabepaxen für die *trust level* haben, stimmten die Interviewten darin überein, dass fremde Keys zumindest zunächst ein geringeres Vertrauen genießen als die eigenen Schlüssel. Vertrauensbildung findet prozesshaft statt und „im ersten Durchlauf spielt man nicht hoch“ (Luhmann 1973: 47).

[I]m Wesentlichen sind wir anonyme Leute. Die meisten kenn ich nicht. Und denen kann ich natürlich auch kein hohes Vertrauen ... zunächst erstmal schenken. Also das ist jetzt *mein* Ansatz. (...) Der normale Keysigning-Teilnehmer kriegt halt einfach 'n Standardwert, der möglichst niedrig eingestellt ist (Int 2008a).

³¹ vgl. <http://www.gnupg.org/gph/en/manual.html>, Abruf: 02.08.2011.

³² <http://www.GnuPG.org/gph/en/manual.html>, Abruf: 02.08.2011

³³ <http://www.gnupg.org/documentation/manuals/gnupg-devel/GPG-Configuration-Options.html>, Abruf: 02.08.2011

Bezifferbares Vertrauen in GnuPG	
Trust level: Wie vertrauenswürdig ist die Person?³²	
unknown	Nothing is known about the owner's judgment in key signing. Keys on your public keyring that you do not own initially have this trust level.
none	The owner is known to improperly sign other keys.
marginal	The owner understands the implications of key signing and properly validates keys before signing them.
full	The owner has an excellent understanding of key signing, and his signature on a key would be as good as your own.
Check level: Wie genau wurde die Identitätsüberprüfung durchgeführt?³³	
0	means you make no particular claim as to how carefully you verified the key.
1	means you believe the key is owned by the person who claims to own it but you could not, or did not verify the key at all. This is useful for a „persona“ verification, where you sign the key of a pseudonymous user.
2	means you did casual verification of the key. For example, this could mean that you verified that the key fingerprint and checked the user ID on the key against a photo ID.
3	means you did extensive verification of the key. For example, this could mean that you verified the key fingerprint with the owner of the key in person, and that you checked, by means of a hard to forge document with a photo ID (such as a passport) that the name of the key owner matches the name in the user ID on the key, and finally that you verified (by exchange of email) that the email address on the key belongs to the key owner.

Tabelle 7.1.: Bezifferbares Vertrauen: *Trust level* und *check level* in der Software *GnuPG*

Letzteres erklärt A112 mir später in einer E-Mail: „Tatsächlich ist die Berechnung der Positionen etwas komplizierter zu beschreiben als ein einfaches Zählen der Unterschriften: Wenn man zwei Schlüssel hat, kann man versuchen, anhand von Unterschriften einen Weg von einem Schlüssel zum anderen zu finden. Als Maß für die Länge dieses Weges zählt man einfach die benötigten Unterschriften. (Beispielsweise kann man mit einer Unterschrift von meinem Schlüssel direkt zu Deinem gelangen. Mein Bruder würde von seinem Schlüssel zu Deinem schon zwei Unterschriften benötigen, eine zu meinem Schlüssel und eine weitere von meinem Schlüssel zu Deinem). Für jeden Schlüssel kann man nun berechnen wie viele Unterschriften man durchschnittlich benötigt, um zu diesem Schlüssel zu gelangen.³⁴ Nach dieser durchschnittlichen Anzahl von Unterschriften ist die Liste sortiert. In der Liste vom xx. xx. 2007 taucht Dein Schlüssel gar nicht auf, weil er von den meisten Schlüsseln noch nicht erreichbar war. Allerdings hat Dich (...) A64 (mit einem derzeitigen Listenplatz 6) unterzeichnet, was mir eine (mit Sicherheit sehr ungenaue) Schätzung Deiner Listenposition nach der nächsten Aktualisierung erlaubt: A64s Schlüssel hat eine durchschnittliche Weglänge von 3.6684 Unterschriften, also hat Dein Schlüssel eine durchschnittliche Weglänge von maximal 4.6684 Unterschriften. Diese Weglänge sehe ich in der Liste bei fünf Schlüsseln an den Positionen 3701 bis 3705. Dein Schlüssel wird aber wahrscheinlich weiter oben in der Liste auftauchen, da er neben A64s Unterschrift auch Unterschriften von anderen Schlüsseln enthält“ (E-Mail von A112).

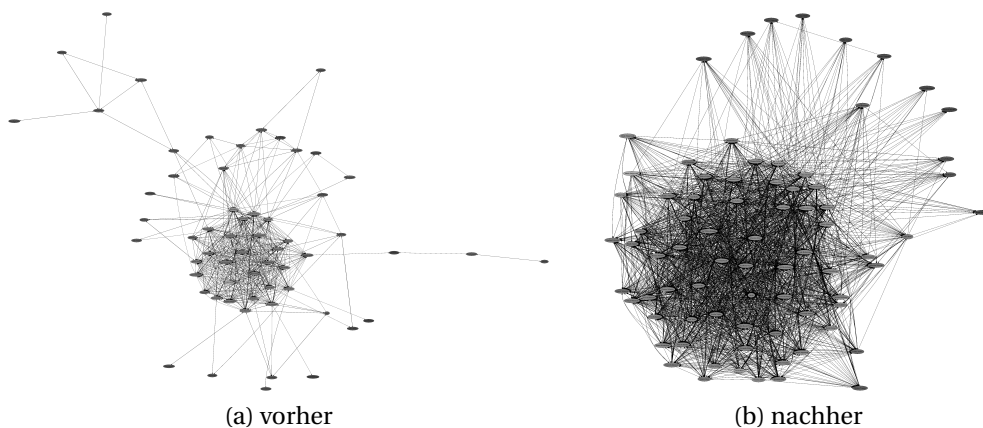


Abb. 7.5.: *Web for Trust* vor und nach einer Keysigning-Party

Bei einer Keysigning-Party wurde explizit Bezug auf diese Rangliste genommen: Auf der Liste der Teilnehmer_innen waren hier die aktuellen Listenplätze explizit vermerkt (vgl.

³⁴http://pgp.cs.uu.nl/mk_path.cgi, Abruf: 14.06.2011

Abb. 7.2, S. 215). Die regelmäßige Teilnahme an Keysigning-Parties bringt den Beteiligten einen sozialen Status ein.

A61: „Du bist auf Rang vier, mein lieber Mann!“ A64 zeigt mir, dass der Rang auf der Liste mit vermerkt ist (KSP 2008b).

Drittens ist beim Signieren der Keys Gegenseitigkeit wichtig, damit die Vertrauensbeziehung zustande kommt. Einzelpersonen fallen in der Community jedoch dadurch auf, dass sie keine Signaturen zurückgeben, sondern nur selbst welche sammeln. Solch abweichendes Verhalten wird registriert und kritisiert.

A61: A69 reicht immer zwei Schlüssel ein, mit einem signiert er auch andere Leute. Und einer dient nur als Signatursammler. Mit dem hat er mittlerweile, ist er auf Position 29 oder so. Also, er hat's ganz nach oben schon geschafft, aber er signiert nur mit einem dieser beiden Schlüssel und den anderen lässt er nur signieren. Da sag ich (...), das Wichtigste ist das Gegenseitige bei der Signatur (Int 2008b).

Neben dem kritisierten A69, der die Gegenseitigkeit des Signieren missachtet, hat noch ein Teilnehmer die Grenze akzeptabler Abweichungen überschritten, indem er über Jahre an Keysigning-Parties teilgenommen hat, seinen Schlüssel signieren ließ, aber den anderen keine signierten Schlüssel zuschickt hat. Er wird nicht nur in einem Interview thematisiert, sondern auch einige Monate später während einer Party direkt dafür kritisiert.

A61: A10 macht eine genaue Prüfung und fragt die Leute ab, kannst du mir sagen, was du für ein Verschlüsselungsverfahren benutzt? Wie lang ist dein Schlüssel? Der stellt 700 Fragen, wenn er einen gegenüber hat, versäumt es aber selbst nachher, die Schlüssel zu signieren. Der besucht Tausende von Parties und signiert keinen Schlüssel. Da krieg ich so'n Hals, ja? (...) Jetzt hab ich ihn nochmal angeschrieben, als ich gemerkt hab, er hat den Schlüssel für meine Party eben eingereicht. Da sag ich, A10, dein Schlüssel fliegt raus, wenn du mich nicht bald signierst. Also du sammelst nur Signaturen und leistest nichts dafür (Int 2008b).

Nachdem er die Liste durchgegangen ist, fragt er, ob jemand da ist, der außerdem noch teilnehmen möchte. Ja, einer meldet sich. Er sagt aber, er habe aber keine Zettel dabei. „Ich würde eure Schlüssel signieren, das ist doch okay?“ Nun beginnt eine kurze Diskussion darüber, ob das geht. A10 wendet ein, dass die Gegenseitigkeit wichtig sei. Die junge Frau ruft in den Raum: „Ja, A10!“ (Sie spielt wohl auf das an, was mir auch schon über A10 erzählt worden war, dass er keine Schlüssel von anderen signiert.) Mehrere steigen darauf ein und fordern A10 auf, sich da selbst mal dran zu

halten. A10 fragt den einen, ob er sich noch schnell Zettel besorgen könnte. Es geht hin und her. Die junge Frau ruft: „Lasst ihn mitspielen!“ (KSP 2008c)

Die Situation ist unüblich: Wer nicht angemeldet ist, kann am Keysigning teilnehmen, wenn sie_er kleine Zettel mitbringt, auf denen die Informationen zum Key stehen, und zwar so viele Zettel, dass alle Gegenüber einen bekommen. In dieser Szene erscheint jemand, der sich keine Zettel ausgedruckt hat und nun anbietet, nur die Schlüssel anderer zu signieren und selbst auf Signaturen zu verzichten. A10 betont nun die Relevanz der Reziprozität beim Keysigning, die er selbst nicht gewährt. Die Listenplätze bilden demnach ab, an wie vielen Parties jemand teilgenommen hat. Sie bilden jedoch *nicht* unbedingt ab, wie hoch das Vertrauen ist, das jemand genießt. A10 hatte auf einer solchen weltweiten Keyserver-Statistik 2007 einen Platz um die 60, obwohl bei den Begegnungen deutliche Zweifel daran geäußert werden, dass er die Maßgabe der Gegenseitigkeit einhält. Im Moment des eigenen Signierens (nach der Party am Computer) werden Schlüssel signiert und an die Inhaber_innen geschickt, ohne zu wissen, ob das Vertrauen auf Gegenseitigkeit beruht. Hier äußern sich die Unzufriedenen, die A10 einen Vertrauensvorschuss gegeben hatten. A10 selbst äußerte sich im Gespräch so, dass er das Signieren jederzeit noch erledigen könne. Wichtig sei, dass der „Handshake“ zustande komme, egal, ob ein Tag oder ein Jahr dazwischen liege. Diese Einschätzung wird offenbar von seinen Gegenübern nicht geteilt.

7.2.4. Keysigning-Parties als ordnende und hierarchisierende Praxis

Das Ritual des Keysigning folgt demnach einer inneren Logik, die wie folgt beschrieben werden kann: Die Möglichkeit, verschlüsselt zu kommunizieren, ist ein erkämpftes Gut. Die *public key cryptography* wurde entwickelt, um allen Menschen diese Möglichkeit einzuräumen. Freie Software ist in diesem Zusammenhang besonders wichtig, weil nur so sichergestellt werden kann, dass keine dritte Stelle eine Hintertür in die Software eingebaut hat. Die *public keys* werden auf Keyservern veröffentlicht, auf die jede_r Schlüssel hochladen und einsehen kann. Es gilt jedoch sicherzustellen, dass man wirklich den passenden Schlüssel der Person herunterlädt, mit der man kommunizieren möchte. Statt diese Zuordnung einer Firma oder einer staatlichen Institution zu überlassen, wird auch sie von der Szene selbst organisiert. Da es unmöglich ist, dass alle Personen sich persönlich treffen, gibt es die Idee des *Web of Trust*: Andere können es übernehmen, an weiter entfernt liegenden Orten Menschen und Schlüssel einander zuzuordnen. Diesen anderen kann um

so mehr Vertrauen geschenkt werden, je mehr Personen aus dem eigenen Umfeld sie getroffen und ihre Keys signiert haben. Es gilt, das *Web of Trust* so eng wie möglich zu knüpfen, um die Informationen auf den Keyservern so vertrauenswürdig wie möglich zu gestalten (vgl. den Workshopbeitrag von O'Mahony/Ferraro (2003), zitiert bei O'Neil 2009: 137). Keysigning ist der körperliche Akt, bei dem für die Verbindung gebürgt wird, die zwischen einem *public key* und der Person oder Organisation, auf die der Key ausgestellt ist, hergestellt wird (vgl. ebd.). Dieser Akt stellt damit die Integrität der Kommunikationspartner_innen her. Gut besuchte Community-Events eignen sich dafür, Keysigning in großem Maßstab durchzuführen. Um Vertrauen herzustellen, bedarf es der Person selbst, ihrer ausgedruckten Key-Informationen, ihrer von einer staatlichen Stelle ausgegebenen gültigen Ausweispapiere und der persönlichen Zusicherung, dass die Key-Informationen wirklich ihre sind. Vertrauen ist subjektiv, so dass die Key-Verwaltungssoftware Abstufungen zulässt. Die Teilnehmer_innen prüfen auch unterschiedlich präzise, was ebenfalls in der Software abgebildet werden kann. Die erwünschte Vertrauensbeziehung, die beim Keysigning entsteht, ist gegenseitig: Beide Signing-Partner_innen schicken sich die signierten Keys im Nachhinein zu, und zwar in einem selten definierten, aber nicht zu großen Zeitabstand nach dem Zusammentreffen.

Soweit das Ritual. Wenn man nun einen Schritt zurücktritt und Keysigning aus der „ethnozentristischen Perspektive des Beobachters“ (Bourdieu) betrachtet, fällt der Blick auf etliche Praktiken und Selbstverständlichkeiten beim Keysigning, von denen drei im Folgenden genauer beleuchtet werden: Die Vertrauensbegriffe, die im Keysigning zum Ausdruck kommen, Keysigning als symbolischer Kapitaltausch und Keysigning als Distinktionspraxis, die spezifische Normen hervorbringt.

Auf den ersten Punkt, das Vertrauen gehen Kelty und Lloyd in ihren Arbeiten ein: Laut Kelty bedeutet Vertrauen beim Keysigning nicht, der Person, der Software oder den signierten bzw. verschlüsselten Daten zu vertrauen, sondern der Person zu vertrauen, die eine Identität überprüft hat. Die Kryptografie stelle kein eigentliches Vertrauen her, sondern mache Vertrauen zu einer kalkulierbaren Größe.³⁵ Lloyd stimmt dem zu: Beim Key-

³⁵ „The notion of who, or what should be trusted is the most interesting - and deeply troubling - aspect of how cryptography is related to human bodies and the classical ways in which they establish trust between each other (by group membership, by repeated interaction, by eye contact or in the case of economic transactions, through the use of money). (...) Trust, in this instance, does not mean whether I trust either the person, the software, or his/her authenticated message - it refers to whether I trust the authenticator. The existence of cryptography does not ensure trust, it produces a situation in which trust is made into a more 'necessary, uniform, like among like, regular, and consequently calculable' entity“ (Kelty 2005c: 137).

signing werde eine rein technische Überprüfung von Identitäten durchgeführt, so dass man darauf vertrauen könne, dass Online-Identitäten mit offiziell bestätigten Identitäten übereinstimmten. Einer seiner Interviewpartner nennt die signierten Schlüssel „virtual passports“ (Lloyd 2007: 97).

In meinen Erhebungen wurden jedoch *zwei* Ebenen von Vertrauen deutlich, nämlich das Vertrauen in das *Web of Trust* selbst und das Vertrauen in Personen. Niklas Luhmanns Begrifflichkeiten spiegeln diesen Unterschied treffend wider: Das *Web of Trust* kann mit ihnen als Beispiel für Systemvertrauen analysiert werden, das er von persönlichem Vertrauen unterscheidet. Vertrauen sei ein komplexitätsreduzierender „Mechanismus“, der es Menschen erlaube, die Welt selektiv zu interpretieren (vgl. Luhmann 1973: 33). Seine Unterscheidung verschiedener Arten von Vertrauen ist im Rahmen dieser Arbeit interessant für die Analyse von Beziehungen, die sich sonst viel im Internet abspielen: Luhmann geht davon aus, dass in sozialen Zusammenhängen, die sich durch Dauerhaftigkeit, gegenseitige Abhängigkeit und eine gewisse Unvorhersehbarkeit auszeichnen, Vertrauen gut gedeihen kann. „Es herrscht das Gesetz des Wiedersehens. Die Beteiligten müssen einander immer wieder in die Augen blicken können. Das erschwert Vertrauensbrüche“ (ebd. 39). Wenn jedoch Kommunikationsmedien dazwischen geschaltet werden, kämen Unwägbarkeiten hinzu, weil die Grundlage der Beziehung nicht die persönliche Bekanntschaft ist (vgl. ebd. 52). Welche Art von Vertrauen in einem Interaktionszusammenhang aufgebaut wird, hänge u.a. davon ab, welche Rolle Individuen in dem Gefüge spielten, ob die Sozialordnung z. B. mit bestimmten Personen steht und fällt. Systemvertrauen etabliere sich in Zusammenhängen, in denen Individuen nicht Garanten des Gefüges seien (vgl. ebd. 50). Was Luhmann über Vertrauen in den Geldwert (als Systemvertrauen) schreibt, lässt sich in ähnlicher Form über das *Web of Trust* sagen: Wer in das *Web of Trust* vertraut, darauf vertraut, dass es reale Personen und digitale Keys zuverlässig miteinander in Verbindung setzt, glaubt an das Funktionieren eines Systems, des Keysignings. Die Signierenden schenken demnach weniger den Personen Vertrauen, die ihnen gegenüber stehen, sondern sie vertrauen vielmehr in die Aussagekraft und Vertrauenswürdigkeit des *Web of Trust* selbst (vgl. ebd. 54).

Keysigning-Parties sollen gleichzeitig gerade die Orte sein, wo Communities sich treffen, sich bei der Passkontrolle in die Augen sehen, um Vertrauensbrüche zu erschweren. Das Internet soll den Individuen eben *nicht* ihre Relevanz im Interaktionsgefüge der Community nehmen: Das Interview mit dem Organisator A61 zeigt, dass zumindest diejenigen, die großen Wert auf Keysigning legen, die Ernsthaftigkeit, Genauigkeit und Über-

zeugung ihrer Gegenüber wahrnehmen. Das Verhalten während einer Keysigning-Party ist damit immer auch eine Selbstdarstellung der Teilnehmenden in Bezug auf ihre Vertrauenswürdigkeit (vgl. Luhmann 1973: 41). Dort, wo Schlüssel der Authentifizierung dienen, um Zugriff auf konkrete, geschützte Daten zu erhalten, spielt persönliches Vertrauen im Luhmannschen Sinne eine ebenso wichtige Rolle wie das Vertrauen in das *Web of Trust* selbst. Es beinhaltet die Erwartung, dass die Signing-Partner_innen bei dem bleiben, was sie beim Keysigning über sich, über ihre Genauigkeit, Ernsthaftigkeit, kurz: über ihre Vertrauenswürdigkeit mitgeteilt haben (vgl. Luhmann 1973: 40 f.). Wenn beim Keysigning persönliches Vertrauen geschenkt wird, wird erwartet, „daß der andere seine Freiheit, das unheimliche Potenzial seiner Handlungsmöglichkeiten, im Sinne seiner Persönlichkeit handhaben wird - oder genauer, im Sinne der Persönlichkeit, die er als die seine dargestellt und sozial sichtbar gemacht hat“ (ebd.).

Keysigning-Parties können zweitens mit Pierre Bourdieu als „Ökonomie des symbolischen Tauschs“ gelesen werden (vgl. zum Prinzip von Gabe und Gegengabe Mauss 1994). Getauscht werden Signaturen, ein symbolisches Kapital, das Vertrauenswürdigkeit, also soziales Kapital, abbildet. Das persönliche Zusammentreffen ist eine der Bedingungen für den Kapitaltransfer: Keysigning-Parties haben auch den Charakter eines Treffens, bei dem sich Menschen kennen lernen können oder bei dem sich Menschen, die bisher nur über das Internet kommuniziert haben, sich persönlich begegnen. Sie verlassen das Internet und zeigen sich „offline“. Persönliches Vertrauen können sie schon dabei aufbauen (vgl. Lin 2004b: 196 f.), es gilt jedoch *auch*, das *Web of Trust* zu stärken und somit die Basis für Systemvertrauen auszubauen. Der Aspekt des persönlichen Kennenlernens von Menschen, die sich aus dem Internet bereits kennen, wird auch von Siobhán O’Mahony und Fabrizio Ferraro im Zusammenhang mit Keysigning betont. Sie untersuchen, aufgrund welcher Aktivitäten *Debian*-Entwickler_innen zu Projektleiter_innen gewählt werden. Dabei gehen sie auf die Relevanz technischer Beiträge zum Projekt ein, aber auch auf Aktivitäten im Bereich „organization building“, nämlich die Beteiligung an Diskussionen auf Mailinglisten und die Teilnahme an face-to-face-Begegnungen. Diese Begegnungen messen sie anhand der Teilnahme an Keysigning-Parties (vgl. O’Mahony/Ferraro 2007: 1092 ff.).³⁶ Die Studie beinhaltet qualitative Interviews mit *Debian*-Mitgliedern und verschiedene statistische Erhebungen. Das in diesem Zusammenhang interessante Ergebnis von O’Mahony/Ferraro ist, dass persönliche Begegnungen mit anderen im Projekt Engagier-

³⁶Sie verwenden die Informationen auf Keyservern für ihre Berechnungen. Dort sind auch die Daten der Signaturen erfasst, so dass die vorangegangenen persönlichen Begegnungen sogar zeitlich grob festgemacht werden können.

ten die Wahrscheinlichkeit enorm ansteigen lassen, dass jemand in eine führende Position gewählt wird: „[W]e found that meeting ten more developers in the community increased the likelihood of becoming a member of the leadership team by 56 percent“ (ebd. 1098). Ihre Erklärungsansätze dafür sind vertrauensvollere Beziehungen zueinander nach einem Treffen, die Möglichkeit, beim Zusammentreffen Koalitionen zu bilden und Stimmen einzuwerben, und die Hoffnung, von der Wahl eines persönlichen Bekannten selbst in der Zukunft des Projektes zu profitieren. Bei persönlichen Zusammentreffen von Engagierten wird soziales Kapital innerhalb der Communities aufgebaut. (Ein Beispiel war der Hinweis auf ein kommendes Wiedersehen bei einem Vortrag, „ich organisiere da mit.“) Die Rangliste der am besten vernetzten Keys bildet dieses soziale Kapital ab: Sie lässt einerseits Schlüsse darauf zu, auf wie vielen Community-Events jemand war. Andererseits drückt die oben zitierte E-Mail aus, dass nicht jede Signatur gleich viel „wert“ ist, um in der Liste aufzusteigen: Die Signatur eines bereits gut vernetzten Keys verkürzt den Weg zum eigenen Key drastisch.

Der Kapitaltausch unterliegt einer zeitlichen Zergliederung (vgl. Bourdieu 1998b: 163 ff.; siehe zum Kontext von Vertrauen bei Online-Auktionen Brinkmann/Seifert 2001: 32 f.). Über das Intervall zwischen der Keysigning-Party und dem Zusenden eines signierten Schlüssels gibt es keine Einigkeit, auch wenn manche Organisator_innen der Zusammenkünfte Stichtage nennen, bis zu denen das Signieren abgeschlossen sein sollte. Es bleibt den Teilnehmenden überlassen, wie viel Zeit sie verstreichen lassen. Die Reaktionen auf A10s Praxis des Nichtsignierens zeigen, dass es eine unabgesprochene Grenze gibt, wie viel Zeit verstrichen sein muss, um davon auszugehen, dass jemand keine signierten Schlüssel mehr zurücksenden wird, dass die Gabe einseitig ist. Abweichendes Verhalten ist hier in einem gewissen Rahmen hinnehmbar, die zeitliche Schwelle, ab der jemandem das Vertrauen entzogen wird, ist jedoch — anders als in Luhmanns Vorstellung — *nicht* klar definiert (vgl. Luhmann 1973: 31). Im Gegensatz zur Ausweissichtung in einer Behörde hängt die Situation nicht nur solange in der Schwebe, bis die Amtsperson den Pass gelesen und akzeptiert hat (vgl. Scheffer 1997: 172), sondern weit darüber hinaus: Wird gegenseitig signiert werden oder nicht? Wann ist klar, dass nichts mehr zurückkommt? Der Austausch von Signaturen als symbolischem Kapital und als Ausdruck von Verbindlichkeit bleibt damit ein störanfälliger und subjektiver Vorgang.

Schließlich wende ich mich den Praktiken der Differenzierung und Distinktion beim Keysigning zu, die auf zwei verschiedenen Ebenen stattfindet: Erstens werden Unterschiede entlang der auf dem amtlichen Dokument erfassten Kategorien (re-)produziert. Dies sind

Differenzkategorien, die mit gesellschaftlichen Herrschaftsverhältnissen verknüpft sind. Zweitens findet eine „ritualspezifische“ Distinktion entlang der Kategorie der Vertrauenswürdigkeit statt.

Die amtlichen Dokumente stehen als Artefakte im Mittelpunkt des Rituals: Sie sind eine Teilnahmebedingung der Veranstaltung. Keysigning-Parties basieren auf dem Konsens, dass die Identitäten der Teilnehmer_innen nicht anders bestimmt werden können als mit Ausweisen, ein Beispiel für die Formalisierung des Rituals. Wie in einer Behörde erlauben nur diese Dokumente die eindeutige Zuordnung von Keys zu menschlichen Körpern und belegen die Integrität der Person. Keysigner_innen machen sich den administrativen Blick einer Amtsperson zueigen, sie stellen die anderen Internet-Nutzer_innen über die Passkontrolle als leibhaftige und damit vertrauenswürdige Personen her (vgl. Scheffer 1997: 186). „Der Paßinhaber kann erwarten, daß ihm angesichts der Belege — und nur mit diesen — geglaubt wird“ (ebd. 191). Bereits aufgebautes persönliches Vertrauen genügt nicht für eine Signatur, die Ausdruck des Systemvertrauens im *Web of Trust* ist. Obwohl im Umgang mit abgelaufenen Dokumenten verschiedene Umgangsformen existieren, wird immer wieder betont, dass „gültige amtliche Dokumente“ vorzuzeigen sind. Vertrauenswürdige Kommunikationspartner_innen sind damit immer auch eindeutig identifizierbare Staatsbürger_innen, im Idealfall gesetzestreue Bürger_innen, die abgelaufene Dokumente gleich ersetzen. Es ist sehr verbreitet, dass Community-Mitglieder unter Spitznamen auftreten — ihre Keys müssen sie jedoch ihren bürgerlichen Namen zuordnen. Spitznamen können höchstens als Zusatzinformation beigefügt werden, wenn der Schlüssel signiert werden soll. Dass Führerscheine in anderen Ländern als „amtliche Dokumente“ gelten, wird nicht von allen akzeptiert. Hier zeigen sich Vorstellungen von Normalität, die durch die starke Formalisierung des Keysignings — wie in der Behörde — andere Varianten des Einsatzes von Dokumenten ausschließt (vgl. ebd. 189). Für diese in Deutschland beobachteten Keysigning-Parties muss also hinzugefügt werden, dass *deutsche* Ausweispapiere die Norm darstellen, während abweichende ausländische Papiere in Einzelfällen als Identitätsnachweise angezweifelt oder abgelehnt werden. Die Wahrscheinlichkeit, dass Personen, die keine Papiere besitzen, zu Keysigning-Parties erscheinen würden, ist ohnehin sehr gering, sie wären jedoch durch das formalisierte Prozedere ausgeschlossen. Die Beobachtungen zeigen zudem, dass ohne Absprache deutsch als Verkehrssprache der Keysigning-Parties gesprochen wird, obwohl die jeweiligen Linuxtage internationales Publikum anziehen und dort zahlreiche Vorträge auf englisch gehalten werden. Die Keysigner_innen, die von der Norm, deutsch zu können, abweichen, müssen

sich selbst zu Wort melden und ein Abhalten des Keysigning auf Englisch einfordern.

Die Ausweispapiere sind zudem der häufigste Inhalt des Small Talk bei Keysigning-Parties: Manche machen Witze über die neuen deutschen Personalausweise und äußern damit eine implizite politische Kritik an der Erfassung biometrischer Daten für die Dokumente. Die Mehrheit der Kurzgespräche bezieht sich jedoch auf die persönlichen Informationen auf dem amtlichen Dokument: Alter, Aussehen (Brillen, Frisuren), Geschlecht, Nationalität und Herkunft. Viele richten ihre Aufmerksamkeit auf ihnen unbekannte Namen oder Dokumente, auf Sprachen und Dialekte. Auf einer Party wurden dieselben Personen immer wieder auf ihre Namen angesprochen, andere versuchten, die Aussprache nachzuahmen oder kommentierten die Namen.

Diese Praktiken stellen Anderssein her und stärken Normen: Die Überprüfung staatsbürgerlicher Identitäten beim Keysigning in Deutschland erfolgt vom privilegierten Standpunkt einer weißen, deutschen Mehrheitsgesellschaft aus. Dieser Standpunkt wird selbst nicht benannt und ist den Beobachteten möglicherweise nicht bewusst (vgl. z. B. Frankenberg 1996; Rommelspacher 1995; McIntosh 2003; Wachendorfer 2001; Wollrad 2005). Von diesem weißen/deutschen/männlichen Standpunkt betrachten Teilnehmende der Keysigning-Party andere Teilnehmende. Sie fordern sie zur Erklärung oder Aussprache ihres Namens auf, bezeichnen „exotische“ Namen als „cool“, wundern sich merklich über mehr als eine Staatsangehörigkeit. Sie unterhalten sich laut hörbar über die Anwesenheit von Frauen, identifizieren deutsche Staatsbürger_innen als Bosnier_innen und setzen selbstverständlich voraus, dass Niederländer_innen deutsch sprechen. Mit diesen Äußerungen über Personen sind Privilegien verbunden, etwa, dass diejenigen, die „der Norm“ entsprechen, bei der Keysigning-Party genau solche Bemerkungen oder Fragen nicht zu hören bekommen. Sie können als dort als Aktivist_innen bestimmter Software- oder Communityprojekte auftreten und die Aufmerksamkeit, die sie bekommen, allein auf ihr Engagement in der Szene, auf ihre individuelle Leistung zurückführen. Ihre amtlichen Dokumente enthalten keine Informationen, die andere Formen von Aufmerksamkeit auf sich ziehen. Wer privilegiert ist, kann es als Witz verbuchen, auf das eigene Passfoto angesprochen zu werden, weil er_sie mit dem eigenen Aussehen im sonstigen Leben vermutlich selten konfrontiert wird. Demgegenüber ist es wahrscheinlicher, dass diejenigen, die hier als Abweichungen von einer Norm identifiziert werden, dies auch in anderen Kontexten erleben, sei es innerhalb der Szene oder der gesamten Gesellschaft. Die Produktion von Anderssein bei Keysigning-Parties hat etwas mit der prominenten Rolle von Ausweispapieren zu tun: Sie enthalten ganz bestimmte Informationen über Personen, die sonst in

der Szene keine so wichtige Rolle spielen. Die Organisationsform des Rituals befördert jedoch das Klassifizieren von Personen nach den Eigenschaften, die auf dem Dokument benannt sind — und das sind *auch* Diskriminierungsachsen. Im Kontext von Keysigning-Parties ist es nicht einfach, zwischen freundlich gemeinten Witzen und dem Bezug auf diskriminierende Muster zu unterscheiden, die tief in den gesellschaftlichen Mainstream eingebettet sind und oft unbewusst aufgerufen werden. Diese Form der Distinktion hat nichts mit der digitalen Vertrauenswürdigkeit der Personen oder ihrer Keys zu tun, sondern sie wirkt dadurch, *dass* bestimmten Personen immer wieder gesagt wird, wodurch sie sich von einer unausgesprochenen Norm unterscheiden. Die Differenzen, die beim Keysigning durch Passkontrolle und Small Talk produziert werden, sind keine absichtlichen diskriminierenden Äußerungen, sondern Ausdruck einer Dominanzkultur in der gesellschaftliche Normen und Hierarchien unreflektiert reproduziert und gestärkt werden, wobei gleichzeitig die Vorstellung von einer egalitären Gesellschaft aufrechterhalten wird (vgl. Rommelspacher 1995: 24 ff.).

Der Rekurs auf vom Staat ausgestellte Dokumente beim Keysigning erscheint widersprüchlich, vor dem Hintergrund, dass staatliche Behörden zu den potenziellen Überwachern von Online-Kommunikation gehören. Bei allem artikulierten Misstrauen nimmt sich die Ausweiskontrolle als Praxis des „Doing State“ aus und als diskursive Reproduktion von Normen der hier tendenziell weißen, männlichen, deutschen Mehrheit. Gleichzeitig *kann* Keysigning auch als eine Loslösung von staatlich ausgestellten Identitätsnachweisen betrachtet werden: Wenn die Rückkopplung an das Dokument geschehen ist, fungiert der signierte Key als virtuelles Ausweisdokument, das nur noch mit einem bürgerlichen Namen und einer E-Mailadresse verbunden ist. *Nach* dem Keysigning ist die Verwendung von Keys zur Authentifikation im Internet ein „Undoing State“, in dem Sinne, dass die Szene sich auf ihr eigenes Verfahren verlässt.

Die zweite Ebene der Distinktion findet innerhalb der Logiken des Keysigning statt. Besonders diejenigen, die das *Web of Trust* als persönliches Vertrauensnetzwerk sehen, grenzen sich von achtlosen Internet-Nutzer_innen im Allgemeinen, aber auch von Teilen der Szene ab: Nur ein Teil der Szene betreibt überhaupt Keysigning, hat sich bereits damit auseinandersetzt oder hält es für relevant. Unter den Keysigner_innen sind zudem diejenigen, die das Ritual nicht ernst genug nehmen oder die den Spaß am Ringen um hohe Listenplätze in den Vordergrund stellen, die also symbolisches Kapital in Form von Signaturen akkumulieren, das ohne Praxisrelevanz ist und eher an einen Fetisch erinnert. Beim Keysigning konstituiert sich ein bestimmter Teil von FLOSS-Communities, und

zwar diejenigen, die öffentlich demonstrieren möchten, wie wichtig ihnen Themen wie verschlüsselte Kommunikation, eindeutige Zurechenbarkeit von Daten zu Personen und ein bewusster und sensibler Umgang mit elektronischen Daten sind. Sie nehmen das Ritual als Akt wahr, in dem die Teilnehmenden etwas über ihre Zuverlässigkeit, Genauigkeit und Ernsthaftigkeit aussagen. Wer in diesem Wahrnehmungsschema Systemvertrauen *und* persönliches Vertrauen erweckt, gilt als vertrauenswürdige_r Kommunikations- und möglicherweise Projektpartner_in.

Das *Web of Trust* ist eine objektivierte Darstellung von Beziehungen innerhalb von FLOSS-Communities; es zeigt, welche Personen sich persönlich begegnet sind und sich voreinander mit ihren staatsbürgerlichen Identitäten in Verbindung gebracht haben. Diese Beziehungen werden beim Keysigning hergestellt und immer wieder aktualisiert („Wir haben uns schon zigfach signiert“ (Int 2008b)). Die Parties sind Orte der Institutionalisierung von Vertrauen, der Akkumulation und des Tauschs von sozialem und symbolischem Kapital (vgl. Bourdieu 1983: 191). Von sozialem Kapital lässt sich dort sprechen, wo Kooperationsnetzwerke *zusätzlich* zu einer virtuellen auf einer physischen Ebene ausgebaut werden (vgl. O'Mahony/Ferraro 2007: 1098 f.). Das objektivierte Vertrauen, das in Schlüsseldateien digital eingeschrieben wird, auf Keyservern veröffentlicht und in Ranglistenplätzen abgebildet wird, bildet ein symbolisches Kapital. Dies kann, wie O'Mahony / Ferraro zeigen, in spezifischen Umfeldern direkt in soziales Kapital umgewandelt werden.

Keysigning-Parties stellen eine ordnende Praxis auf Community-Events dar: Sie verschaffen den Teilnehmenden in verschiedener Hinsicht einen Überblick über die Anwesenden, die sich gemäß einer bestimmten Regel körperlich optimal anordnen, um einander wahrzunehmen. Szenegänger_innen sehen sich (wieder), sie können bürgerliche Namen, Gesichter und Projekte einander zuordnen, die sie unter Umständen aus dem Internet bereits kennen. Die kurzen Gespräche während der Identitätskontrolle zeigen, welche Relevanz vor allem Nationalität und Herkunft der Teilnehmer_innen zugesprochen wird. Auf Keysigning-Parties scheint die Leute zu interessieren, wer zu den Communities dazugehört. Zugehörigkeit entsteht auf den zwei erwähnten Ebenen, nämlich auf der Ebene des Rituals selbst und auf der Ebene gesamtgesellschaftlicher Herrschaftsverhältnisse, die mehr oder weniger subtil den Small Talk durchziehen. Die Ordnungsliebe, die in dem Ritual zum Ausdruck kommt, ordnet ebenfalls die Szene nach Geschlecht, Wissen, Herkunft, Staatsbürgerschaft und Sprache.

7.3. Empirische Zwischenergebnisse

In diesem Kapitel wurden zwei Themenkomplexe untersucht, die sich bei der teilnehmenden Beobachtung in Linux-Communities als sehr präsent gezeigt haben. Im ersten Abschnitt ging es um verschiedene Praktiken, die mit der technischen Absicherung von Computern zu tun haben. Passwörter und Zugriffsrechte unter Linux sind Beispiele für grundlegende Ordnungen, die ins Betriebssystem eingebaut sind. Sie werden in der Praxis von Systemadministrator_innen und -nutzer_innen ständig reproduziert, sind jedoch unhinterfragte Grundlagen von Linux, die Praktiken technisch ermöglichen oder verhindern, Zugänge gestatten oder verwehren. Über den konkreten Umgang mit ihnen sind — wie gesehen — soziale Aushandlungen innerhalb von Gruppen nötig.

Wie bereits in den vorhergehenden Kapiteln sind auch hier sehr unterschiedliche Praktiken zu beobachten. Ihre Bandbreite findet sich bereits innerhalb einer LUG, sie kommt aber durch die Konfrontation mit Beobachtungen aus dem professionellen IT-Umfeld noch klarer zum Vorschein: Insgesamt wird das Spektrum der Praktiken charakterisiert durch ein Spannungsfeld von theoretischem Wissen, technischen Möglichkeiten und praktischer Handhabung: So zeigt der Umgang mit Passwörtern gerade in der LUG, also außerhalb eines professionellen und dementsprechend verregelten Rahmens, eine Diskrepanz zwischen theoretischem Wissen über Passwortschutz und großzügigen Umgangsweisen mit Passwörtern. Die LUG stellt sich diesbezüglich als Raum des Ausprobierens und Lernens für technische Themengebiete dar.

Der Umgang mit Zugriffsrechten zeigt eine Diskrepanz zwischen technischen Konfigurationsmöglichkeiten und der sozialen Aushandlung ihrer Nutzung: Die technischen Möglichkeiten sind *sehr* ausgefeilt. Wie detailliert Zugriffsrechte in einem konkreten Zusammenhang implementiert werden, ist ein komplexer und aufwändiger Aushandlungsprozess. In dem beobachteten mittelständischen Unternehmen wurde an der nie getroffenen Entscheidung deutlich, wie unterschiedlich der Sinn für die Relevanz von Zugriffsrechten zwischen Abteilungen mit verschiedenen Arbeitslogiken verteilt ist. Im Rahmen meiner Beobachtungen stellen widersprüchliche Praktiken somit die Grenzen der technischen Absicherungsmöglichkeiten dar. Wie gezeigt, haben die Praktiken nicht unbedingt etwas mit dem Wissensstand der Beteiligten zu tun, die nämlich oft wider besseren Wissens handeln, wenn sie Passwörter nicht sicher aufbewahren oder sich nicht um klare Rechterege-lungen bemühen.

Technische Sicherheitsmaßnahmen (wie Passwörter, Zugriffsrechte, Firewalls) wirken auf verschiedene Weise differenzierend: Viele machen auf technische Unsicherheiten oder Verstöße gegen Konventionen aufmerksam oder legen durch ihre eigenen Praktiken des demonstrativen Nichtwissenwollens fremder Passwörter eine starke Betonung auf technische Sicherheit. Sie konstituieren sich damit demonstrativ als diejenigen, die Passwörter als sensibel erkennen und die privaten Daten anderer respektieren. Wenn dies auch gerade in der LUG nicht unbedingt als Belehrung gegenüber denen wirkt, die ihre Passwörter etwa laut aussprechen, so kann es doch als eine Aufführung von „Best Practice“ interpretiert werden.

Was die Verwaltung von Zugriffsrechten angeht, ist das soziale *und* technische Wissen über angemessene Regelungen zentral. In der alltäglichen Praxis von Systemadministrator_innen sind Zugriffsrechte ein wichtiger Bestandteil von situativer praktischer Problemlösungskompetenz. Zudem schreiben Systemadministrator_innen soziale oder organisatorische Hierarchien in Form von Softwarekonfigurationen fort. Diese werden ihnen von Firmenpolitiken, rechtlichen Rahmenbedingungen und von ihrer eigenen fachlichen Verantwortung aufgegeben. Sie arbeiten zur Absicherung von Computersystemen mit klaren Klassifizierungen: Ihre Logik der Praxis beinhaltet, dass die Hierarchien von Accounts auf den Rechnern klar festgelegt werden, dass User_innen Gruppen mit bestimmten Zugriffsrechten zugeordnet werden. Fein konfigurierbare Zugriffsrechte und Firewalls zeigen, wie viele Restriktionen implementiert werden, damit — rein technisch — niemand die Möglichkeit hat, ohne explizite Absprachen auszuscheren. Eine arbeitsteilige soziale Organisation von Rechten, Pflichten, Zuständigkeiten und Verantwortung wird durch die technologische Differenzierung reproduziert. Technische Sicherheit ist mit Unterscheidungen eng verknüpft: Sie wird über Ausschlüsse in Form von ungleich verteilten Zugriffsrechten und andere Praktiken der Systemadministration, etwa Firewalls, hergestellt. Der diesbezügliche Konsens ist, dass die Sicherheit eines Computersystems desto besser gewährleistet ist, je weniger Personen weitreichende Rechte darauf haben. Systemadministrator_innen pflegen einen autoritären und von Misstrauen geprägten Umgang mit Zugriffsrechten. Das liegt daran, dass sie für die Server verantwortlich sind, auf denen unwissende oder böswillige Nutzer_innen Schaden anrichten könnten. Ihr professioneller „Sinn für das Spiel“ verbietet ihnen einen vertrauensvollen Umgang.

Die Zugänge zu Rechnern und die Art des Zugangs bilden in diesem Zusammenhang das spezifische Kapital, um das es geht — auch wenn in den beobachteten Kontexten kein Kampf um die Kapitalverteilung stattfindet, sondern sie als gegeben hingenommen wird.

Diese Zugänge kann man innerhalb der Communities bekommen, wenn man — je nach Situation — über soziales, symbolisches und/oder kulturelles Kapital verfügt: Zugriffe auf Server einer Community bekommen diejenigen, die über ihr Engagement eine gewisse Bekanntheit innerhalb dieser Community haben, die gezeigt haben, dass sie „wissen, was sie tun“, und die sich (falls das Projekt es verlangt) beim Keysigning zu einer bürgerlichen Identität bekannt haben, mit der ihre Aktivitäten verknüpfbar sind. Root- oder User-Accounts kann man auch auf privaten Servern von Bekannten oder Freund_innen bekommen, wobei persönliches Vertrauen das ausschlaggebende Vergabekriterium ist. Als Kapital bezeichne ich diese Zugänge in Form von Accounts und Rechten, weil es sich um exklusive materielle Ressourcen handelt: Ein Serveraccount kann Speicherplatz oder Konfigurationsvorlagen bieten. Er kann die Möglichkeit geben, Programme von einem anderen Ort im Internet auszuführen, also auch nationale Gesetzgebungen zu umgehen. Er ermöglicht es, eine selbstbestimmte Infrastruktur mit Datenhoheit aufzubauen, für Einzelne, kleine Gruppen oder für die Öffentlichkeit. Kulturelles Kapital steht hier mit auf dem Spiel: Zugänge mit weitreichenden Rechten gehen eher an diejenigen, die glaubhaft machen können, dass sie „wissen, was sie tun“, dass sie mit den Daten, auf die sie Zugriff erhalten umzugehen wissen.

Im zweiten Teil des Kapitels wurden Keysigning-Parties als wiederkehrendes Ritual auf Community-Events untersucht. Bei Keysigning-Parties geht es — ähnlich wie bei genau konfigurierten Zugriffsrechten — darum, Zurechenbarkeit zu schaffen. In diesem Fall werden Online- und Offline-Identitäten zu integren Persönlichkeiten zusammengeführt. Mit dem *Web of Trust* haben die Communities ein formalisiertes System dafür etabliert. Die soziale Distinktion beim Keysigning wirkt, wie gesehen, auf zwei verschiedenen Ebenen: Einerseits ist Keysigning ein konstitutives Ritual für diejenigen, die sich öffentlich als vertrauenswürdig zeigen möchten. Sie stellen durch ihre Anwesenheit eine Verbindung zwischen ihrer Online-Identität und ihrer Person her und bekennen sich zu einem bewussten Umgang mit digitalen Daten. Darin steht die Abgrenzung von denen, die sich ausschließlich anonym im Netz bewegen wollen, und von denen, die die Sensibilität vor allem fremder Daten nicht ernst nehmen. Andererseits steht hier statt dem Computer ein anderes Artefakt im Mittelpunkt, nämlich das amtliche Ausweispapier, das den einzig möglichen Identitätsnachweis darstellt. Die Informationen, die der Ausweis enthält, regen die Teilnehmer_innen zu Gesprächen an, die sich entlang von Kategorien strukturieren, die auch Diskriminierungskategorien sein können: Herkunft, Alter, Geschlecht, Staatsangehörigkeit, Sprache und Aussehen werden hier diskursiv und oft stereotyp reproduziert.

7.4. Theoretische Reflexion: Distinktion über formalisierendes Denken

In beiden Teilen dieses Kapitels wurden unterschiedliche Praktiken untersucht, in denen — so die vorgeschlagene Lesart — durch Formalisierung Strukturen geschaffen oder reproduziert werden. Zum einen sind formale Strukturen in technische Artefakte eingelassen, zum anderen ist ein sozialer Akt nach einem formalen Schema aufgebaut. Beide Arten von formalisierenden Praktiken sollen abschließend als solche theoretisch reflektiert werden. Dabei geht es mir darum, Differenzierung und Ausschluss über grundsätzliche Praktiken des Formalisierens zu benennen.

Passwörter und Zugriffsrechte sollen auf Computern Klarheiten schaffen: Soziale Ordnungen können über diese Mechanismen in Softwarekonfigurationen abgebildet werden. Es gibt weitere solcher Mechanismen, Passwörter und Zugriffsrechte gehören jedoch zu den Linux-Grundlagen, deren Sinn niemand hinterfragen würde: Es muss klar sein, wer sich an einem Rechner anmelden kann, wer was lesen oder verändern darf. Dennoch zeigt der praktische Umgang mit beiden Schutzmechanismen, dass sie in der Praxis oft einen Computer nicht vor unbefugtem Zugriff schützen würden. Die Maschinen halten mehr Präzision und Konfigurationsmöglichkeiten vor, als die beobachteten Menschen in ihrer Praxis verwenden.

Das im zweiten Teil des Kapitels untersuchte *Web of Trust* stellt eine Formalisierung von Beziehungen dar, die die Communities sich selbst gegeben haben, um eine Struktur zur Identifikation von Personen aufzubauen. Keysigning-Parties sind formalisiert, weil sie einem Ablaufplan folgen, auch wenn die Organisator_innen oder Teilnehmenden ihn teilweise situativ abwandeln. Die Teilnehmenden werden im Raum angeordnet, „gemustert“ (A61), nach Kriterien klassifiziert, die mit dem *Web of Trust* zu tun haben oder nicht. Die entstandenen Beziehungen werden auf einer Skala von Vertrauenswürdigkeit und in einem Netz abgebildet, das statistische Berechnungen ermöglicht. Eine stark abstrahierende und formalisierende Denkweise aus der Informatik setzt sich hier in der sozialen Organisation der Szene fort. Die Analyse hat auch die Grenzen der Formalisierbarkeit von Vertrauen im *Web of Trust* aufgezeigt: Der „ideale“ Ablauf des Keysigning steht im Gegensatz zu tatsächlichen Praktiken der Teilnehmenden, die teilweise weit auseinandergehen, was die Genauigkeit oder die Gegenseitigkeit angeht. Vor dem Hintergrund dieser verschiedenen Umgänge mit der Identitätszuordnung muss *innerhalb* der Logik des Keysigning

hinterfragt werden, ob sie eigentlich „funktioniert“, in dem Sinne, dass sie soziale Realität angemessen abbildet.

Die beobachteten Praktiken stehen damit in vielen konkreten Situationen im Widerspruch zu den Logiken, die das Betriebssystem bzw. das Ablaufschema Keysigning verlangen. Zur Erläuterung dieser Widersprüche lohnt sich ein Blick in die Literatur, die Formalisierung grundsätzlich und in der Informatik problematisiert.³⁷

„Die Informatik lässt sich als eine Disziplin verstehen, die sich primär mit Abstraktion, Klassifikation und Formalisierung befasst. Statische und dynamische Strukturen werden zunächst identifiziert und explizit beschrieben sowie modelliert, um Prozesse durch Software unterstützen oder sie automatisieren zu können“ (Bath 2009: 165 f.).

Wer Software entwickle, habe es stark eingeübt, wiederkehrende Strukturen und Ähnlichkeiten zu sehen und formal zu beschreiben. Dabei würden Besonderheiten, die nicht in die betrachteten Strukturen passten, leicht übersehen (vgl. Maaß/Rommes 2007: 97). Anliegen aus der realen Welt würden durch Formalisierung stark vereindeutigt. Dies schließe Mehrdeutigkeiten, Kontingenzen und die Situiertheit von realweltlichem Wissen aus (vgl. Bath 2009: 166, unter Bezug auf Crutzen 2007: 40). Diese Wissensrepräsentation, die auf die Aufklärung zurückgeht und auch die westlichen Wissenschaften prägt, ist aus feministischer Perspektive grundsätzlich hinterfragt worden: Alles Irrationale, Emotionale oder Chaotische werde nicht nur in Formalisierungsprozessen nicht abgebildet, sondern es sei auch nicht formalisierbar (vgl. ebd. 165). Damit komme eine weitere Dimension der Vergeschlechtlichung in technische Artefakte: Das von der Formalisierung Ausgeschlossene seien nämlich meistens die Teile von symbolischen Dichotomien, die weiblich konnotiert seien. Die symbolisch männlich konnotierten Attribute seien die formalisierbaren (vgl. Willis 1997). Technologien schrieben in Praktiken des Formalisierens Illusionen von Neutralität und Objektivität ein, die jedoch implizit ein „Anderes“ mitdefinieren. Dadurch werde die Geschlechterordnung auf einer symbolischen Ebene perpetuiert (vgl. Bath 2009: 192; siehe auch Klinger 1995). Bath resümiert drei kritische Perspektiven auf das Formale bzw. auf Formalisierung:

1. In Formalisierungsprozessen werde *eine* angeblich neutrale Sicht als *die* objektive verbreitet. Dadurch würden spezifische Kontexte und soziale Ungleichheiten ignoriert und somit Ausschlüsse produziert. Die hinter der Formalisierung stehenden,

³⁷Ich stütze mich im Folgenden vor allem auf die ausführliche Literaturlaufarbeitung von Corinna Bath (2009: 165 ff.).

immer auch politischen Entscheidungen über das, was einbezogen wird, würden verdeckt.

2. Die Entwickler_innen, die Subjekte der Formalisierung, erschienen als Effekt des Formalen nicht. Da sie ihre Perspektiven und Relevanzen durch bestimmte Entscheidungen in den Formalisierungsprozess einschrieben, seien sie und ihre Annahmen offenzulegen.
3. Noch grundsätzlicher sei der Dualismus „formal“ versus „nichtformal“ eine der zahlreichen Dichotomien, die symbolische Hierarchien fortschreiben (vgl. Bath 2009: 171; siehe ausführlicher ebd. 165-213).

Formalisierung kann demnach als Möglichkeitsbedingung von Ausschlüssen verstanden und kritisiert werden: In der Informatik ist sie eine Voraussetzung dafür, dass Herrschaftsstrukturen in Technologien eingeschrieben werden. Sie ist ein Prozess, in dem das Politische durch vermeintliche Objektivität negiert wird (vgl. ebd. 178). Dies wirft folgende Fragen an die oben beschriebenen formalisierenden Praktiken auf: Welche politischen Entscheidungen haben zur Etablierung welcher vermeintlich objektiven Sicht geführt? Wer hat sie getroffen? Welche Annahmen stehen hinter ihnen? Was wird ausgeschlossen und fällt damit der symbolischen Hierarchie formal vs. nichtformal zum Opfer?

Die Entscheidungen in der Geschichte der hier untersuchten Formalisierungspraktiken nachzuzeichnen, ist nicht einfach. Zugriffsrechte wurden implementiert, als die Daten mehrerer Benutzer_innen auf denselben Maschinen gespeichert wurden. Dies war seit dem so genannten Time-Sharing der Fall, als sich verschiedene User_innen die Rechenzeit von Großrechnern teilten. Das Prinzip des Time-Sharing wurde 1957 von Bob Bemer beschrieben und im selben Jahr von John McCarthy umgesetzt.³⁸ Unix wurde von vorn herein, also ab 1969 als Mehrbenutzerbetriebssystem in den *Bell Laboratories* von *AT & T* entwickelt. Dateiberechtigungen gab es seit der ersten Version, die Aufteilung in Eigentümer_innen, Gruppen und Andere besteht seit UNIX-V4 1974.³⁹ Hinter der Implementierung von Zugriffsrechten und dem Schutz digitaler Daten durch Passwörter und Verschlüsselung steht ein Grundsatz, der auch in der Hackerethik formuliert ist: „Öffentliche Daten nützen, private Daten schützen.“⁴⁰ Über private Daten sollen nur die Eigentümer_innen bestimmen können. Als öffentliche Daten gelten in der in FLOSS-Communi-

³⁸vgl. [http://de.wikipedia.org/wiki/Time-Sharing_\(Informatik\)](http://de.wikipedia.org/wiki/Time-Sharing_(Informatik))

³⁹<http://de.wikipedia.org/wiki/Unix-Dateirechte>, <http://de.wikipedia.org/wiki/Dateiberechtigung>.

⁴⁰<http://www.ccc.de/hackerethics>

ties verbreiteten *Open Data*-Philosophie solche Datenbestände, die im Interesse der Allgemeinheit sind und die deshalb auch für alle zugänglich sein sollten. Privat versus öffentlich ist damit eine zentrale Dichotomie, in der Zugänge zu Informationen wahrgenommen werden.

Phil Zimmermann erklärte 1991, er habe *PGP* aus politischen Gründen geschrieben, um Verschlüsselung zu ermöglichen, Einzelnen die Kontrolle über ihre Privatsphäre in Bezug auf digitale Kommunikation zu geben.⁴¹ Wann das erste ritualisierte Keysigning stattgefunden hat, ist unklar, ausformuliert wurde die Anleitung für das Ritual jedoch zuerst im Jahr 2000 von V. Alex Brennen vom *MIT*.⁴² Die Entscheidung, zur Identifizierung von Personen auf staatliche Ausweisedokumente zurückzugreifen, ist eine politische Entscheidung, da die Dokumente — wie oben problematisiert wurde — Gespräche hervorbringen, in denen Menschen klassifiziert werden. Gleichzeitig gilt ein Pass oder Personalausweis unhinterfragt als neutraler und glaubwürdiger Identitätsnachweis, während etwa Führerscheine diese Qualität von vielen Teilnehmer_innen abgesprochen wird. Durch die Formalisierung von Keysigning entstehen demnach Ausschlüsse, weil der Identitätsnachweis auf ganz bestimmte Dokumente eingengt wurde, die potenziell diskriminierende Kategorien abrufen und zugleich Normen stabilisieren.

Das jeweils Nichtformalisierbare wird sichtbar, wenn man die oben aufgezeigten Widersprüche in den Praktiken betrachtet, die sich nie genau in die formalisierten Ablauf- und Handlungspläne einpassen lassen: Das *Web of Trust* stößt als Formalisierung von Vertrauensbeziehungen auf Grenzen. Die Vorstellung von Teilnehmer_innen beinhaltet, dass sie Keysigning ernst nehmen. Dazu gehört es, die staatsbürgerlichen Identitäten der Anderen *genau* zu prüfen und sich der Gegenseitigkeit von Keysigning bewusst zu sein. Wie genau welcher Schritt genommen werden muss, ist Aushandlungssache, und die Praktiken sind sehr heterogen: Weder haben alle Teilnehmenden Prüfsummen berechnet, noch ist die Überprüfung der Prüfsummen allen gleichermaßen wichtig; manchen genügt eine, manche fordern den Vergleich von drei Summen und demonstrieren damit öffentlich ihre Genauigkeit, ihre Vertrauenswürdigkeit. Worauf wird bei der Kontrolle der Papiere geachtet? Welches Tempo wählen sie in einem Ritual, das auf Geschwindigkeit hin optimiert ist, aber große Gewissenhaftigkeit erfordert? Wie entscheiden sich einzelne, mit der Situation umzugehen, wenn ihr Gegenüber kein „gültiges amtliches Dokument“ vorzeigt? In all diesen Punkten lässt das Ritual Raum für persönliche Entscheidungen und Unachtsam-

⁴¹<http://www.philzimmermann.com/EN/essays/WhyIWrotePGP.html>

⁴²http://cryptnet.net/fdp/crypto/keysigning_party/en/keysigning_party.html

keiten. Die Vorannahmen, die in der Formalisierung stecken, gehen von Computernutzer_innen aus, die ihre privaten Daten gut schützen (oder als Systemadministrator_in: die Daten der anderen Nutzer_innen ihrer Systeme). Sie sind sich optimalerweise der Risiken und Angriffsszenarien bewusst und handeln mit diesem Wissen konsequent und rational, sie nehmen sich die Zeit, Zugriffsrechte zu vereinbaren und feinkörnig einzustellen. Diese Annahmen gehen auch von Keysigner_innen aus, die alle Schritte des Rituals sehr genau beachten, weil sie sich der Bedeutung des *Web of Trust* als Ganzes bewusst sind, dessen Verlässlichkeit sie durch ihre eigene Unachtsamkeit schwächen würden. Genau dieses Bewusstsein haben laut meinen Beobachtungen nicht alle Involvierten. Sie haben, gemessen an den Annahmen, die hinter Zugriffsrechten und Keysigning stehen, zu viel Vertrauen: Wer das eigene Passwort laut sagt, vertraut darauf, dass dieses Wissen in der LUG nicht missbraucht wird. Wer — wie als Extremfall beobachtet — den Gegenüber beim Keysigning nicht ansieht, vertraut darauf, dass niemand mit einem fremden oder falschen Ausweis dort hinkommen würde. Diese Praktiken des Formalisierens stehen nicht zuletzt auch im Widerspruch zum informellen Kontext von Lernen in Linux User Groups.

Genau diese Unachtsamkeit, die persönlichen Entscheidungen beim Keysigning, der alltägliche Umgang mit Passwörtern und Zugriffsrechten, die Gutgläubigkeit, das Vertrauen ohne Kontrolle, stehen im Widerspruch zu den Annahmen, die hinter der jeweiligen Formalisierung stehen. Sie sind in diesem Zusammenhang das Nichtformalisierbare, das innerhalb der rationalen Logiken nicht vorgesehen ist: „Menschliches“, praktisches und sehr situatives Verhalten ist nicht formalisierbar. Die in diesem Kapitel untersuchten Praktiken versuchen, dies durch Abläufe, Regeln und technische Vorkehrungen abzufangen, um die auf dem Spiel stehenden digitalen Daten zu schützen. Die Formalisierung von sozialer Organisation legt Misstrauen nahe und gibt Nutzer_innen klare Schritte an die Hand, um sich zu schützen, sich Klarheit über Accounts, Schreibrechte und Identitäten zu verschaffen. Dieses Misstrauen ist ein habituell verinnerlichtes Misstrauen derjenigen, die um technische Einbruchsszenarien und staatliche Überwachungsmöglichkeiten digitaler Kommunikation mitsamt ihrer Geschichte wissen. Zu diesem Wissen gehört es auch, menschliche Anwender_innen als potenzielle „Schwachstellen“ erkannt zu haben: Sie begehen Fehler oder Unachtsamkeiten, kennen sich nicht aus oder sind schlichtweg eben nicht misstrauisch, so dass es zu Datenverlusten oder leicht überwachbarer digitaler Kommunikation kommt.

Und hier kommen wir wieder zu eingeübten habituellen Dispositionen: Distinktion wirkt weniger in Bezug auf Personen, die qua Rechtevergabe nicht an bestimmte Dateien her-

ankommen oder deren Key nicht signiert ist. Sie wirkt vielmehr da, wo Leute zeigen, ob sie diese *illusio* verinnerlicht haben, ob ihnen die Bedeutung von digitalem Datenschutz bewusst ist. Sie wirkt dort, wo manche demonstrieren, dass sie die rationale Logik der Formalisierung auch zu ihrer praktischen Logik machen. Anhand der Grenzen von formalisierten Abläufen in der Praxis konnte gezeigt werden, dass auch in diesem Bereich kein einheitlicher Habitus in FLOSS-Communities existiert: Diejenigen, die formalisierte Vorgaben nicht so ernst nehmen, tun dies in den hier gesehenen Kontexten nicht aus Kritik an den Vorgaben. Sie haben den Habitus nicht inkorporiert, mit dem sie ihre Praxis an formalisierten Vorgaben ausrichten würden und verstehen würden, warum sie dies tun.

8. Forschungsergebnisse und -perspektiven

In der vorliegenden Arbeit habe ich die Praxis von Free/Libre und Open Source Software-Communities untersucht. Dabei war mein Ausgangspunkt die Annahme, dass freie Software von der Idee her einem emanzipatorischen Ansatz folgt: Sie wird dezentral von vielen Gruppen und Einzelpersonen entwickelt und im Internet mitsamt Quellcode und Informationen zur freien Verfügung gestellt. Wissen wird in zahllosen Supportforen, Mailinglisten und Chatkanälen, aber auch bei Face-to-face-Begegnungen innerhalb von Communities vermittelt. FLOSS-Communities haben damit das Anliegen, Menschen zur Gestaltung und souveränen Nutzung von Computertechnologien zu befähigen.

Anknüpfend an vorliegende Literatur, die sich kritisch mit der Realisierung dieses Anspruchs auseinandersetzt, habe ich mich aus feministischer Perspektive mit der Praxis von FLOSS-Communities beschäftigt, und zwar mit den direkten Treffen, den persönlichen Begegnungen von Community-Mitgliedern in Linux User Groups und auf Community-Events. Inspiriert von vorhandener Literatur und ersten Erfahrungen in der Szene habe ich Differenzierungsprozesse und stilistische Distinktion innerhalb der Szene in den Vordergrund meiner Datenauswertung gestellt. Ich habe nach den konstitutiven Praktiken gefragt und nach dem impliziten praktischen Wissen, das in diesen Praktiken zum Einsatz kommt. Außerdem habe ich untersucht, wie die technischen Artefakte in diese Praktiken eingebunden sind. Damit wollte ich sichtbar machen, wie in FLOSS-Communities Unterschiede zwischen den Beteiligten hergestellt werden. Dahinter stand die Annahme, dass über die Schaffung von Differenz Machtverhältnisse in den sich als offen verstehenden Communities entstehen bzw. stabilisiert werden. Ziel der Arbeit war es, die Linien zu benennen, entlang derer dies geschieht und dabei die Rolle der technischen Artefakte einzubeziehen.

8.1. Habitus und Kapitalformen in FLOSS-Communities

8.1.1. Konstitutive Praktiken von FLOSS-Communities

Zur Bearbeitung dieser Forschungsfragen habe ich Bourdieus Konzepte von Habitus und Kapital herangezogen. Die konstitutiven Praktiken, das in ihnen mobilisierte Wissen und die in ihnen praktizierte Differenzierung sollten durch eine Untersuchung des Habitus und der relevanten Kapitalformen gezeigt werden. Bei den konstitutiven Praktiken von FLOSS-Communities ging es mir nicht um die erklärten gemeinsamen Praktiken (etwa das Teilen von → *Quellcode*), sondern um implizite Praktiken, die den erklärten zugrunde liegen. Auf der Suche nach dem, „was hier eigentlich vor sich geht“, habe ich vier Praxiskomplexe benannt, die den „offiziellen“ gemeinsamen Praktiken von FLOSS-Gruppen zugrunde liegen. Diese vier Komplexe sind (1) technische Fachsprache und ein entsprechender Humor, (2) Wissenstransfer in informellen Lernkontexten, (3) Techniknutzung und -anpassung und (4) die Formalisierung von Artefakten und sozialen Ritualen.

8.1.2. Vom homogenen Habitus zu Habitusfragmenten von Linux-Nutzer_innen

In diesen vier Komplexen habe ich Habitusfragmente aufgezeigt, die ich zur Bearbeitung der Fragen nach implizitem, praktischem Wissen und nach Differenzkategorien heranziehe. Dabei ist gerade in Linux User Groups — als Plattform für Wissensaustausch und Treffpunkt für Menschen mit sehr heterogenen Wissensständen und Anwendungsgebieten — auffällig, dass kein einheitlicher Habitus zu beobachten ist. Die Heterogenität der Habitus zeigt, dass entgegen der immer noch verbreiteten Annahme nicht nur Professionelle Linux nutzen. In LUGs treffen vielmehr Personen mit den unterschiedlichsten Wissensständen, Anwendungsfällen und persönlichen Voraussetzungen aufeinander. Die Lernprozesse, die sie in Bezug auf Computer und Software durchmachen, wurden im Detail sichtbar. Ich bin aufgrund dieser Vielfalt im Laufe der Datenauswertung zu der Bezeichnung *Habitusfragmente* übergegangen. Linux-Nutzer_innen haben Habitusfragmente in unterschiedlichem Ausmaß verinnerlicht und die Bandbreite ihrer Praktiken ist in allen Praxiskomplexen dementsprechend groß. Habitus muss demnach als individueller, relativer und dynamischer Raum gedacht werden: Je nach Situation und Thema hat jede_r Beteiligte mehr oder weniger der relevanten Habitusfragmente internalisiert. Geht es in

einer LUG beispielsweise um Netzwerkanalyse haben bestimmte Personen in dieser Situation mehr Habitus parat als andere. Wenn weitere Personen hinzukommen oder sich das Thema verschiebt, verschiebt sich die gesamte Situation in Bezug auf den Habitus: Wie viele Habitusfragmente jede_r Beteiligte internalisiert hat, steht immer in Beziehung zum Thema und den anderen Anwesenden. Da die Auseinandersetzung mit freier Software für alle Beteiligten ein fortwährender Lernprozess ist, entwickeln sich auch ihre Habitus ständig weiter.

Habitus als dynamischen Raum zu denken, wird dem Forschungsfeld aus verschiedenen Gründen gerechter als eine Typologisierung von Linux-Nutzer_innen: Es unterstellt keine Homogenität, es berücksichtigt die individuellen und kontingenten Lernprozesse, die in einem informellen Rahmen ablaufen, und es vereindeutigt nicht über Gebühr durch die Konstruktion von Typen. Eine Vereindeutigung der Habitusfragmente innerhalb des Raumes ist möglich, aber mit Vorsicht zu genießen, weil sie auf niemanden in allen Situationen so zutreffen kann. Der Raum, der die Habitus umfasst, besteht aus den unterschiedlichsten Zwischenstadien, in denen sich die meisten der Beobachteten befinden: in ihren individuellen, durch Informalität und Zufall geprägten Lernprozessen.

Dieser Raum des Habitus hat fließende Grenzen zu den Habitusfragmenten anderer Felder, in denen die Personen sich bewegen. Wie viel Zeit Einzelne in ihre Lernprozesse investieren (können), wie sie in informellen Kontexten auftreten oder ob sie zu autonomem Lernen in der Lage sind, ist stark von den Voraussetzungen abhängig, die sie bereits mitbringen und die außerhalb meiner Datenerhebung liegen. Diese Anteile des Habitus sind nicht szenespezifisch: Sie werden sehr wahrscheinlich nicht erst in FLOSS-Communities erworben, und sie sind nicht an technische Artefakte gekoppelt. Was im Spielraum des Habitus liegt, wo die Grenzen des Möglichen sind, ist sehr situativ. Gerade im informellen LUG-Umfeld waren keine explizit zum *Ausschluss* aus der Gruppe führenden Praktiken bzw. Habitusfragmente zu beobachten. Dazu hängt es zu sehr von Individuen ab, wie sehr sie sich auf Neulinge einlassen. Zudem habe ich nur die Anwesenden beobachtet, nicht diejenigen, die — aus mir unbekanntem Gründen — nicht wiedergekommen sind. Deutlich ist hier lediglich die Abgrenzung von proprietärer Software und ihren Nutzer_innen.

Eine Person, die *alle* beobachteten Habitusanteile inkorporiert hätte, verstünde die Fachsprache und die technischen Bezüge im sceneinternen Humor und hätte einen Sinn für diesen speziellen Humor. Sie könnte einerseits den „I-get-it“-Effekt (vgl. Friedman/Fried-

man 2003: 13) unter Ebenbürtigen genießen. Sie könnte andererseits die Sprache auch einsetzen, um sich als technische_n Expert_in zu erkennen zu geben oder Dritte aus der fachlichen Kommunikation auszuschließen. Die Wahrscheinlichkeit wäre hoch, dass eine solche Person mit vergeschlechtlichten Stereotypen hantiert, die Frauen Desinteresse oder Inkompetenz in Bezug auf Computertechnologien zuschreiben. Im Bereich Wissen und Wissenstransfer würde die Person über ein umfangreiches technisches Grundlagenwissen verfügen, über situative praktische Problemlösungskompetenz und die Fähigkeit zu erfolgreichem Experimentieren. Sie hätte große Teile dieses Wissens autodidaktisch und in informellen Lernkontexten von FLOSS-Communities erworben, träte in diesen Kontexten selbst als glaubwürdige_r Helfer_in auf, übernehme tendenziell Aufgaben, die ihr Spaß machen, da sie keine einforderbaren Leistungen erbringt. Im Bereich Techniknutzung würde sie die nötige Zeit in eigene Lernprozesse und in Problembearbeitung investieren, die Wertungen kennen, die bestimmter Software zugeschrieben werden. Sie würde textbasierte Programme verwenden. Die Person hätte ein Auge für die Effizienz und Ästhetik von Software. Da sie Software grundsätzlich als menschengemacht und gestaltbar ansieht, würde sie sich Software individuell anpassen, wobei sie ausgefeilte Konfigurationsmöglichkeiten schätzen würde oder faszinierend fände. Sie würde Neues erproben und technische vor nicht-technischen Lösungen bevorzugen, das Betriebssystem Linux in seiner Struktur begreifen, und sie hätte die genaue Bedienung internalisiert. Für sie wäre die Zweckentfremdung von Software die Kür der Nutzung freier Programme. Im Bereich Vertrauen und Datenschutz würde die Person Daten entlang der Dichotomie privat versus öffentlich denken. Sie stünde staatlichen Behörden als potenziellen Überwachern von Kommunikation grundsätzlich misstrauisch gegenüber und würde Angriffsszenarien auf Computer und Überwachungsmöglichkeiten kennen. Die Person wäre auch gegenüber Nutzer_innen, die nicht denselben Habitus im Umgang mit Daten zeigen, misstrauisch und würde an formalisierte Abläufe zur Eindämmung menschlicher Fehlbarkeit und zur Organisation von Sicherheitsanliegen der Communities glauben. Sie würde diese rationale Logik zu ihrer eigenen praktischen Logik machen, indem sie von Communities organisierte Rituale wie Key signing-Parties sehr ernst nimmt und die Regeln genau befolgt. Sie würde demonstrativ „best practice“ aufführen und glaubhaft vermitteln, dass sie sich der Sensibilität digitaler Daten bewusst ist.

Eine Person, die die beobachteten Habitusfragmente nicht inkorporiert hätte, würde die Fachsprache nicht verstehen und wäre damit auch von dem computeraffinen Humor ausgeschlossen. Sie würde sich nicht zutrauen mitzureden, wenn es um technische Fragen zu

Linux geht. Die Person hätte kein oder wenig Vorwissen und wäre nicht in der Lage, technische Probleme in Fachsprache zu beschreiben, sie einzugrenzen oder zu lösen. Für sie wäre es schwierig, sich diese Grundlagen autodidaktisch anzueignen oder sie in einem informellen Lernkontext zu erwerben. Sie würde sich die Probleme tendenziell von anderen lösen lassen, ohne dabei die Problemlösungskompetenz selbst zu erwerben. Im Bereich Techniknutzung wäre sie es nicht gewohnt, viel Zeit mit technischer Problemlösung zu verbringen. Sie bliebe gegenüber dem Prestige einer bestimmten Software gleichgültig und würde grafische Programme verwenden, sich nicht damit beschäftigen, was hinter ihren „Kulissen“ vorgeht. Was die Ästhetik von Software angeht, so ist die Wahrnehmung dieser Person auf die Gestaltung der grafischen Oberfläche (mit ihren Einschreibungen) beschränkt. Die Person würde Software für sich nicht als (um)gestaltbares Artefakt betrachten und wäre mit umfangreicher Konfiguration überfordert. Sie hätte beim Ausprobieren Misserfolge und würde nichttechnische Lösungen vorziehen. Bei Datenschutz und Umgang mit Daten könnte diese Person Risiken nicht einschätzen und ginge unbewusster bzw. gutgläubiger mit digitalen Daten um. Was auf dem Spiel steht, warum welche Daten zu schützen sind, was es heißt, einen Server im Internet zu schützen, ist ihr nicht bekannt, ebenso wie die Ideen hinter stark formalisierten Abläufen wie dem Aufbau des *Web of Trust*. Wenn sie sich mit Themen wie Datenschutz beschäftigt hat, so macht sie sich die Logik nicht zueigen, sondern ihre Praxis steht in Widerspruch zu dem erworbenen Wissen.

Das Konzept des Habitus geht von einer kollektiven Vereinheitlichung von Praktiken aus: Der Habitus erzeugt und ordnet Praktiken, die kollektiv aufeinander abgestimmt sind, weil die entsprechende Gruppe homogene Existenzbedingungen hat (vgl. Bourdieu 1999: 99,109). Die detaillierte Untersuchung der vier Praxiskomplexe in FLOSS-Communities zeigt jedoch, wie breit das Spektrum unterschiedlicher Praktiken ist. Dies wirft die Frage auf, ob in FLOSS-Communities überhaupt von kollektiven Praktiken gesprochen werden kann. Handelt es sich nicht vielmehr um individuelle Praktiken? Sind Individuen nicht, wie Bernard Lahire (2004) schreibt, sozial mobiler und in ihrem Leben mit immer neuen Einflüssen konfrontiert, so dass ein rigides Habituskonzept nicht greift (vgl. auch Gaboriau/Gaboriau 2006)? Unter Bezug auf Jörg Strübing gehe ich von einer Vermischung individueller und kollektiver Praktiken bzw. Arbeitsstile aus.¹ Kollektive Verfahrensweisen

¹Der Soziologe hat Anfang der 1990er Jahre das Berufsfeld der Softwareentwicklung untersucht, das damals im Prozess der Professionalisierung war (vgl. Strübing 1992: 10) und einen hohen Anteil an Quereinsteiger_innen aufwies. Seine Ergebnisse hier anzuführen, bietet sich an, weil auch FLOSS-Communities oft keine institutionalisierten oder nicht gleichermaßen institutionalisierte Gruppen sind.

entstünden innerhalb von Arbeitsteams, die mit vergleichbaren „organisatorischen und herrschaftsstrukturellen Vorgaben“ (ebd. 150) konfrontiert sind, in die auch neue Teammitglieder eingeführt werden. Kollektive Routinen entstünden damit aus gemeinsamer Praxis heraus. Gleichzeitig hätten Arbeitsstile immer individuelle Momente, die sich mit den unterschiedlichen subjektiven Dispositionen Einzelner erklären ließen:

„Ob jemand in der Lage ist, ‚leicthändig‘ Risiken einzugehen und auszuhalten oder etwa einen besonderen Blick für Exaktheit im Detail hat: All dies sind Kompetenzen, deren Grundlagen in der Erwerbsarbeit weit vorgelagerten Sozialisationsphasen gelegt werden“ (Strübing 1992: 150).

Dabei spielten die unterschiedlichen beruflichen Werdegänge und die damit verbundenen Erfahrungen eine wichtige Rolle. Von ihnen hänge etwa ab, ob Personen bereit sind, kollektive Handlungsmuster oder Arbeitsstile anzunehmen. Strübing hat diesbezüglich starke Unterschiede festgestellt zwischen denen, die ihre Computerkenntnisse über einen „theoretisch-strukturellen Zugang“ wie ein Studium erlernt haben, und denen, die durch Selbststudium oder Umschulungen einen „pragmatisch-problembezogenen“ Zugang zu den gemeinsamen Tätigkeiten hatten (ebd.). In Linux User Groups unterliegt die Kooperation nicht denselben strukturellen Voraussetzungen wie die Zusammenarbeit von lohnarbeitenden Softwareentwicklungsteams, so dass nicht derselbe Grad an Vereinheitlichung zu erwarten ist. Zudem erlaubt die Erhebung mittels teilnehmender Beobachtung keine systematische Ergründung der Werdegänge verschiedener Linux-Nutzer_innen, um ihre Stile mit ihren bereits mitgebrachten Dispositionen in klare Verbindung zu bringen. Die teilnehmende Beobachtung führt vielmehr dazu, dass die individuellen Praktiken in den Blick rücken: Die große Vielfalt an Programmen und die verschiedenen Lösungswege für Probleme schaffen auf der einen Seite wenig normierte Nutzungspraktiken und einen hohen Grad an Spezialisierung.

Auf der anderen Seite kommen vereinheitlichende Normen doch ins Spiel: Die Allgegenwart der Fachsprache setzt für die Kommunikation bestimmte Normen, etwa, dass jemand ein Problem angemessen beschreiben können muss, um gehört zu werden oder zumindest keine Bemerkungen auf sich zu ziehen, weil er_sie sich nicht präzise genug ausdrückt. Die Bezüge auf einen kulturellen Kanon in der Sprache und in der Software werden in wiederkehrenden Witzen aktualisiert. Dies wirkt dahingehend normierend, dass Neulinge sich Teile des Kanons aneignen, um am Humor der Szene teilzuhaben. In Bezug auf Kleidungsstile in der Szene sprechen Nafus et al. von einem „de-gendering pro-

cess“ (Nafus et al. 2006: 26): Frauen machten die Erfahrung, dass sie wegen ihrer Kleidung besondere Aufmerksamkeit auf sich zögen und passten sich deshalb einem Stil an, den sie als „männlich“ bezeichneten (vgl. ebd. 25 f.). In den Bereichen des informellen Lernens und der Nutzungspraktiken entstehen geteilte Normen, wenn erfahrenere Nutzer_innen Softwareempfehlungen aussprechen. Sie rekurren dazu oft auf das Prestige bestimmter Software und verbreiten es so weiter. Zudem greifen viele Lernende auf dieselbe Software zurück, die ihnen Helfer_innen aus ihrer Praxis heraus empfehlen, worüber es zu einer gewissen Vereinheitlichung kommt. Eine gewisse Vereinheitlichung ästhetischer Vorstellungen über Einzelpersonen hinaus wurde am Beispiel der Gestaltung von Desktops sichtbar. Diese Normierung tritt jedoch keineswegs notwendigerweise ein, sondern ist — analog zu den Lernprozessen — kontingent.

8.1.3. Die gegenseitige Konstitution von Kapital und technischen Artefakten in der Praxis

Habitus und Feld stehen in einem wechselseitigen Bedingungsverhältnis. Der Habitus wird erst durch das spezifische Feld ermöglicht und konstituiert es gleichzeitig. Im Fall von FLOSS-Communities zeichnet sich das Feld durch die Allgegenwart technischer Artefakte aus. Wie gezeigt, sind die szenespezifischen Habitusfragmente diejenigen, die nur *mit* technischen Artefakten erworben und ausagiert werden können. Technische Artefakte sind damit ein essentieller Teil der objektiven Strukturen des Feldes. Diese Verwobenheit von Habitus mit Technologien spiegelt sich in der feldspezifischen Ausprägung der Kapitalformen wider.

In FLOSS-Communities stehen soziales, symbolisches und kulturelles Kapital auf dem Spiel: Soziales Kapital besteht aus den mobilisierbaren Beziehungen innerhalb der gesamten FLOSS-Szene oder der speziellen Community. Innerhalb der größeren Szene bedeutet soziales Kapital, auf Community-Events Bekannte wiederzutreffen, sich in der Szene zu orientieren, was aktive Personen und Projekte angeht. Wer selbst mit einem konkreten Projekt assoziiert wird, häuft sehr schnell soziales Kapital an. Diese Beziehungen sind etwa dann mobilisierbar, wenn man jemanden um Hilfe bei einem technologischen Problem oder bei der Vermittlung von Kontakten bitten kann. In einer LUG ermöglicht soziales Kapital es den Beteiligten vor allem, spontan Helfer_innen zu finden oder innerhalb der Gruppe selbst als Spezialist_in für bestimmte Fragen bekannt zu sein. Ein solcher Ruf ist eng mit symbolischem Kapital verknüpft: Wer für ein bestimmtes Spezialwissen bekannt

ist, kann um so besser als glaubwürdige_r Helfer_in auftreten. Bis dahin ist soziales Kapital nichts, was FLOSS-Communities von diversen anderen sozialen Zusammenhängen oder Szenen unterscheidet. Unter Einbeziehung der technischen Artefakte kommt eine spezifische Ausformung von sozialem Kapital hinzu. Diese Beziehungen können dann auch über technologische Mittel hergestellt und erhalten werden: Die ständig mobilisierbare Hilfe Dritter kann dann darin bestehen, über Chat sofort zahlreiche Online-Kontakte etwas fragen zu können. Es kann ebenso aus Zugangsdaten zu nichtöffentlichen elektronischen Ressourcen bestehen, etwa zu dem → *Wiki* einer Gruppe, wo gemeinsam gesammelte Informationen zusammengetragen wurden, oder zu Servern von Bekannten, auf die man schnell und ortsunabhängig zugreifen kann. Die technischen Zugänge werden über persönliches Vertrauen und ein geteiltes Verständnis von der Sensibilität der Daten weitergegeben — der Zugang zu technischen Ressourcen ist damit an soziale und symbolische Ressourcen gebunden. Soziales und symbolisches Kapital werden auch bei dem szenespezifischen Ritual des Keysigning angesammelt: Bei einer späteren Kontaktaufnahme per E-Mail ist den Kommunikationspartner_innen ersichtlich, dass sie den jeweils anderen Schlüssel signiert haben, dass sie sich also bereits auf einem Community-Event begegnet sind. Selbst wenn zu dem Zeitpunkt kein persönlicher Kontakt entstanden ist, so ist das Zusammentreffen digital dokumentiert und kann als Anknüpfungspunkt verwendet werden. Das symbolische Kapital digitaler Signaturen kann so in soziales Kapital umgewandelt werden. Die Kontaktaufnahme wird durch technologische Mittel aus der Pseudonymität des Internet gehoben. Dies gilt vor allem dann, wenn beide Kommunikationspartner_innen einen *Sinn* für Keysigning haben. Sozialität ist in FLOSS-Communities nicht vom Technischen trennbar: Technische Artefakte und formalisierte Prozesse konstituieren soziale Beziehungen und werden umgekehrt in sozialen Netzwerken hergestellt.

Kulturelles Kapital ist zentral in FLOSS-Communities. Inkorporiertes Kulturkapital ist die wichtigste Ressource in FLOSS-Communities. Community-Mitglieder eignen sich die Fachsprache, das technische Wissen, die situative praktische Problemlösungskompetenz an, sie investieren Zeit und haben ein Zeitgefühl für die Dauer bestimmter Prozesse. Sie machen sich mit der Zeit einen Sinn für Ästhetik zueigen und erlernen gewisse körperliche Bewegungsabläufe. Objektiviertes Kulturkapital wird etwa dort sichtbar, wo sie souverän ihre Hard- und Software manipulieren, Teile selbst umbauen oder umprogrammieren. Es ist mit inkorporiertem Kulturkapital derart verknüpft, dass sie wissen oder lernen danach zu fragen, welche genauen Chipsätze in welchen Teilen verbaut sind und ob sie von Linux unterstützt werden. Sehr viele Community-Mitglieder verfügen eben nicht nur über

das objektivierte Kulturkapital, sondern *auch* über das nötige inkorporierte Kulturkapital zu seiner Nutzung. All dies fällt denen leichter, die die Fähigkeit mitbringen, autodidaktisch und in informellen Kontexten zu lernen, in denen sie oft auf sich selbst gestellt sind. Kulturelles Kapital wird zwar von einigen in Form von LPI-Zertifikaten institutionalisiert, die ihre beruflichen Chancen damit erhöhen möchten. Innerhalb des informellen Settings des Community-Lebens erfolgt Institutionalisierung von Kulturkapital jedoch anders: Vorträge auf Community-Events und Artikel in Fachzeitschriften sind keine szenespezifische Besonderheit, dass solche Beiträge online zugänglich sind und damit langfristig einer breiten Öffentlichkeit zur Verfügung stehen, schon eher. Dies hat sich in anderen gesellschaftlichen Bereichen unter anderem durch die Praxis des Open-Source-Umfeldes etabliert, Vorträge online zu stellen oder Livestreams anzubieten. Belege für die eigenen Fähigkeiten sammeln viele Community-Mitglieder auch über online archivierte Beiträge im Bereich freie Software an. Die Namensnennung als Beitragende_r zu Softwareprojekten oder die öffentlich zugänglichen Archive von Mailinglisten und Support-Foren sind dauerhafte Belege für ihr Engagement (vgl. Kelty 2005a: 425). Speziell bei den persönlichen Treffen von Linux User Groups habe ich solche Institutionalisierung von Kulturkapital kaum beobachtet. Einen dauerhaften Beleg in objektivierter Form gibt es in diesem Rahmen nicht, sondern hier werden soziales, symbolisches und inkorporiertes kulturelles Kapital direkt, offline und undokumentiert gehandelt.

Dieses kulturelle Kapital könnte ohne die technischen Artefakte nicht eingesetzt werden. Viele der entsprechenden Habitusfragmente können in der Praxis nur in Verbindung mit Computern zum Einsatz kommen und ihre sozialen Wirkungen entfalten: Situative Probleme können ohne Rechner nicht gelöst werden, Probleme können ohne Tests nicht eingegrenzt werden, zum Ausprobieren bedarf es einer Maschine. Die speziellen zeitlichen Erfordernisse sind den Technologien oft eigen, ohne sie verliert das spezifische Zeitgefühl der Linux-Nutzer_innen seine Relevanz. Die Praktiken, die am angesehensten sind, etwa die Zweckentfremdung von Software oder innerhalb der LUG das Lösen technischer Probleme, hängen ganz von der Präsenz technischer Artefakte ab. Das inkorporierte Kulturkapital ist — mit Latour gesprochen — immer das Kapital eines Hybrid-Akteurs aus Mensch und Maschine.

8.2. Zur Herstellung von Differenz in FLOSS-Communities

Zu meiner Forschungsfrage *Wie und entlang welcher Kriterien wird in den konstitutiven Praktiken von Linux-Communities Differenz hergestellt?* hatte ich die Hypothese aufgestellt, dass die zentralen Differenzkategorien Gender, Herkunft oder Ethnizität, Klasse, Sprache und Umgang mit Computern sind. Die Arbeit hat im Detail gezeigt, wie in der Praxis Unterschiede hergestellt werden und wie sie sozial bedeutsam gemacht werden (oder nicht). Dabei wurde deutlich, dass die in den Hypothesen vermuteten Kategorien *in sich* viel ausdifferenzierter sind. Jede von ihnen tritt in zahlreichen Facetten auf. Eine Ausnahme bildet Ethnizität, wozu ich keine nennenswerten Daten erhoben habe (siehe dazu auch die offen gebliebenen Fragen in Abschnitt 8.3).

8.2.1. Wissen und Vorbildung

Von einer Differenzkategorie ‚Klasse‘ zu sprechen, ist aufgrund der Konzeption meiner Studie kaum möglich. Bildung, Wissen und Vorwissen spielen als Aspekte dieser Kategorie jedoch eine zentrale Rolle in FLOSS-Communities. Der Erwerb von kulturellem Kapital in den Communities ist eng an den bereits erfolgten Erwerb von kulturellem Kapital in anderen Feldern gekoppelt. Kapitel 5 hat gezeigt, dass die an FLOSS-Projekten Beteiligten das nötige Kapital, etwa die situative praktische Problemlösungskompetenz, in informellen Lernkontexten erwerben. Diese Informalität wird von der bisherigen FLOSS-Forschung als besonderer Vorteil bezeichnet. Gleichzeitig ist vielerorts die soziale Selektivität von informellen Zusammenhängen gezeigt worden.

In FLOSS-Communities bedeutet dies, dass diejenigen, die eh schon über weniger Kapital verfügen, um so mehr Schwierigkeiten haben, mit ihren Problemen bzw. unzureichenden Problembeschreibungen gehört zu werden. Auf sich gestellte Individuen müssen den LUJG-Situationen gewachsen sein. Dazu sind sie eher in der Lage, wenn sie bereits vorher über bestimmtes Vorwissen oder einen technikaffinen Habitus verfügen. Hier kommt die Computerbildung ins Spiel, die viele Community-Mitglieder bereits *vor* ihrer Szenezugehörigkeit auf ebenfalls informellem Weg erworben haben. Dies ist besonders bei männlich sozialisierten Personen wahrscheinlich, die zuerst in den 1990er Jahren oder früher mit Computern in Berührung kamen (und das vergeschlechtlichte Stereotyp gleich mit erlernt haben). In der Szene bewegen sich viele, die sich ihr komplettes linux- oder computerbezogenes Wissen autodidaktisch bzw. in informellen Kooperationen angeeignet haben.

Ich vertrete die These, dass dies nicht unbedingt mit ihren formalen Bildungsabschlüssen korreliert, ohne dies systematisch erhoben zu haben, und entgegen der Befunde von Ghosh et al. (2002: 12). Die erforderliche Zeit haben viele von ihnen bereits in jungen Jahren investiert und dabei zentrale Fähigkeiten ausgebildet, um technische Probleme einzugrenzen und durch eigene Eingriffe zu beheben, sich im erfolgreichen Ausprobieren zu üben, die damals noch gängigeren textbasierten Programme zu erlernen, Erfahrung und ein Gefühl dafür zu bekommen, wie lange etwas dauern könnte, einen Blick für historisch gewachsene Software bekommen, die Glaubwürdigkeit von Lösungsvorschlägen einzuschätzen. Für erworbenes Faktenwissen gibt es zwar kaum übergreifende, klare Maßstäbe, weil die Lernprozesse individuell, nichtlinear und informell gestaltet werden und weil die Technologien selbst so schnell weiterentwickelt werden. Um so zentraler sind diese grundlegenden Kompetenzen, die bei situativen und lokalen Problemen zum Einsatz kommen. Wie viel Erfahrung jemand bereits mit informellem Lernen hat, bestimmt stark darüber mit, wie er_sie sich in FLOSS-Communities als Lernumfeld zurecht findet und mit welchen Erfolgchancen. Ein computerbezogenes Spezialwissen der letzten 20–30 Jahre ist ein wichtiger Schlüssel zur weiteren Anhäufung oder zur Aktualisierung solchen Wissens.

8.2.2. Geschlecht

„Gender“ ist angesichts der komplexen Herstellung und Wirkungsweise von Geschlecht in FLOSS-Communities ein zu pauschales Etikett: Auf der Mikroebene der Praxis werden laufend Bezüge sowohl zur Makroebene gesellschaftlicher Strukturen als auch zur Repräsentationsebene der Diskurse und kulturellen Symbole hergestellt. Geschlecht wird auf allen drei Ebenen soziale Bedeutung verliehen, was sich in allen vier untersuchten Praxis-komplexen zeigte:

Besonders präsent sind Stereotype, also der Einfluss der Repräsentationsebene auf Praktiken und Strukturen. Community-Mitglieder, die sich als heterosexuell und männlich verorten, rufen die Personen, die sie als Frauen wahrnehmen, ständig als solche an. Damit mobilisieren sie einen bestimmten Bestand an Vorannahmen in Bezug auf ihr inkorporiertes Kulturkapital. Diese Vorannahmen sind sehr oft Reproduktionen eines (in der ganzen Gesellschaft verbreiteten) Stereotyps, das nur entlang eines zweigeschlechtlichen, heterosexuellen Geschlechtersystems funktioniert: Dieses Stereotyp besagt, dass „Frauen“ sich nicht für die Funktionsweise von Computern interessieren und dass sie potenzielle

Objekte sexuellen Interesses darstellen. Es bietet Anschluss für die verschiedenen beobachteten Praktiken, etwa für die Reproduktion von Heteronormativität im Humor oder für sexistischen Humor, der Frauen mit passiven Sexualobjekten assoziiert, die wie Computer von „Männern“ gesteuert werden. Die Vorannahme, „Frauen“ verfügten über wenig linuxspezifisches Kulturkapital, ist die Basis dafür, dass im technischen Linux-Support Männlichkeiten und Weiblichkeiten hergestellt werden, und zwar tendenziell durch alle Beteiligten: Wie weit die Hilfsbereitschaft von Helfer_innen geht, was dem Gegenüber abverlangt wird und ob dabei fremde Computer vereinnahmt werden, all dies sind *auch* geschlechtlich konnotierte Interaktionen. Dasselbe Stereotyp ist in Form von Genderskripten (vgl. Akrich 1992, 1995; Rommes 2002) in freie Software eingeschrieben, wie an den Beispielen eines Standard-Hintergrundbildes und der Access Points gezeigt.

Dieser Bezug auf „Frauen“ als nicht technikaffin konstruiert „Männer“ als vermeintlich homogene Expertengruppe, auch wenn die gesamte Gruppe sehr heterogen ist, was den Bestand an kulturellem Kapital bzw. an inkorporierten Habitusfragmenten betrifft. Aus der hinzugezogenen Literatur ging hervor, dass auch das informelle Lernen im Bereich Computerkompetenzen, das dem Lernen innerhalb von FLOSS-Communities vorgelagert ist, unter bestimmten Voraussetzungen erfolgt. Hier fließen bereits in der Gesellschaft verbreitete Vorannahmen ein, die Mädchen und Frauen weniger Interesse oder Begabung unterstellen. Anhand von vorhandener Literatur habe ich ebenfalls argumentiert, dass die in Kapitel 7 untersuchten Praktiken des Formalisierens die in der Gesellschaft etablierte Geschlechterordnung auf symbolischer Ebene fortschreiben. Das Geschlechterstereotyp ist so präsent, dass es ebenfalls als Habitusfragment, als inkorporiertes Wahrnehmungsschema, bezeichnet werden kann: Es strukturiert die Wahrnehmung von Personen und die Praxis und wird gleichzeitig durch Praxis aktualisiert bzw. stabilisiert. Es handelt sich hierbei nicht um einen szenespezifischen Habitus oder um szenespezifische Strukturen, sondern um ein in großen Teilen der Gesellschaft verbreitetes Bild. Dazu gehört in FLOSS-Communities, dass auch diejenigen, die sich selbst nicht auf Geschlechterstereotype beziehen, nicht dagegen einschreiten, wenn andere es tun. Auf diversen Community-Events wurde als Ziel geäußert, mehr Frauen für die FLOSS-Szene zu gewinnen. Die meisten Community-Mitglieder haben jedoch kaum Problembewusstsein dafür, dass sie ihre eigene Praxis tiefgreifend hinterfragen müssten, um Diskriminierung entgegenzuwirken. Meine Beobachtungen zeigen vielmehr, dass Geschlecht eine der zentralen Differenzkategorien ist. Mit zunehmender Involviertheit einzelner Frauen nimmt zwar das Vorurteil ab, sie interessierten sich nicht für freie Software. Sie werden jedoch weiterhin als „Frauen“

adressiert und mit den Geschlechterstereotypen konfrontiert, die in Witze, Lernverhältnisse und Software eingeschrieben sind.

8.2.3. Stilistische Distinktion über den Umgang mit technischen Artefakten

Meine Hypothese, dass Unterschiede auch über den Umgang mit technischen Artefakten hergestellt werden, kann jetzt ebenfalls differenzierter formuliert werden: Linux-Nutzer_innen bilden Routinen im Umgang mit ihren Artefakten heraus, von denen sie durch den technischen Support andere profitieren lassen. Dabei kommen nicht nur die genannten Differenzkategorien ins Spiel, sondern auch die Distinktion über Stil: Nicht jede Praktik muss distinguierend wirken oder so gemeint sein. In allen vier Praxiskomplexen gab es jedoch Beispiele, die zumindest einen Interpretationsspielraum dafür eröffnen, „Stil als Distinktionsvehikel“ (Bohn 1991: 87) zu lesen.

Im Bereich von Sprache und Humor können Expertengespräche und Insiderwitze zur Selbstinszenierung vor denjenigen genutzt werden, die die technische Fachsprache nicht verstehen. Selbst wenn dies nicht immer intendiert ist, schließt die Verwendung bestimmter Begriffe oder anderer Zeichen Personen aus. Unter den Szenegänger_innen kommt der „secret handshake“ (Friedman/Friedman 2003) zustande, er wirkt jedoch oft gleichzeitig als Überlegenheitsdemonstration auf weitere Anwesende. Community-Mitglieder laden die technischen Artefakte und ihre Nutzung in vielen Situationen ästhetisch auf. Dies kann einen ironischen Anteil haben, etwa bei den „Glaubensfragen“, welche genaue Software sie verwenden. Gleichzeitig kann die Routine, die sie sich angeeignet haben als Teil der Stilisierung wirken: Wer eine Software nutzt, der Prestige zugeschrieben wird, zeigt nicht nur, dass er_sie den Sinn dafür hat und die Software zu schätzen weiß. Er_sie kann auch die eigene Effizienz im Umgang mit diesem „Tool“ vor denen vorführen, die etwas lernen wollen. Dabei kann es, wie gesehen, inhaltlich auch um ein anderes Thema gehen, in das das Spezialwissen etwa über einen Texteditor wirkungsvoll eingebracht wird. Die Wirkungen können sein, dass andere einhaken, um nach bestimmten Details der Bedienung oder Konfiguration zu fragen oder angesichts der Geschwindigkeit gar nicht mitkommen. Manche inszenieren ihre Lösungswege durch einen besonders souveränen Umgang mit den Artefakten und einer gleichzeitigen Kommunikation mit dem anwesenden Publikum. Lösungswegen wird eine unterschiedliche Ästhetik beigemessen, für die die jeweils Anwesenden einen entsprechenden Habitus erlernt haben müssen, um sie zu verstehen. So gelten automatisierbare Wege oder Wege, die Konventionen einhalten als

„sauber“, solche, für die Personen eine besondere Kreativität eingesetzt haben als „elegant“, solche, die diesen Kriterien nicht entsprechen, als „quick and dirty hack“. Artefakte zu personalisieren ist Teil der Stilisierung und findet auf den verschiedensten Wissensstufen statt, von Hardwareumbauten über Anpassungen der grafischen Oberfläche bis hin zu ausgefeilten textbasierten Konfigurationen. Dabei zeigen die Community-Mitglieder deutlich unterschiedliche Geschmäcker, etwa für schlichte Oberflächen, für schlanke textbasierte Programme oder für ‚stylische‘, aufwändige Oberflächen. Mit diesen Geschmäckern wird oft implizit Verschiedenes assoziiert: Teilweise blicken diejenigen, die voraussetzungsvolle textbasierte Programme einsetzen herunter auf diejenigen, die Wert auf ästhetisch gestaltete Oberflächen legen. Sie betonen, dass für sie nicht die Aufmachung, sondern die technische Funktionalität im Vordergrund stehe.

Diese Formen von Distinktion über Stilisierung finden wohlgerne *möglicherweise*, aber nicht unbedingt statt. Sie werden nur von den Personen wahrgenommen, die in der jeweiligen Inszenierung überhaupt eine Relevanz, ein Zeichen von besonderer Versiertheit oder Kreativität lesen. Situativ *können* Einzelne „ihr Gesicht verlieren“, wenn sie zeigen, dass sie den entsprechenden Habitus nicht beherrschen. Andererseits können sie auch „über den Dingen stehen“, die Inszenierung wahrnehmen, das Spiel aber nicht mitspielen. Zudem ist aufgrund der Diversität der Szene und der Software die Akzeptanz für Verschiedenheit insgesamt groß. Für diese stilisierenden Praktiken ist wiederum genau der Bezug auf freie Software, auf technische Artefakte, essentiell, ohne die die szenespezifische Stilisierung nicht denkbar wäre.

Resümierend kann ich zur praktischen Herstellung von Unterschieden in FLOSS-Communities festhalten, dass Wissen bzw. inkorporiertes Kulturkapital, Geschlecht und Stil die deutlichsten Kategorien sind, entlang derer dies geschieht. Damit wirken Differenzkategorien und szenespezifisches akkumuliertes Kapital in den Habitus der Community-Mitglieder zusammen. Dies hat verschiedene Implikationen:

Erstens sind die Kapitalbestände der Linux-Nutzer_innen stark im Fluss, während dies für Differenzkategorien nicht gilt. Die Herstellung von Unterschieden zur Abgrenzung von anderen über szenespezifisches Kapital bleibt problematisch, rückt jedoch in ein anderes Licht, wenn das soziale Gefüge sich durch die allgegenwärtigen Lernprozesse ständig verändert. Der Prozess, in dem die Beteiligten vor allem das kulturelle Kapital erwerben, ist langwierig, sozial und kulturell bereits voraussetzungsreich. Dennoch nehmen viele Community-Mitglieder den Aneignungsprozess ja auf sich und verändern damit ihre Stellung

in diesem Gefüge. Im Gegensatz dazu ist die Zuschreibung von Geschlecht strukturell tief verankert, nicht leicht veränderbar, ‚an den Körper geheftet‘. Bei allem Kapitalerwerb bleibt die Identifizierung der Menschen als „Frauen“ oder „Männer“ der Interaktion vorgeschaltet und strukturiert sie. Die Herstellung von Differenz über Geschlecht wäre damit auf einer anderen Ebene des Habituskonzeptes zu verorten als die über Kapital hergestellte. (Und dies ist vielleicht eher die Ebene, die Bourdieu meint, nicht die Lernebene. Zudem ist ein kulturelles Kapital, das Personen Geschlecht zuschreibt, kein szenespezifisches.)

Zweitens zeigt diese Zusammensetzung des Habitus, dass nicht die reine Konkurrenz um akkumulierbare Ressourcen auf dem Spiel steht.² Sie werden vielmehr auf tieferer Ebene untermauert durch strukturelle Diskriminierung, die nicht szenespezifisch ist, hier aber durch den starken Technologiebezug spezifisch ausgeprägt ist.

Drittens drängt die beobachtete Herstellung von Differenz, und besonders die Materialbeispiele zum Thema Sexismus, die Frage nach der Veränderbarkeit des Feldes auf. Bohn fragt (wie etliche Kritiker_innen, die Bourdieu Determinismus vorwerfen), ob Situationen lediglich „Reproduktionsveranstaltungen vorgängig fixierter Strukturen“ seien (ebd. 106). Die Bandbreite von Habitusfragmenten insgesamt widerlegt dies auf der einen Seite: Die Individuen finden sehr verschiedene Umgänge mit allen untersuchten Praktiken. Sie nehmen in die Artefakte eingeschriebene Skripte an oder lehnen sie offen kritisch ab, sie lesen Selbstinszenierung respektvoll als Expertise oder kritisieren sie als für die fragende Person nicht hilfreich. Auf der anderen Seite wirkt die Produktion von Geschlechterdifferenzen durch ihre häufige Wiederkehr auf allen Ebenen doch wie eine „Reproduktionsveranstaltung“. Akteur_innen haben hier eher die Wahl zwischen Umgangsweisen damit, sie *müssen* jedoch ständig erneut damit umgehen. Dies betrifft theoretisch alle Geschlechter, praktisch jedoch die von abwertenden Zuschreibungen Betroffene. Die Mehrheit der Community-Mitglieder scheint sich mit hegemonialen Vorstellungen von Männlichkeit zu identifizieren, sie nicht zu hinterfragen und bringt damit kein Problembewusstsein dafür mit, wenn in der Praxis Differenzen über dieser Kategorie hervorgebracht werden. Das Material führt zudem zu der These, dass diejenigen, die als „Frauen“ identifiziert werden, aufgrund von strukturierenden Vorannahmen relativ mehr linuxspezifisches kulturelles Kapital inkorporieren müssen als diejenigen, die als „Männer“ wahrgenommen werden,

²Cornelia Bohn wirft in Bezug auf Bourdieus Konzept diese Frage kritisch auf: Ist seine Soziologie eine Wissenschaft von der Konkurrenz um Mangelgüter? Sind Tausch, Kampf und Konkurrenz die entscheidenden strukturbildenden Mechanismen (vgl. Bohn 1991: 97 ff.)?

um zumindest auf der Kapitalebene von Wissen und Stil als Ebenbürtige behandelt zu werden.

8.3. Fazit und Ausblick

FLOSS-Communities tragen dazu bei, Wissen über und Kompetenzen im praktischen Umgang mit Computern, speziell mit freier Software, zu vermitteln. Linux User Groups praktizieren durch ihren technischen Support durchaus einen offenen Umgang mit Wissen. Ich habe untersucht, wie dies in der Praxis genau geschieht. Die detaillierten Einsichten in die Praxis ausgewählter FLOSS-Communities haben gezeigt, wie dort im Wissenstransfer auch Differenzen reproduziert werden, und zwar entlang der Kategorien von technischem Wissen und Geschlecht. Zudem findet eine oft subtile Distinktion über Stile statt, die die Beteiligten im Umgang mit freier Software pflegen. Dies führe ich darauf zurück, dass die Community-Mitglieder sich und ihre Praxis bzw. ihre Pädagogik nicht als solche reflektieren. Reflexion und eine Bereitschaft zur individuellen und kollektiven Selbsthinterfragung wäre eine Voraussetzung für Veränderung. Diese Veränderung könnte zunächst in einem bewussteren Umgang mit dem informellen und stark vergeschlechtlichten Kontext der Communities bestehen. Letztendlich vertrete ich jedoch die Ansicht, dass diese Bereitschaft zur Selbsthinterfragung weiter gehen müsste. Dies würde bedeuten, dass alle Beteiligten sich mit ihren eigenen Privilegien, mit ihrer eigenen Verstrickung in gesamtgesellschaftliche Dominanzverhältnisse auseinandersetzen müssten. Sie müssten dazu die Bereitschaft entwickeln, sie aufzugeben, und es müsste bewusst an einer anderen, nicht differenzierenden bzw. diskriminierenden Praxis gearbeitet werden. Anregungen können dafür etwa feministische und postkoloniale Theorien wie die Kritische Weißseinsforschung geben (vgl. z. B. Frankenberg 1996; Rommelspacher 1995; McIntosh 2003; Wachendorfer 2001; Wollrad 2005). Das ist ein sehr hoher Anspruch: Ein solcher Prozess kann sich nicht auf FLOSS-Communities beschränken, sondern müsste in der ganzen Gesellschaft stattfinden — hier nehme ich mich selbst nicht aus. Ohne solche Prozesse tendieren die Communities dazu, sich in ähnlicher Zusammensetzung, mit ähnlichen Praktiken und Selbstverständlichkeiten zu reproduzieren.

Meine eingangs formulierte Annahme, dass FLOSS-Communities emanzipatorisch mit Wissen umgehen, muss ich damit revidieren, auch wenn sich keineswegs ein homogenes Bild von Praxis ergeben hat. Die Idee der Emanzipation wird jedoch von den Betei-

ligten selbst kaum über den Horizont der *GNU General Public License* hinaus geäußert. Es wurde deutlich, dass Äußerungen über Politik sich auf Anspielungen beschränken, die aktuelle Netzpolitik kritisieren, oder auf eine Ironie, die mit der Idee spielt, politische Probleme technisch lösen zu können. Die politischen Bezüge freier Software, die mein ursprüngliches Interesse am Thema provoziert haben, finden sich demnach in den untersuchten Communities kaum. Dies ist in anderen Teilen der Szene und in benachbarten Szenen anders: Liberale, libertäre und anarchistische Versatzstücke sind insgesamt starke Bezugspunkte für viele, die sich FLOSS-Communities oder Hackerkreisen zurechnen. Vor dem Hintergrund meiner Ergebnisse verwundert vor allem der Bezug auf Anarchismus, ist doch die Praxis sehr weit von der Idee der Herrschaftsfreiheit entfernt. Der liberale Bezug auf individuelle Freiheit hingegen ist mit meinen Beobachtungen vereinbar. Dies impliziert folgende Interpretation: Individuen verteidigen sich, ihre Computer und die Freiheit mit ihnen tun zu können, was sie möchten, gegen jegliche Einschränkung, vor allem gegen staatliche Regulierung. Dazu organisieren sie sich als Bürgerrechtler_innen mit netzpolitischem Schwerpunkt. Aufgrund der aufgezeigten Heterogenität der Szene erscheint es durchaus möglich, dass dies ihr einziger politischer Konsens ist.

Hier möchte ich mit den offenen Fragen anknüpfen, die die vorliegende Arbeit aufwirft. Sie liegen auf zwei unterschiedlichen Ebenen: Zunächst ergeben sich aus der Anlage der Arbeit Fragen bezüglich ihrer Reichweite. Weiter unten greife ich dagegen Fragen auf, die über FLOSS-Communities hinausgehen.

Um die Praxis von FLOSS-Communities zu untersuchen, habe ich die Datenerhebung auf die Mikroebene fokussiert. An verschiedenen Stellen wurde deutlich, dass eine Rückkopplung der Praxisbeobachtungen an sozialstrukturelle Merkmale der Beobachteten wichtig wäre, um Aussagen über die sozialen Voraussetzungen der in den Communities aktiven Personen zu machen. Das Leben, das sie außerhalb der Communities führen, hat mich nur insoweit interessiert, als sie es innerhalb der Communities selbst thematisiert haben. Dies führte dazu, dass ich nur von Einzelnen etwas über Bildungshintergrund, beruflichen Werdegang oder Privatleben wusste und dies aus Analyse herausgehalten habe. Den Zusammenhang zwischen Linux User Groups und der sozialen Selektivität von informellem Lernen habe ich über Literatur hergestellt. Wie genau makrosoziologische Kategorien hier auf der Mikroebene wirken, bleibt ein Forschungsdesiderat (vgl. auch Bremer 2007: 283 f.). Über diese Beziehung könnten durch eine entsprechende sozialstrukturelle Rückkopplung weitere Erkenntnisse gewonnen werden.

Hieran schließt sich eine weitere offen gebliebene Frage an, und zwar, welche Rolle ökonomisches Kapital spielt. Dieses Thema tauchte in Linux User Groups eher am Rande auf, z. B. wenn Einzelne über ihre Lohnarbeit oder *Hartz IV*sprachen oder über Hardwarekäufe. Um zu untersuchen, ob und wie welches in Communities erworbene Kapital in ökonomisches umgesetzt wird, wäre demnach eine Erhebung nötig, die über den Rahmen des Community-Lebens hinausgeht. Anhand meiner Daten kann ich kaum Aussagen darüber treffen, ob ökonomisches Kapital tatsächlich eine derart marginale Rolle spielt oder ob es — in Bourdieus Worten — nur besonders gut verschleiert wird.

Wie eingangs thematisiert, ist meine teilnehmende Beobachtung eine Datenerhebung aus situierter Perspektive. Die Ergebnisse müssen als solche gelesen werden. Dies bedeutet, dass ich für bestimmte Achsen von Diskriminierung oder Herrschaftsverhältnisse sensibilisierter bin als für andere, was sich in der Beobachtung, in der Auswertung und der theoretischen Einordnung niedergeschlagen hat. Zu weiteren möglichen Differenzkategorien, etwa Alter, Behinderung oder Hautfarbe, habe ich lediglich vereinzelte Notizen. Teilweise habe ich mich komplett gegen eine Beobachtung der Situationen entschieden, um mögliche Differenzierungen nicht dadurch zu reproduzieren, dass ich mich mit meinem Notizbuch neben die Beteiligten setze. Ich möchte also dazu anregen, das Forschungsfeld von anderen Positionen aus im Hinblick auf Differenz und Distinktion zu analysieren. Dies könnte das Bild ausdifferenzieren, das so nur aus einer weißen, feministischen, Linux-Novizinnen-Perspektive besteht. Zudem konnte ich verschränkte Wirkungen verschiedener Diskriminierungsachsen oder Differenzkategorien, im Sinne einer intersektionalen Studie kaum dingfest machen.

In Anbetracht der wenigen Forschungsarbeiten, die überhaupt zu Linux User Groups vorliegen, könnte die weitere Forschung auf mehr LUGs ausgeweitet werden. Meine Ergebnisse stützen sich angesichts der Anzahl existierender Gruppen auf ein sehr kleines Sample.

Weiterführende Fragen ergeben sich daraus, dass die vorliegende Arbeit zum Zeitpunkt ihrer Fertigstellung in einem veränderten politischen Kontext steht: Durch die 2006 gegründete *Piratenpartei Deutschland* hat sich der Kreis netzpolitischer Akteur_innen erweitert und entscheidend verändert. Bereits die zwei Prozent Zweitstimmen, die sie bei der Bundestagswahl 2009 erreichte, haben Einfluss darauf ausgeübt, wie die im Bundestag vertretenen Parteien netzpolitische Themen behandelt haben (vgl. Niedermayer 2010: 843). 2011 ist die Partei mit 8,9 Prozent der Stimmen ins Berliner Abgeordnetenhaus ein-

gezogen. Ich möchte vor allem drei an meine Arbeit anknüpfende Forschungsdesiderate hervorheben:

Für die politische Soziologie wäre es erhellend, die Schnittstellen von FLOSS-Communities, netzpolitischen Akteur_innen, Netzaktivist_innen und der Piratenpartei zu untersuchen. Dies könnte neue Erkenntnisse darüber bringen, wie es um politische Einstellungen und (partei-)politische Aktivitäten in diesem Umfeld bestellt ist. Diese Szenen oder Gruppierungen sind — so meine Annahmen — allesamt in sich heterogen, was politische Ziele, Wege zur ihrer Erreichung und Zusammensetzung der Anhängerschaft angeht. Sie weisen gleichzeitig starke personelle Überschneidungen auf. Deshalb gehe ich bei einem relevanten Personenanteil davon aus, dass er in einem ähnlich technikaffinen Umfeld „technisch sozialisiert“ wurde, das ich in dieser Arbeit umrissen habe, und dass bei ihm ähnliche Habitusfragmente zu finden sind. Was bedeutet dies für die politische Ausrichtung und die internen Prozesse der Piratenpartei und anderer netzpolitischer Akteursgruppen?

Der frappierend geringe Frauenanteil in der Piratenpartei, vor allem unter den 15 Mandatsträger_innen im Berliner Abgeordnetenhaus, ist in den Medien 2011 so stark thematisiert worden, dass sich die Mitglieder teilweise mit Geschlechterfragen auseinandersetzen müssen.³ Dies geschieht auf eine Weise, die zu begleiten sehr spannend wäre: In einem Kontext, der sich selbst gerne als „postgender“ bezeichnet und damit suggeriert, Geschlecht sei keine relevante Differenzkategorie mehr, provoziert der Begriff „Feminismus“ starke Abwehrreaktionen. Die Akteur_innen, die die parteiinterne Debatte betreiben, wählen andere Begriffe, um Ungleichheiten und Diskriminierung entlang verschiedener Kategorien zu thematisieren, etwa „Equalismus“⁴ und nennen sich selbst den „Kegelclub“ innerhalb der Piratenpartei, die Plattform für „Weibliches“. Auf Twitter verschlagworten sie ihre Nachrichten selbstironisch mit „genderpopender“. Die Piratenpartei hat das Ziel, den Zwang zum geschlechtseindeutigen Vornamen abzuschaffen und dadurch der geschlechtlichen Selbstverortung von Individuen Raum zu geben. Personen, die sich nicht als Männer oder Frauen definieren, nennen sich „Eichhörnchen“. In welche Richtung sich diese Debatte entwickeln wird, ist offen: Wirkt sich die Verniedlichung geschlechterpolitischer Forderungen entpolitisierend aus? Oder kann es durch die veränderte Wortwahl gelingen, über Diskriminierung und strukturelle soziale Ungleichheiten entlang der Kategorie Geschlecht ins Gespräch zu kommen? Kann das mit Begriff „Equa-

³vgl. die Aufzeichnung einer Diskussion auf der Veranstaltung *Open Mind* im November 2011 (<http://politology.de/2011/11/16/equalismus-bei-den-piraten-auf-der-om11/>)

⁴vgl. <https://wiki.piratenpartei.de/Kegelklub/PositionspapierEqualismus>

lismus“ angestrebte Zusammendenken verschiedener Differenzkategorien in einem liberalen, aber in großen Teilen antifeministischen Umfeld die Akzeptanz für feministische Anliegen erhöhen?

Schließlich ist vor dem Hintergrund dieser Arbeit der Einsatz von Software für die politische Kommunikation und Organisation ein Forschungsthema. Neben zahlreichen anderen Werkzeugen verwendet die Piratenpartei eine freie Software, *Liquid Feedback* zu parteiinternen Meinungsbildung, Entscheidungsfindung und Stimmendelegation. Damit erprobt sie neue Möglichkeiten, Transparenz herzustellen, Partizipation zu erweitern und themenorientiert zu arbeiten. Die Software bringt demnach Elemente von repräsentativer und direkter Demokratie zusammen (vgl. Niedermayer 2010: 848 f.).⁵ Was bedeutet es für diese emanzipatorischen und demokratischen Ansprüche, dass die sozialen und politischen Koordinierungsprozesse in einer Software formalisiert werden? Welche Einschreibungen trägt die Software in sich und wie wirken sie sich auf politische Prozesse aus?

Schließlich möchte ich die Frage aufwerfen, warum keine gesellschaftlichen Debatten über eine *demokratische* Gestaltung von Computertechnologien geführt werden. Soziale und politische Fragen, die im weitesten Sinne mit ‚dem Internet‘ und der Kontrolle über Software zu tun haben, kursieren täglich in den Nachrichten. Dennoch werden kaum Visionen von Alternativen entwickelt, höchstens von einem sehr eingeschränkten Kreis von Akteur_innen. Die vorliegende Arbeit liefert diverse Erklärungsansätze für die ausbleibende Auseinandersetzung. Dennoch halte ich sie für nötig: Welche Kommunikationsinfrastrukturen wollen wir? Wer entwickelt sie, stellt sie der Allgemeinheit zur Verfügung und pflegt sie? Wem überlassen wir durch ausbleibende Einmischung das Feld? Was bedeutet dies in Bezug auf Demokratie, Partizipation, Macht und Differenz? Dies sind essentielle Fragen, die kaum grundsätzlich und breit diskutiert werden. Hier bietet die selbstorganisierte Aneignungspraxis von FLOSS-Communities durchaus Anknüpfungspunkte: Die Community-Mitglieder haben bzw. entwickeln ein Verständnis von Computern als menschengemachten und damit gestaltbaren Maschinen. Sie zeigen, dass es möglich ist, sie sich gemeinsam anzueignen, um sie zu verändern.

⁵vgl. auch das Interview von Tim Pritlove mit dem Entwickler von *Liquid Feedback* Jan Behrens (<http://chaosradio.ccc.de/cre158.html>, Abruf: 05.11.2011)

Literaturverzeichnis

- Acker 2003:** Acker, Joan: The Continuing Necessity of 'Class' in Feminist Thinking. In: Knapp, Gudrun-Axeli (Hrsg.): Achsen der Differenz. Münster: Westfälisches Dampfboot, 2003, S. 49–72
- Adam 1995:** Adam, Alison: Embodying Knowledge. A Feminist Critique of Artificial Intelligence. In: The European Journal of Women's Studies 2 (1995), Nr. 3, S. 355–377
- Adams 1999:** Adams, Douglas: Per Anhalter durch die Galaxis. München: Wilhelm Heyne Verlag, 1999
- Akrich 1992:** Akrich, Madeleine: The De-Description of Technical Objects. In: Bijker, Wiebe (Hrsg.)/Law, John (Hrsg.): Shaping Technology/Building Society. Studies in Sociotechnical Change. Cambridge/Mass.: MIT Press, 1992, S. 205–224
- Akrich 1995:** Akrich, Madeleine: User Representations: Practices, Methods and Sociology. In: Rip, Arie (Hrsg.)/Misa, Thomas (Hrsg.) /Schot, Johan (Hrsg.): Managing Technology in Society. London/New York: Pinter, 1995, S. 167–184
- Akrich/Latour 1992:** Akrich, Madeleine/Latour, Bruno: A Summary of a Convenient Vocabulary for the Semiotics of Human and Nonhuman Assemblies. In: Bijker, Wiebe (Hrsg.)/Law, John (Hrsg.): Shaping Technology/Building Society. Studies in Sociotechnical Change. Cambridge/Mass.: MIT Press, 1992, S. 259–264
- von Alemann et al. 2010:** Alemann, Ulrich von/Erbentraut, Philipp /Walther, Jens: Das Parteiensystem der Bundesrepublik Deutschland. Wiesbaden: VS Verlag für Sozialwissenschaften, 2010
- Amann/Hirschauer 1997:** Amann, Klaus (Hrsg.)/Hirschauer, Stefan (Hrsg.): Die Befremdung der eigenen Kultur. Zur ethnographischen Herausforderung soziologischer Empirie. Frankfurt am Main: Suhrkamp, 1997
- Andersen/Woyke 2009:** Andersen, Uwe (Hrsg.)/Woyke, Wichard (Hrsg.): Handwörterbuch des politischen Systems der Bundesrepublik Deutschland. Wiesbaden: VS Verlag für Sozialwissenschaften, 2009

- Andresen 1987:** Andresen, Sünne: Frauen, Technik und Neue Technologien. Anmerkungen zur Verschiebung der Diskussion von der „weiblichen Technikfeindschaft“ zu geschlechtsspezifischen Zugangsweisen zum Computer. Berlin: Wissenschaftszentrum Berlin für Sozialforschung, 1987
- Atkinson et al. 2002:** Atkinson, Paul (Hrsg.)/Coffey, Amanda (Hrsg.)/Delamont, Sara (Hrsg.)/Lofland, John (Hrsg.) /Lofland, Lyn (Hrsg.): Handbook of Ethnography. Sage Publications, 2002
- Audehm 2001:** Audehm, Kathrin: Die Macht der Sprache. Performative Magie bei Pierre Bourdieu. In: Wulf, Christoph (Hrsg.)/Göhlich, Michael (Hrsg.) /Zirfas, Jörg (Hrsg.): Grundlagen des Performativen. Eine Einführung in die Zusammenhänge von Sprache, Macht und Handeln. Juventa, 2001, S. 101–128
- Baacke 2007:** Baacke, Dieter: Jugend und Jugendkulturen: Darstellung und Deutung. Weinheim: Juventa Verlag, 2007
- Bader 2001:** Bader, Roland: Learning Communities im Internet. Aneignung von Netzkompetenz als gemeinschaftliche Praxis. Eine Fallstudie in der pädagogischen Weiterbildung. Münster: LIT Verlag, 2001
- Bagozzi/Dholakia 2006:** Bagozzi, Richard P./Dholakia, Utpal M.: Open Source Software User Communities: A Study of Participation in Linux User Groups. In: Management Science 52 (2006), Nr. 7, S. 1099–1115
- Barlösius 2006:** Barlösius, Eva: Pierre Bourdieu. Frankfurt am Main: Campus, 2006
- Bath 2002:** Bath, Corinna: Genderforschung in der Informatik: 10 Jahre zurück — 10 Jahre voraus? In: Fiff-Kommunikation (2002), 09, S. 41–46
- Bath 2009:** Bath, Corinna: De-Gendering informatischer Artefakte: Grundlagen einer kritisch-feministischen Technikgestaltung. Version: 2009. <http://elib.suub.uni-bremen.de/edocs/00102741-1.pdf>, Abruf: 17.11.2013
- Bath et al. 2010:** Bath, Corinna/Schelhowe, Heidi /Wiesner, Heike: Informatik: Geschlechteraspekte einer technischen Disziplin. In: Becker, Ruth (Hrsg.)/Kortendiek, Beate (Hrsg.): Handbuch Frauen- und Geschlechterforschung: Theorie, Methoden, Empirie. Wiesbaden: VS Verlag für Sozialwissenschaften, 2010, S. 821–833

-
- Beaufaÿs 2003:** Beaufaÿs, Sandra: Wie werden Wissenschaftler gemacht? Beobachtungen zur wechselseitigen Konstitution von Geschlecht und Wissenschaft. Bielefeld: transcript, 2003
- Becker-Schmidt 2003:** Becker-Schmidt, Regina: Zur doppelten Vergesellschaftung von Frauen. Soziologische Grundlegung, empirische Rekonstruktion. In: *gender...politik...online* (2003). http://www.fu-berlin.de/sites/gpo/soz_eth/Geschlecht_als_Kategorie/Die_doppelte_Vergesellschaftung_von_Frauen/becker_schmidt_ohne.pdf, Abruf: 26.10.2013
- Becker-Schmidt 2007:** Becker-Schmidt, Regina: Geschlechterdifferenz — Geschlechterverhältnis: Soziale Dimensionen des Begriffs „Geschlecht“. In: Hark, Sabine (Hrsg.): *Dis/Kontinuitäten: Feministische Theorie*. VS Verlag für Sozialwissenschaften, 2007, S. 115–127
- Beckert-Zieglschmid 2005:** Beckert-Zieglschmid, Claudia: „Der Apfel fällt nicht weit vom Stamm“? Eine Anwendung der Theorie von Pierre Bourdieu auf Lebensstile und Ernährungspraxis Jugendlicher. Norderstedt: Books on Demand GmbH, 2005
- Behnke/Meuser 1999:** Behnke, Cornelia/Meuser, Michael: *Geschlechterforschung und qualitative Methoden*. Opladen: Leske + Budrich, 1999
- Behnke et al. 2006:** Behnke, Joachim (Hrsg.)/Gschwend, Thomas (Hrsg.)/Schindler, Delia (Hrsg.) /Schnapp, Kai-Uwe (Hrsg.): *Methoden der Politikwissenschaft. Neuere qualitative und quantitative Analyseverfahren*. Baden-Baden: Nomos, 2006
- Belliger/Krieger 2006:** Belliger, Andréa/Krieger, David J.: Einführung in die Akteur-Netzwerk-Theorie. In: Belliger, Andréa (Hrsg.)/Krieger, David J. (Hrsg.): *ANThology*. Ein einführendes Handbuch zur Akteur-Netzwerk-Theorie. Bielefeld: transcript, 2006, S. 13–50
- Bendrath 2001:** Bendrath, Ralf: PGP — die ersten zehn Jahre. Version: 2001. <http://www.heise.de/tp/r4/artikel/7/7175/1.html>, Abruf: 26.10.2013
- Benkler 2002:** Benkler, Yochai: Coase's Penguin, or, Linux and "The Nature of the Firm". In: *The Yale Law Journal* 112 (2002), Nr. 3, S. 369–446. <http://www.jstor.org/stable/1562247>, Abruf: 26.10.2013
- Berdou 2007:** Berdou, Evangelia: *Managing the Bazaar: Commercialization and peripheral participation in mature, community-led Free/Open source software projects*.

Version: 2007. http://www2.lse.ac.uk/media@lse/study/pdf/PhD_Berdou_2007.pdf, Ab-
ruf: 26.10.2013

Berg/Milmeister 2007: Berg, Charles/Milmeister, Marianne: Im Dialog mit den Daten das eigene Erzählen der Geschichte finden. über die Kodierverfahren der Grounded Theory Methodologie. In: *Historical Social Research* 19 (2007), S. 182–210

Berg/Fuchs 1993a: Berg, Eberhard (Hrsg.)/Fuchs, Martin (Hrsg.): *Kultur, soziale Praxis, Text. Die Krise der ethnographischen Repräsentation*. Suhrkamp, 1993

Berg/Fuchs 1993b: Berg, Eberhard/Fuchs, Martin: *Phänomenologie der Differenz. Reflexionsstufen ethnographischer Repräsentation*. In: Berg, Eberhard (Hrsg.)/Fuchs, Martin (Hrsg.): *Kultur, soziale Praxis, Text. Die Krise der ethnographischen Repräsentation*. Frankfurt am Main: Suhrkamp, 1993, S. 11–108

von Beyme 2010: Beyme, Klaus von: *Das politische System der Bundesrepublik Deutschland. Eine Einführung*. Wiesbaden: VS Verlag für Sozialwissenschaften, 2010

Bezkourov 1999: Bezkourov, Nikolai: A Second Look at the Cathedral and the Bazaar. In: *First Monday* 4 (1999), Nr. 12. <http://dx.doi.org/10.5210/fm.v4i12.708>, Ab-
ruf: 26.10.2013.
– DOI 10.5210/fm.v4i12.708

Blasius/Winkler 1989: Blasius, Jörg/Winkler, Joachim: Gibt es die "feinen Unterschiede"? Eine empirische Überprüfung der Bourdieuschen Theorie. In: *Kölner Zeitschrift für Soziologie und Sozialpsychologie* 41 (1989), S. 72–94

Blazek 1999: Blazek, Helmut: *Männerbünde. Eine Geschichte von Faszination und Macht*. Berlin: Links Verlag, 1999

Bohn 1991: Bohn, Cornelia: *Habitus und Kontext. Ein kritischer Beitrag zur Sozialtheorie Bourdieus*. Opladen: Westdeutscher Verlag, 1991

Bohnsack 2000: Bohnsack, Ralf: *Rekonstruktive Sozialforschung. Einführung in Methodologie und Praxis qualitativer Forschung*. Opladen: Leske + Budrich, 2000

Bohnsack et al. 2003: Bohnsack, Ralf (Hrsg.)/Marotzki, Winfried (Hrsg.) /Meuser, Michael (Hrsg.): *Hauptbegriffe Qualitative Sozialforschung. Ein Wörterbuch*. Opladen: Leske + Budrich, 2003

-
- Bolstad 2006:** Bolstad, Sverre H.: Learning and knowledge in FLOSS. Situated learning and organizational knowledge-conversion in community-based free/libre open source software development. (2006). <http://flosshub.org/sites/flosshub.org/files/Learning-and-knowledge-in-FLOSS.pdf>, Abruf: 26.10.2013
- Bonaccorsi/Rossi 2006:** Bonaccorsi, Andrea/Rossi, Cristina: Comparing Motivations of Individual Programmers and Firms to Take Part in the Open Source Movement: From Community to Business. In: Knowledge, Technology, & Policy 18 (2006), Nr. 4, S. 40–64
- Bourdieu 1982:** Bourdieu, Pierre: Les rites d’institution. In: Actes de la recherche en sciences sociales 43 (1982), S. 8–63
- Bourdieu 1983:** Bourdieu, Pierre: Ökonomisches Kapital, kulturelles Kapital, soziales Kapital. In: Kreckel, Reinhard (Hrsg.): Soziale Ungleichheiten (Soziale Welt: Sonderband 2). Göttingen: Verlag Otto Schwartz & Co., 1983, S. 183–198
- Bourdieu 1985:** Bourdieu, Pierre: Sozialer Raum und Klassen. 2 Vorlesungen. Frankfurt am Main: Suhrkamp, 1985
- Bourdieu 1987:** Bourdieu, Pierre: Die feinen Unterschiede. Kritik der gesellschaftlichen Urteilskraft. Frankfurt am Main: Suhrkamp, 1987
- Bourdieu 1988:** Bourdieu, Pierre: Homo academicus. Frankfurt am Main: Suhrkamp, 1988
- Bourdieu 1990:** Bourdieu, Pierre: Was heißt Sprechen? Die Ökonomie des sprachlichen Tauschs. Wien: Braumüller, 1990
- Bourdieu 1992:** Bourdieu, Pierre ; Steinrück, Margareta (Hrsg.): Die verborgenen Mechanismen der Macht. Schriften zu Politik & Kultur. Hamburg: VSA-Verlag, 1992
- Bourdieu 1993:** Bourdieu, Pierre: Soziologische Fragen. Frankfurt am Main: Suhrkamp, 1993
- Bourdieu 1997a:** Bourdieu, Pierre: Meditationen. Zur Kritik der scholastischen Vernunft. Frankfurt am Main: Suhrkamp, 1997
- Bourdieu 1997b:** Bourdieu, Pierre: Der Tote packt den Lebenden. Schriften zu Politik und Kultur 2. Hamburg: VSA-Verlag, 1997
- Bourdieu 1998a:** Bourdieu, Pierre: Gegenfeuer. Wortmeldungen im Dienste des Widerstands gegen die neoliberale Invasion. UVK Universitätsverlag Konstanz, 1998

- Bourdieu 1998b:** Bourdieu, Pierre: *Praktische Vernunft. Zur Theorie des Handelns*. Frankfurt am Main: Suhrkamp, 1998
- Bourdieu 1999:** Bourdieu, Pierre: *Sozialer Sinn. Kritik der theoretischen Vernunft*. 3. Aufl. Frankfurt am Main: Suhrkamp, 1999
- Bourdieu 2000:** Bourdieu, Pierre: *Propos sur le champ politique*. Lyon: Presses Universitaires de Lyon, 2000
- Bourdieu 2004:** Bourdieu, Pierre: *Teilnehmende Objektivierung*. In: Ohnacker, Elke (Hrsg.)/Schultheis, Franz (Hrsg.): *Schwierige Interdisziplinarität*. Münster: Westfälisches Dampfboot, 2004, S. 172–186
- Bourdieu 2005a:** Bourdieu, Pierre: *Die männliche Herrschaft*. Frankfurt am Main: Suhrkamp, 2005
- Bourdieu 2005b:** Bourdieu, Pierre: *The social structures of the economy*. Cambridge/Malden: Polity Press, 2005
- Brand/Schmid 2006:** Brand, Andreas/Schmid, Alfons: *Ist ein Open-Source-Projekt nur kooperativ? Die Koordination der Zusammenarbeit im KDE-Projekt*. Version: 2006. http://www.opensourcejahrbuch.de/download/jb2006/chapter_03/osjb2006-03-04-brandschmid.pdf, Abruf: 26.10.2013. In: Lutterbeck, Bernd (Hrsg.)/Bärwolff, Matthias (Hrsg.) /Gehring, Robert A. (Hrsg.): *Open Source Jahrbuch 2006. Zwischen Softwareentwicklung und Gesellschaftsmodell*. 2006, S. 123–137
- Braun-Thürmann 2006:** Braun-Thürmann, Holger: *Ethnografische Perspektiven: Technische Artefakte in ihrer symbolisch-kommunikativen und praktische-materiellen Dimension*. In: Rammert, Werner (Hrsg.)/Schubert, Cornelius (Hrsg.): *Technografie. Zur Mikrosoziologie der Technik*. Frankfurt am Main/New York: Campus, 2006, S. 199–221
- Breidenstein 2006:** Breidenstein, Georg: *Teilnahme am Unterricht. Ethnographische Studien zum Schülerjob*. Wiesbaden: VS Verlag für Sozialwissenschaften, 2006
- Bremer 2007:** Bremer, Helmut: *Soziale Milieus, Habitus und Lernen. Zur sozialen Selektivität des Bildungswesen am Beispiel der Weiterbildung*. Weinheim/München: Juventa Verlag, 2007

-
- Brinkmann/Seifert 2001:** Brinkmann, Ulrich/Seifert, Matthias: „Face to Interface“: Zum Problem der Vertrauenskonstitution im Internet am Beispiel von elektronischen Auktionen. In: Zeitschrift für Soziologie 30 (2001), Nr. 1, S. 23–47
- Brunst/Erben 2008:** Brunst, Bettina/Erben, Meinhard: IT-Compliance — Haftungsrisiken für Unternehmen und Geschäftsführung. In: Information Management & Consulting (2008), Nr. 3, S. 25–31
- Buchstein 1996a:** Buchstein, Hubertus: Bittere Bytes: Cyberbürger und Demokratietheorie. In: Deutsche Zeitschrift für Philosophie 44 (1996), S. 583–607
- Buchstein 1996b:** Buchstein, Hubertus: Virtuelle Demokratie. Zum Verhältnis von Internet und Demokratie. In: Die Neue Gesellschaft/Frankfurter Hefte 43 (1996), S. 165–170
- Buchstein 2001:** Buchstein, Hubertus: Modernisierung der Demokratie durch e-Voting? In: Leviathan 29 (2001), S. 147–155
- Burchardt 2004:** Burchardt, Hans-Jürgen: Zeitenwende. Politik nach dem Neoliberalismus. Stuttgart: Schmetterling Verlag, 2004
- Burzan 2007:** Burzan, Nicole: Soziale Ungleichheit. Eine Einführung in die zentralen Theorien. Wiesbaden: VS Verlag für Sozialwissenschaften, 2007
- Callon/Latour 2006:** Callon, Michel/Latour, Bruno: Die Demontage des großen Leviathans: Wie Akteure die Makrostruktur der Realität bestimmen und Soziologen ihnen dabei helfen. In: Belliger, Andréa (Hrsg.)/Krieger, David J. (Hrsg.): ANThology. Ein einführendes Handbuch zur Akteur-Netzwerk-Theorie. Bielefeld: transcript, 2006, S. 75–101
- Carroll 2007:** Carroll, J. M.: Learning In Communities: Introduction to the Special Issue. In: Computer Supported Cooperative Work (CSCW) 16 (2007), Nr. 4-5, S. 373–374
- Castells 2002:** Castells, Manuel: Die Macht der Identität. Teil 2 der Trilogie Das Informationszeitalter. Opladen: Leske + Budrich, 2002
- Castells 2003:** Castells, Manuel: Der Aufstieg der Netzwerkgesellschaft. Teil 1 der Trilogie Das Informationszeitalter. Opladen: Leske + Budrich, 2003
- Castells 2005:** Castells, Manuel: Die Internet-Galaxie. Internet, Wirtschaft und Gesellschaft. VS Verlag für Sozialwissenschaften, 2005

- Clifford 1993:** Clifford, James: Über ethnographische Autorität. In: Berg, Eberhard (Hrsg.)/Fuchs, Martin (Hrsg.): Kultur, soziale Praxis, Text. Die Krise der ethnographischen Repräsentation. Frankfurt am Main: Suhrkamp, 1993, S. 109–157
- Clifford/Marcus 1986:** Clifford, James (Hrsg.)/Marcus, George E. (Hrsg.): Writing Culture. The Poetics and Politics of Ethnography. Berkeley, Los Angeles, London: University of California Press, 1986
- Cockburn 1988:** Cockburn, Cynthia: Die Herrschaftsmaschine. Geschlechterverhältnisse und technisches Know-how. Berlin: Argument, 1988
- Cockburn/Ormrod 1993:** Cockburn, Cynthia/Ormrod, Susan: Gender & Technology in the Making. London/Thousand Oaks/New Delhi: Sage Publications, 1993
- Coleman 2004:** Coleman, Gabriella: The Political Agnosticism of Free and Open Source Software and the Inadvertent Politics of Contrast. In: Anthropological Quarterly 77 (2004), Nr. 3, S. 507–519
- Coleman 2005:** Coleman, Gabriella: The Social Construction of Freedom in Free and Open Source Software: Hackers, Ethics, and the Liberal Tradition. Chicago, University of Chicago, Diss., 2005
- Coleman 2009:** Coleman, Gabriella: Code is speech: Legal Tinkering, Expertise, and Protest among Free and Open Source Software Developers. In: Cultural Anthropology 24 (2009), Nr. 3, S. 420–454
- Coleman 2010:** Coleman, Gabriella: The Hacker Conference. A Ritual Condensation and Celebration of a Lifeworld. In: Anthropological Quarterly 83 (2010), Nr. 1, S. 47–72
- Coleman 2011:** Coleman, Gabriella: Anonymous: From the Lulz to Collective Action. Version: 2011. <http://mediacommons.futureofthebook.org/tne/pieces/anonymous-lulz-collective-action>, Abruf: 26.10.2013
- Coleman/Golub 2008:** Coleman, Gabriella/Golub, Alex: Hacker practice. Moral genres and the cultural articulation of liberalism. In: Anthropological Theory 8 (2008), Nr. 3, S. 255–277
- Coleman/Hill 2005:** Coleman, Gabriella/Hill, Benjamin: The Social Production of Ethics in Debian and Free Software Communities: Anthropological Lessons for Vocational

Ethics. In: Koch, Stefan (Hrsg.): Free and Open Source Software Development. Idea Group, 2005, S. 273–295

Connell/Messerschmidt 2005: Connell, Robert W./Messerschmidt, James W.: Hegemonic Masculinity: Rethinking the Concept. In: Gender & Society 19 (2005), Nr. 6, S. 829–859

Crapanzano 1986: Crapanzano, Vincent: Hermes' Dilemma: The Masking of Subversion in Ethnographic Description. In: Writing Culture. The Poetics and Politics of Ethnography. Berkeley, Los Angeles, London: University of California Press, 1986, S. 51–76

Crowston/Howison 2004: Crowston, Kevin/Howison, James: The social structure of Free and Open Source software development. Version: 2004. <http://flosshub.org/sites/flosshub.org/files/crowstonhowison.pdf>, Abruf: 26.10.2013

Crowston/Howison 2006: Crowston, Kevin/Howison, James: Hierarchy and Centralization in Free and Open Source Software Team Communications. In: Knowledge, Technology, & Policy 18 (2006), Nr. 4, S. 65–85

Crowston et al. 2007: Crowston, Kevin/Howison, James/Masango, Chengetai /Eseryel, U. Y.: The role of face-to-face meetings in technology-supported self-organizing distributed teams. In: IEEE Transactions on Professional Communications 50 (2007), Nr. 3, S. 185–203

Crutzen 2007: Crutzen, Cecile: Das Unsichtbare und das Sichtbare des artifiziellen Handelns im täglichen Leben. In: Kurswechsel. Zeitschrift für gesellschafts-, wirtschafts- und umweltpolitische Alternativen (2007), Nr. 3, S. 35–46

Davis 2008: Davis, Kathy: Intersectionality as buzzword. In: Feminist Theory 9 (2008), Nr. 1, S. 67–85

Deegan 2002: Deegan, Mary J.: The Chicago School of Ethnography. In: Atkinson, Paul (Hrsg.)/Coffey, Amanda (Hrsg.)/Delamont, Sara (Hrsg.)/Lofland, John (Hrsg.) /Lofland, Lyn (Hrsg.): Handbook of Ethnography. London, Thousand Oaks, New Delhi: Sage Publications, 2002, S. 11–25

Degele 2002: Degele, Nina: Einführung in die Techniksoziologie. München: Wilhelm Fink Verlag, 2002

- Degele/Winker 2007:** Degele, Nina/Winker, Gabriele: Intersektionalität als Mehrebenenanalyse. Version: 2007. http://www.tu-harburg.de/agentec/winker/pdf/Intersektionalitaet_Mehrebenen.pdf, Abruf: 26.10.2013
- Degele/Winker 2009:** Degele, Nina/Winker, Gabriele: Intersektionalität. Zur Analyse sozialer Ungleichheiten. Bielefeld: transcript, 2009
- Diaz-Bone 2002:** Diaz-Bone, Rainer: Kulturwelt, Diskurs und Lebensstil. Eine diskurstheoretische Erweiterung der bourdieuschen Distinktionstheorie. Opladen: Leske + Budrich, 2002
- Dinkelaker 2010:** Dinkelaker, Jörg: Koordination von Körpern. Eine vernachlässigte Dimension pädagogischer Professionalität. In: Hof, Christiane (Hrsg.)/Ludwig, Joachim (Hrsg.) /Schäffer, Burkhard (Hrsg.): Professionalität zwischen Praxis, Politik und Disziplin. Baltmannsweiler: Schneider Verlag, 2010, S. 186–202
- Drahos/Braithwaite 2002:** Drahos, Peter/Braithwaite, John: Information Feudalism: Who Owns the Knowledge Economy? London: Earthscan Publications, 2002
- Düx/Sass 2005:** Düx, Wiebken/Sass, Erich: Lernen in informellen Kontexten. Lernpotenziale in Settings des freiwilligen Engagements. In: Zeitschrift für Erziehungswissenschaft (2005), Nr. 3, S. 394–411
- Ebrecht 2004:** Ebrecht, Jörg: Die Kreativität der Praxis. Überlegungen zum Wandel von Habitusformationen. In: Bourdieus Theorie der Praxis. Erklärungskraft — Anwendung — Perspektiven. Wiesbaden: VS Verlag für Sozialwissenschaften, 2004, S. 225–241
- Ebrecht/Hillebrandt 2004:** Ebrecht, Jörg (Hrsg.)/Hillebrandt, Frank (Hrsg.): Bourdieus Theorie der Praxis. Erklärungskraft — Anwendung — Perspektiven. Wiesbaden: VS Verlag für Sozialwissenschaften, 2004
- Eckenfels 2012:** Eckenfels, Mela: Alle Macht dem Volke. Vier Liquid-Democracy-Tools im Test. In: Linux-Magazin (2012), Nr. 01, S. 68–72
- Eder 1989:** Eder, Klaus (Hrsg.): Klassenlage, Lebensstil und kulturelle Praxis: Beiträge zur Auseinandersetzung mit Pierre Bourdieus Klassentheorie. Frankfurt am Main: Suhrkamp, 1989
- Elter 2010:** Elter, Andreas: Bierzelt oder Blog? Politik im digitalen Zeitalter. Hamburg: Hamburger Edition, 2010

-
- Emerson et al. 1995:** Emerson, Robert M./Fretz, Rachel I. /Shaw, Linda L.: Writing ethnographic fieldnotes. Chicago/London: The University of Chicago Press, 1995
- Engler 1993:** Engler, Steffani: Fachkultur, Geschlecht und soziale Reproduktion. Eine Untersuchung über Studentinnen und Studenten der Erziehungswissenschaft, Rechtswissenschaft, Elektrotechnik und des Maschinenbaus. Weinheim: Deutscher Studienverlag, 1993
- Ess 1996:** Ess, Charles: The Political Computer: Democracy, CMC, and Habermas. In: Ess, Charles (Hrsg.): Philosophical perspectives on computer-mediated communication. State University of New York Press, 1996, S. 197–230
- Fabian 1993:** Fabian, Johannes: Präsenz und Repräsentation. Die Anderen und das anthropologische Schreiben. In: Kultur, soziale Praxis, Text. Die Krise der ethnographischen Repräsentation. Frankfurt am Main: Suhrkamp, 1993, S. 335–364
- Feilner/Kames 2009:** Feilner, Markus/Kames, Bastian: Verwaltungskram. LDAP-Administrations-Clients im Vergleich. In: Linux-Magazin (2009), Nr. 5. <http://www.linux-magazin.de/Heft-Abo/Ausgaben/2009/05/Verwaltungskram>, Abruf: 26.10.2013
- Feller et al. 2005:** Feller, Joseph (Hrsg.)/Fitzgerald, Brian (Hrsg.)/Hissam, Scott A. (Hrsg.)/Lakhani, Karim L. (Hrsg.): Perspectives on Free and Open Source Software. Cambridge/Mass.: The MIT Press, 2005
- Flick 1992:** Flick, Uwe: Entzauberung der Intuition. Systematische Perspektiven-Triangulation als Strategie der Geltungsbegründung qualitativer Daten und Interpretationen. In: Hoffmeyer-Zlotnik, Jürgen H. P. (Hrsg.): Analyse verbaler Daten. Opladen: Westdeutscher Verlag, 1992, S. 11–55
- Flick 1999:** Flick, Uwe: Qualitative Forschung. Theorien, Methoden, Anwendung in Psychologie und Sozialwissenschaften. Reinbek bei Hamburg: Rowohlt, 1999
- Flick 2002:** Flick, Uwe: Qualitative Sozialforschung. Eine Einführung. Reinbek bei Hamburg: Rowohlt, 2002
- Foucault 1976a:** Foucault, Michel: Mikrophysik der Macht. Über Strafjustiz, Psychiatrie und Medizin. Berlin: Merve-Verlag, 1976
- Foucault 1976b:** Foucault, Michel: Überwachen und Strafen. Die Geburt des Gefängnisses. Frankfurt am Main: Suhrkamp, 1976

- Foucault 1977:** Foucault, Michel: Der Wille zum Wissen. Sexualität und Wahrheit I. Frankfurt am Main: Suhrkamp, 1977
- Frankenberg 1996:** Frankenberg, Ruth: Weiße Frauen, Feminismus und die Herausforderung des Antirassismus. In: Fuchs, Brigitte (Hrsg.)/Habinger, Gabriele (Hrsg.): Rassismen & Feminismen. Differenzen, Machtverhältnisse und Solidarität zwischen Frauen. Wien: Promedia, 1996, S. 51–66
- Fried 2008:** Fried, Andri: Gemeinschaftsbildung bei Freier/Libre und Open Source Software? Studie über die Ubuntu-GNU/Linux Community ubuntuusers.de. Potsdam: unveröff. Magisterarbeit an der Wirtschafts- und Sozialwissenschaftlichen Fakultät der Universität Potsdam, 2008
- Friedman/Friedman 2003:** Friedman, Linda W./Friedman, Hershey H.: A framework for the study of computer-oriented humor (COHUM). Version: 2003. <http://cisnet.baruch.cuny.edu/papers/cis200301.pdf>, Abruf: 26.10.2013
- Friedrichs/Lüdtke 1977:** Friedrichs, Jürgen/Lüdtke, Hartmut: Teilnehmende Beobachtung. Einführung in die sozialwissenschaftliche Feldforschung. Weinheim/Basel: Beltz Verlag, 1977
- Fritzsche 2003:** Fritzsche, Bettina: Pop-Fans. Studie einer Mädchenkultur. Opladen: Leske + Budrich, 2003
- Fröhlich 1994:** Fröhlich, Gerhard: Kapital, Habitus, Feld, Symbol. Grundbegriffe der Kulturtheorie bei Pierre Bourdieu. In: Mörth, Ingo (Hrsg.)/Fröhlich, Gerhard (Hrsg.): Das symbolische Kapital der Lebensstile. Zur Kulturosoziologie der Moderne nach Pierre Bourdieu. Frankfurt am Main/New York: Campus, 1994, S. 31–54
- Gaboriau/Gaboriau 2006:** Gaboriau, Patrick/Gaboriau, Philippe: Bernard Lahire. La Culture des Individus. Dissonances culturelles et distinction de soi. In: L'Homme janvier - juin (2006), S. 177–178. <http://lhomme.revues.org/index2313.html>, Abruf: 26.10.2013
- Ganten/Alex 2004:** Ganten, Peter H./Alex, Wulf: Debian GNU/Linux. Berlin/Heidelberg/New York: Springer, 2004
- Ganz 2010:** Ganz, Kathrin: Netzpolitik und soziale Gerechtigkeit. Ein queer-feministisch-intersektional inspirierter Blick auf die Themen der digitalen Bürgerrechtsbewegung. Version: 2010. <http://www.feministisches-institut.de/wp-content/uploads/2010/02/netzpolitik.pdf>, Abruf: 26.10.2013

-
- Gebauer/Wulf 1993:** Gebauer, Gunter (Hrsg.)/Wulf, Christoph (Hrsg.): Praxis und Ästhetik. Neue Perspektiven im Denken Pierre Bourdieus. Frankfurt am Main: Suhrkamp, 1993
- Geertz 1987:** Geertz, Clifford: Dichte Beschreibung. Beiträge zum Verstehen kultureller Systeme. Frankfurt am Main: Suhrkamp, 1987
- Geertz 1993:** Geertz, Clifford: Die künstlichen Wilden. Der Anthropologe als Schriftsteller. Frankfurt am Main: Fischer Taschenbuch Verlag, 1993
- Ghosh/Glott 2005:** Ghosh, Rishab A./Glott, Ruediger: Skills Survey Interim Report. (2005). http://flosspols.org/deliverables/FLOSSPOLS-D10-skills%20survey_interim_report-revision-FINAL.pdf, Abruf: 26.10.2013
- Ghosh et al. 2002:** Ghosh, Rishab A./Glott, Rüdiger/Krieger, Bernhard /Robles, Gregorio: Free/Libre and Open Source Software: Survey and Study. Part 4: Survey of Developers. Version: 2002. http://www.flossproject.org/report/FLOSS_Final4.pdf, Abruf: 26.10.2013
- Ghosh 2005:** Ghosh, Rishab A.: Why Collaboration is Important (Again). In: Ghosh, Rishab A. (Hrsg.): CODE. Collaborative Ownership and the Digital Economy. The MIT Press, 2005, S. 1–11
- Giacquinta et al. 1993:** Giacquinta, Joseph B./Bauer, Jo A. /Levin, Jane E.: Beyond technology's promise: An examination of children's educational computing at home. New York: Cambridge University Press, 1993
- Gianetti 2004:** Gianetti, Claudia: Ästhetik des Digitalen. Ein intermediärer Beitrag zu Wissenschaft, Medien- und Kunstsystemen. Wien/New York: Springer-Verlag, 2004
- Gildemeister 2007:** Gildemeister, Regine: Die soziale Konstruktion von Geschlechtlichkeit. In: Hark, Sabine (Hrsg.): Dis/Kontinuitäten: Feministische Theorie. VS Verlag für Sozialwissenschaften, 2007, S. 55–72
- Gildemeister/Wetterer 1992:** Gildemeister, Regine/Wetterer, Angelika: Wie Geschlechter gemacht werden. Die soziale Konstruktion der Zweigeschlechtlichkeit und ihre Reifizierung in der Frauenforschung. In: Knapp, Gudrun-Axeli (Hrsg.): TraditionenBrüche. Freiburg: Kore, 1992, S. 201–254
- Glaser/Strauss 2005:** Glaser, Barney G./Strauss, Anselm L.: Grounded Theory. Strategien qualitativer Forschung. Bern: Huber, 2005

- Glott et al. 2007:** Glott, Rüdiger/Meiszner, Andreas /Sowe, Sulayman K.: FLOSSCom - Using the Principles of Informal Learning Environments of FLOSS Communities to Improve ICT Supported Formal Education. Version: 2007. http://opensource.mit.edu/papers/FLOSSCom_WP2_Phase_1_Report_v070709_1.pdf, Abruf: 20.02.2011
- Goffman 1967:** Goffman, Erving: Stigma. Über Techniken der Bewältigung beschädigter Identität. Frankfurt am Main: Suhrkamp, 1967
- Goffman 2001:** Goffman, Erving: Interaktion und Geschlecht. Frankfurt am Main: Campus Verlag, 2001
- Gottschall 1995:** Gottschall, Karin: Geschlechterverhältnis und Arbeitsmarktsegregation. In: Becker-Schmidt, Regina (Hrsg.)/Knapp, Gudrun-Axeli (Hrsg.): Das Geschlechterverhältnis als Gegenstand der Sozialwissenschaften. Frankfurt am Main: Campus, 1995, S. 125–162
- Grassmuck 2004:** Grassmuck, Volker: Freie Software. Zwischen Privat- und Gemeineigentum. Bonn: Bundeszentrale für politische Bildung, 2004
- Gröndahl 2001:** Gröndahl, Boris: The Script Kiddies Are Not Alright. Abgrenzungen und Differenzierungen unter Hackern. Version: 2001. <http://www.heise.de/tp/artikel/9/9266/1.html>, Abruf: 26.10.2013
- Göhlich/Zirfas 2007:** Göhlich, Michael/Zirfas, Jörg: Lernen. Ein pädagogischer Grundbegriff. Stuttgart: W. Kohlhammer, 2007
- Göktürk/Çetin 2007:** Göktürk, Mehmet/Çetin, Görkem: Out of Box Experience Issues of Free and Open Source Software. In: Jacko, Julie A. (Hrsg.): Human-Computer Interaction. Interaction Design and Usability. Berlin/Heidelberg: Springer-Verlag, 2007, S. 774–783
- Hagemann-White 2007:** Hagemann-White, Carol: Wir werden nicht zweigeschlechtlich geboren. . . . In: Hark, Sabine (Hrsg.): Dis/Kontinuitäten: Feministische Theorie. VS Verlag für Sozialwissenschaften, 2007, S. 27–37
- Hammersley/Atkinson 1983:** Hammersley, Martyn/Atkinson, Paul: Ethnography. Principles in Practice. London/New York: Tavistock Publications, 1983
- Haralanova 2010:** Haralanova, Christina: L'apport des femmes dans le développement du logiciel libre. Montréal: Abschlussarbeit an der Universität Québec, 2010 http://www.ludost.org/sites/www.ludost.org/files/memhark_complete_0.pdf

-
- Haraway 1995:** Haraway, Donna: Ein Manifest für Cyborgs. Feminismus im Streit mit den Technowissenschaften. In: Hammer, Carmen (Hrsg.)/Stieß, Immanuel (Hrsg.): Die Neuerfindung der Natur. Primaten, Cyborgs und Frauen. Frankfurt am Main/New York: Campus, 1995, S. 33–72
- Harding 1991:** Harding, Sandra: Feministische Wissenschaftstheorie : Zum Verhältnis von Wissenschaft und sozialem Geschlecht. Hamburg: Argument, 1991
- Harding 2006:** Harding, Sandra: Science and Social Inequality. Feminist and Postcolonial Issues. Urbana/Chicago: University of Illinois Press, 2006
- Hark 2005:** Hark, Sabine: Dissidente Partizipation. Frankfurt am Main: Suhrkamp, 2005
- Hark 2007:** Hark, Sabine (Hrsg.): Dis/Kontinuitäten: Feministische Theorie. Wiesbaden: VS Verlag für Sozialwissenschaften, 2007
- Haug 2003:** Haug, Wolfgang F.: High-Tech-Kapitalismus. Analysen zu Produktionsweise, Arbeit, Sexualität, Krieg und Hegemonie. Hamburg: Argument, 2003
- Heinlein 2009:** Heinlein, Peer: LPIC-1. Vorbereitung auf die Prüfung des Linux Professional Institute. München: Open Source Press, 2009
- Heintz 1993:** Heintz, Bettina: Die Herrschaft der Regel. Zur Grundlagengeschichte des Computers. Frankfurt am Main: Campus, 1993
- Heller/Nuss 2004:** Heller, Lydia/Nuss, Sabine: Open Source im Kapitalismus: Gute Idee — falsches System? Version: 2004. http://www.opensourcejahrbuch.de/download/jb2004/chapter_05/V-5-HellerNuss.pdf, Abruf: 26.10.2013. In: Lutterbeck, Bernd (Hrsg.)/Gehring, Robert A. (Hrsg.): Open Source Jahrbuch 2004. Zwischen Softwareentwicklung und Gesellschaftsmodell. 2004, S. 385–405
- Herring 1997:** Herring, Susan C.: The Rhetorical Dynamics of Gender Harrassment On-Line. In: The Information Society 15 (1997), Nr. 3, S. 151–167
- Hilgers 2009:** Hilgers, Mathieu: Habitus, Freedom, and Reflexivity. In: Theory & Psychology (2009), Nr. 19, S. 728–755
- Hillebrandt 2004:** Hillebrandt, Frank: Die verborgenen Mechanismen der Materialität. Überlegungen zu einer Praxistheorie der Technik. In: Bourdieus Theorie der Praxis. Erklärungskraft — Anwendung — Perspektiven. Wiesbaden: VS Verlag für Sozialwissenschaften, 2004, S. 19–45

- von Hippel 2005:** Hippel, Eric von: Democratizing Innovation. 2005 <http://web.mit.edu/evhippel/www/books/DI/DemocInn.pdf>
- Hitzler et al. 2001:** Hitzler, Ronald/Bucher, Thomas /Niederbacher, Arne: Leben in Szenen. Formen jugendlicher Vergemeinschaftung heute. Opladen: Leske + Budrich, 2001
- Hitzler/Pfadenhauer 2006:** Hitzler, Ronald/Pfadenhauer, Michaela: Bildung in der Gemeinschaft. Zur Erfassung der Kompetenzaneignung in Jugendszenen. In: Tully, Claus J. (Hrsg.): Lernen in flexibilisierten Welten. Wie sich das Lernen der Jugend verändert. Weinheim/München: Juventa, 2006, S. 237–254
- Hoffmann 1987:** Hoffmann, Ute: Computerfrauen. Welchen Anteil haben Frauen an Computergeschichte und -arbeit? München: Rainer Hampp Verlag, 1987
- Hoffmeyer-Zlotnik 1992:** Hoffmeyer-Zlotnik, Jürgen H. P. (Hrsg.): Analyse verbaler Daten. über den Umgang mit qualitativen Daten. Opladen: Westdeutscher Verlag, 1992
- Hofmann 1997:** Hofmann, Jeanette: Über Nutzerbilder in Textverarbeitungsprogrammen — Drei Fallbeispiele. In: Dierkes, Meinolf (Hrsg.): Technikgenese. Befunde aus einem Forschungsprogramm. Berlin: Edition Sigma, 1997, S. 71–97
- Hofmann 1999:** Hofmann, Jeanette: Writers, Texts and Writing Acts: Gendered User Images in Word Processing Software. In: Wajcman, Judy (Hrsg.)/MacKenzie, Donald (Hrsg.): The Social Shaping of Technology. Buckingham/Philadelphia: Open University Press, 1999, S. 222–243
- Holzcamp 1995:** Holzcamp, Klaus: Lernen. Subjektwissenschaftliche Grundlegung. Frankfurt am Main/New York: Campus, 1995
- Horstmann 1998:** Horstmann, Alexander: Zwischen Verklärung und Erwachen: Zur Konstruktion von Wirklichkeit in ethnologischen Beiträgen mit Fallbeispielen aus der Literatur und Südthailand. In: Südostasien-Gesellschaft, SEACOM (Hrsg.): Zwischen Verklärung und Erwachen. Eine kritische Betrachtung ethnographischer Beiträge zur Südostasienkunde. Berlin: SEACOM Edition, 1998, S. 6–16
- Hradil 1999:** Hradil, Stefan: Soziale Ungleichheit in Deutschland. Opladen: Leske + Budrich, 1999

-
- Håpnes/Rasmussen 1991:** Håpnes, Tove/Rasmussen, Bente: The Production of Male Power in Computer Science. In: Lehto, Anna-Marja (Hrsg.)/Eriksson, Inger (Hrsg.): Proceedings of the IFIP-Conference on Women, Work and Computerization. Helsinki, 1991, S. 407–423
- Huff/Cooper 1987:** Huff, Charles/Cooper, Joel: Sex Bias in Educational Software: The Effect fo Designers' Stereotypes on the Software They Design. In: Journal of Applied Social Psychology 17 (1987), Nr. 6, S. 519–532
- Hughes 1993:** Hughes, Eric: A Cypherpunk's Manifesto. Version: 1993. <http://www.activism.net/cypherpunk/manifesto.html>, Abruf: 26.10.2013
- Hunger 2002:** Hunger, Francis: Computer als Männermaschine. Computer in einem patriarchalen und kapitalistischen Kontext. Version: 2002. http://www.irmielin.org/wp-content/uploads/2012/01/computer_als_maennermaschine.pdf, Abruf: 26.10.2013
- Huysman/Lin 2005:** Huysman, Marleen/Lin, Yuwei: Learn to solve problems: A virtual ethnographic case study of learning in a GNU/LINUX user group. In: The Electronic Journal for Virtual Organizations and Networks 7 (2005), S. 56–69
- Hörning/Reuter 2004:** Hörning, Karl H. (Hrsg.)/Reuter, Julia (Hrsg.): Doing Culture. Neue Positionen zum Verhältnis von Kultur und sozialer Praxis. Bielefeld: transcript, 2004
- Imhorst 2004:** Imhorst, Christian: Die Anarchie der Hacker. Richard Stallman und die Freie-Software-Bewegung. Version: 2004. <http://texte.datenteiler.de/wp-content/uploads/2010/05/DAdH.pdf>, Abruf: 26.10.2013
- Imhorst 2005:** Imhorst, Christian: Anarchie und Quellcode — Was hat die freie Software-Bewegung mit Anarchismus zu tun? (2005). http://www.opensourcejahrbuch.de/download/jb2005/chapter_05/osjb2005-05-02-imhorst.pdf, Abruf: 26.10.2013
- Janke/Niehues 1995:** Janke, Klaus/Niehues, Stefan: Echt abgedreht: die Jugend der 90er Jahre. München: Beck, 1995
- Janning 1991:** Janning, Frank: Pierre Bourdieus Theorie der Praxis. Analyse und Kritik der konzeptionellen Grundlegung einer praxeologischen Soziologie. Opladen: Westdeutscher Verlag, 1991

- John 2006:** John, Sara: Un/Realistically Embodied: The Gendered Conceptions of Realistic Game Design. Paper at the Gender and Interaction Workshop, Venice. Version: 2006. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.7357&rep=rep1&type=pdf>, Abruf: 26.10.2013
- Jung 2006:** Jung, Patricia: Frauen-freie Zone Open Source? Version: 2006. http://www.opensourcejahrbuch.de/download/jb2006/chapter_05/osjb2006-05-04-jung.pdf, Abruf: 26.10.2013. In: Lutterbeck, Bernd (Hrsg.)/Bärwolff, Matthias (Hrsg.) /Gehring, Robert A. (Hrsg.): Open Source Jahrbuch 2006. Zwischen Softwareentwicklung und Gesellschaftsmodell. 2006, S. 235–250
- Jörissen/Mattig 2007:** Jörissen, Benjamin/Mattig, Ruprecht: Die rituelle Gestaltung techno-sozialer Lernarrangements. Computerunterricht in der Schule und im Kinderclub. In: Wulf, Christoph (Hrsg.): Lernkulturen im Umbruch. Rituelle Praktiken in Schule, Medien, Familie und Jugend. Wiesbaden: VS Verlag für Sozialwissenschaften, 2007, S. 146–183
- Kahlert/Kajatin 2004:** Kahlert, Heike (Hrsg.)/Kajatin, Claudia (Hrsg.): Arbeit und Vernetzung im Informationszeitalter. Wie neue Technologien die Geschlechterverhältnisse verändern. Frankfurt am Main/New York: Campus Verlag, 2004
- Kannonier-Finster/Ziegler 1998:** Kannonier-Finster, Waltraud (Hrsg.)/Ziegler, Meinrad (Hrsg.): Exemplarische Erkenntnis: zehn Beiträge zur interpretativen Erforschung sozialer Wirklichkeit. Innsbruck/Wien: Studien-Verlag, 1998
- Kehily/Nayak 1997:** Kehily, Mary J./Nayak, Anoop: 'Lads and Laughter': humour and the production of heterosexual hierarchies. In: Gender and Education 9 (1997), Nr. 1, S. 69–87
- Kelle 1999:** Kelle, Helga: Geschlechterunterschiede oder Geschlechterunterscheidung? Methodologische Reflexion eines ethnographischen Forschungsprozesses. In: Dausien, Bettina (Hrsg.)/Herrmann, Martina (Hrsg.)/Oechsle, Mechthild (Hrsg.)/Schmerl, Christiane (Hrsg.) /Stein-Hilbers, Marlene (Hrsg.): Erkenntnisprojekt Geschlecht. Feministische Perspektiven verwandeln Wissenschaft. Opladen: Leske + Budrich, 1999, S. 304–324
- Kelle 2001:** Kelle, Helga: „Ich bin der die das macht“ Oder: über die Schwierigkeit, „doing gender“-Prozesse zu erforschen. In: Feministische Studien (2001), Nr. 2, S. 39–56
- Kelle 2007:** Kelle, Udo: „Emergence“ vs. „Forcing“ of Empirical Data? A Crucial Problem of „Grounded Theory“ Reconsidered. In: Historical Social Research 19 (2007), S. 133–156

-
- Kelty 2005a:** Kelty, Christopher: Free Science. In: Feller, Joseph (Hrsg.)/Fitzgerald, Brian (Hrsg.)/Hissam, Scott A. (Hrsg.) /Lakhani, Karim R. (Hrsg.): *Perspective on Free and Open Source Software*. Cambridge (Mass.)/London: MIT-Press, 2005, S. 415–430
- Kelty 2005b:** Kelty, Christopher: Geeks, Social Imaginaries, and Recursive Publics. In: *Cultural Anthropology* 20 (2005), Nr. 2, S. 185–214
- Kelty 2005c:** Kelty, Christopher: Trust among the Algorithms: Ownership, Identity, and the Collaborative Stewardship of Information. In: Ghosh, Rishab A. (Hrsg.): *CODE: Collaborative Ownership in the Digital Economy*. Cambridge/Mass.: MIT Press, 2005, S. 127–151
- Kelty 2008a:** Kelty, Christopher M.: Logical Instruments: Regular Expressions, AI and thinking about thinking. (2008). http://kelty.org/or/papers/Kelty_Franchi_LogicalInstruments_2009.pdf, Abruf: 26.10.2013
- Kelty 2008b:** Kelty, Christopher M.: *Two Bits. The Cultural Significance of Free Software*. Durham/London: Duke University Press, 2008
- Klein/Friedrich 2003:** Klein, Gabriele/Friedrich, Malte: *Is this real? Die Kultur des HipHop*. Frankfurt am Main: Suhrkamp, 2003
- Kleiner 2010:** Kleiner, Dmytri: The Telekommunist Manifesto. Version: 2010. http://www.networkcultures.org/_uploads/%233notebook_telekommunist.pdf, Abruf: 26.10.2013
- Klinger 1995:** Klinger, Cornelia: Beredtes Schweigen und verschwiegenes Sprechen. Genus im Diskurs der Philosophie. In: Bußmann, Hadumod (Hrsg.)/Hof, Renate (Hrsg.): *Genus. Zur Geschlechterdifferenz in den Kulturwissenschaften*. Stuttgart: Kröner, 1995, S. 34–59
- Klinger/Knapp 2005:** Klinger, Cornelia/Knapp, Gudrun-Axeli: Achsen der Ungleichheit — Achsen der Differenz. Verhältnisbestimmungen von Klasse, Geschlecht, "Rasse"/Ethnizität. In: *Transit — Europäische Revue* (2005), Nr. 29. <http://www.iwm.at/read-listen-watch/transit-online/achsen-der-ungleichheit-achsen-der-differenz/>, Abruf: 26.10.2013
- Knoblauch/Heath 2006:** Knoblauch, Hubert/Heath, Christian: *Die Workplace Studies*. In: *Technografie. Zur Mikrosoziologie der Technik*. Frankfurt am Main: Campus, 2006, S. 141–161

- Knoblauch 2001:** Knoblauch, Hubert A.: Erving Goffmans Reich der Interaktion. In: Interaktion und Geschlecht. Frankfurt am Main: Campus, 2001, S. 7–49
- Knorr Cetina 1998:** Knorr Cetina, Karin: Sozialität mit Objekten. In: Technik und Sozialtheorie. Frankfurt am Main: Campus, 1998, S. 83–120
- Koops 1999:** Koops, Bert-Jaap: The Crypto Controversy. Den Haag: Kluwer Law International, 1999
- Krafft 2010:** Krafft, Martin F.: A Delphi study of the influences on innovation adoption and process evolution in a large open-source project. The case of Debian. 2010
- Krais/Gebauer 2002:** Krais, Beate/Gebauer, Gunter: Habitus. Bielefeld: transcript, 2002
- Kreckel 1983:** Kreckel, Reinhard: Theorien sozialer Ungleichheit im Übergang. In: Kreckel, Reinhard (Hrsg.): Soziale Ungleichheiten (Soziale Welt: Sonderband 2). Göttingen: Verlag Otto Schwartz & Co., 1983, S. 3–12
- Kreckel 2004:** Kreckel, Reinhard: Politische Soziologie der sozialen Ungleichheit. Frankfurt am Main: Campus, 2004
- Krieger 2004:** Krieger, Bernhard: Free/Libre and Open Source Software: Policy Support. FLOSSPOLS Deliverable D05 Ethical Guidelines for Online Research. Version: 2004. <http://flosspols.org/deliverables/FLOSSPOLS-D05-revised-ethicalguidelines.pdf>, Abruf: 26.10.2013
- Krishnamurthy 2002:** Krishnamurthy, Sandeep: Cave or Community? An Empirical Examination of 100 Mature Open Source Projects. Version: 2002. <http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/960/881>, Abruf: 26.10.2013
- Krishnamurthy 2006:** Krishnamurthy, Sandeep: On the Intrinsic and Extrinsic Motivation of Free/Libre/Open Source (FLOSS) Developers. In: Knowledge, Technology, & Policy 18 (2006), Nr. 4, S. 17–39
- von Krogh et al. 2003:** Krogh, Georg von/Spaeth, Sebastian /Lakhani, Karim R.: Community, joining, and specialization in open source software innovation: a case study. In: Research Policy (2003), Nr. 32, S. 1217–1241
- Krotz 2005:** Krotz, Friedrich: Neue Theorien entwickeln. Köln: Herbert von Halem Verlag, 2005

-
- Krätzig 2010:** Krätzig, Sebastian: Die Piratenpartie als ein Puzzlestück in der neuen Konfiguration der Mehrparteienrepublik. In: Neue soziale Bewegungen 23 (2010), Nr. 4, S. 91–97
- Kutteroff et al. 2010:** Kutteroff, Albrecht/Behrens, Peter/König, Tina /Schmid, Thomas: JIM 2010. Jugend, Information, (Multi-)Media. Basisstudie zum Medienumgang 12- bis 19-Jähriger in Deutschland / Medienpädagogischer Forschungsverbund Südwest. Version: 2010. <http://www.mpfs.de/fileadmin/JIM-pdf10/JIM2010.pdf>, Abruf: 26.10.2013. 2010. – Forschungsbericht
- König/Strübing 2010:** König, Anika/Strübing, Jörg: Continuous Structuring in Open Source Software Development. In: Lucas, Klaus (Hrsg.)/Roosen, Peter (Hrsg.): Emergence, Analysis and Evolution of Structures Concepts and Strategies Across Disciplines Series: Understanding Complex Systems. Berlin/Heidelberg: Springer, 2010, S. 226–243
- Lahire 2004:** Lahire, Bernard: La culture des individus: Dissonances culturelles et distinction de soi. Paris: La Découverte, 2004
- Lakhani/von Hippel 2003:** Lakhani, Karim R./Hippel, Eric von: How open source software works: „free“ user-to-user assistance. In: Research Policy (2003), Nr. 32, S. 923–943. <http://flosshub.org/sites/flosshub.org/files/lakhani2003.pdf>, Abruf: 26.10.2013
- Latour 1987:** Latour, Bruno: Science in Action. Cambridge (MA)/London: Harvard University Press, 1987
- Latour 1995:** Latour, Bruno: Wir sind nie modern gewesen. Versuch einer symmetrischen Anthropologie. Berlin: Akademie-Verlag, 1995
- Latour 1996:** Latour, Bruno: Der Berliner Schlüssel. Erkundungen eines Liebhabers der Wissenschaften. Berlin: Akademie-Verlag, 1996
- Latour 2002:** Latour, Bruno: Die Hoffnung der Pandora. Frankfurt am Main: Suhrkamp, 2002
- Latour 2006:** Latour, Bruno: Ethnografie einer Hochtechnologie: Das Pariser Projekt ‚Aramis‘ eines automatischen U-Bahn-Systems. In: Rammert, Werner (Hrsg.)/Schubert, Cornelius (Hrsg.): Technografie. Zur Mikrosoziologie der Technik. Frankfurt am Main/New York: Campus, 2006, S. 25–60

- Lave/Wenger 1991:** Lave, Jean/Wenger, Etienne: Situated learning. Legitimate peripheral participation. Cambridge: Cambridge University Press, 1991
- Lenz 2010:** Lenz, Ilse: Intersektionalität. Zum Wechselverhältnis von Geschlecht und sozialer Ungleichheit. In: Becker, Ruth (Hrsg.)/Kortendiek, Beate (Hrsg.): Handbuch Frauen- und Geschlechterforschung: Theorie, Methoden, Empirie. Wiesbaden: VS Verlag für Sozialwissenschaften, 2010, S. 158–165
- Levy 1993:** Levy, Steven: Crypto Rebels. Version: 1993. http://www.wired.com/wired/archive/1.02/crypto.rebels_pr.html, Abruf: 26.10.2013
- Levy 2001:** Levy, Steven: Crypto. How the Code Rebels Beat the Government — Saving Privacy in the Digital Age. New York: Penguin Books, 2001
- Levy 2010:** Levy, Steven: Hackers. Heroes of the computer revolution. Sebastopol/CA: O'Reilly, 2010 [1984]
- Lie 1995:** Lie, Merete: Technology and Masculinity. The Case of the Computer. In: The European Journal of Women's Studies 2 (1995), Nr. 3, S. 379–394
- Lin 2004a:** Lin, Yuwei: Contextualising knowledge-making in Linux user groups. Version: 2004. <http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1187/1107>, Abruf: 26.10.2013
- Lin 2004b:** Lin, Yuwei: Hacking Practices and Software Development: A Social Worlds Analysis of ICT Innovation and the Role of Free/Libre Open Source Software. Version: 2004. <http://flosshub.org/sites/flosshub.org/files/lin2.pdf>, Abruf: 26.10.2013
- Lin 2005a:** Lin, Yuwei: Inclusion, diversity and gender equality: Gender Dimensions of the Free/Libre Open Source Software Development. Version: 2005. http://flosshub.org/sites/flosshub.org/files/lin3_gender.pdf, Abruf: 26.10.2013
- Lin 2005b:** Lin, Yuwei: A Techno-Feminist Perspective on the Free/Libre Open Source Software Development. Version: 2005. <http://flosshub.org/sites/flosshub.org/files/lin5.pdf>, Abruf: 26.10.2013
- Lin 2006:** Lin, Yuwei: Hybrid Innovation: The Dynamics of Collaboration between the FLOSS Community and Corporations. In: Knowledge, Technology, & Policy 18 (2006), Nr. 4, S. 86–100

-
- Livingstone 1999:** Livingstone, David W.: Informelles Lernen in der Wissensgesellschaft. In: Kompetenz für Europa — Wandel durch Lernen — Lernen im Wandel. Referate auf dem internationalen Fachkongress Berlin (1999), S. 65–92
- Lloyd 2007:** Lloyd, Andreas: 'A system that works for me' - An anthropological analysis of computer hackers' shared use and development of the Ubuntu system. (2007). <http://flosshub.org/sites/flosshub.org/files/ubuntu.pdf>, Abruf: 26.10.2013
- Lorber 2003:** Lorber, Judith: Gender-Paradoxien. Opladen: Leske + Budrich, 2003
- Lovink 2009:** Lovink, Geert: Dynamics of Critical Internet Culture (1994–2001). Version: 2009 (Verteidigung der Dissertation 2002). http://networkcultures.org/_uploads/tod/TOD1_dynamicsofcriticalinternetculture.pdf, Abruf: 26.10.2013
- Lovink/Schultz 2010:** Lovink, Geert/Schultz, Pit: Jugendjahre der Netzkritik. Essays zu Web 1.0 (1995 - 1997). Version: 2010. http://www.networkcultures.org/_uploads/tod/TOD%232.pdf, Abruf: 26.10.2013
- Lucht/Paulitz 2008:** Lucht, Petra (Hrsg.)/Paulitz, Tanja (Hrsg.): Recodierungen des Wissens. Stand und Perspektiven der Geschlechterforschung in Naturwissenschaften und Technik. Frankfurt am Main: Campus, 2008
- Luhmann 1973:** Luhmann, Niklas: Vertrauen. Ein Mechanismus der Reduktion sozialer Komplexität. Stuttgart: Ferdinand Enke Verlag, 1973
- Luthiger Stoll 2006:** Luthiger Stoll, Benno: Spass und Software-Entwicklung. Zur Motivation von Open-Source-Programmierern. Version: 2006. <http://www.dissertationen.unizh.ch/2006/luthigerstoll/diss.pdf>, Abruf: 26.03.07
- Lutz/Wenning 2001:** Lutz, Helma/Wenning, Norbert: Differenzen über Differenz — Einführung in die Debatten. In: Lutz, Helma (Hrsg.)/Wenning, Norbert (Hrsg.): Unterschiedlich verschieden. Differenz in der Erziehungswissenschaft. Opladen: Leske + Budrich, 2001, S. 11–24
- Lübke 2004:** Lübke, Valeska: Cyber Gender. Geschlecht und Körper im Internet. Königstein im Taunus: Helmer, 2004
- Maasen 1999:** Maasen, Sabine: Wissenssoziologie. Bielefeld: transcript, 1999

- Maaß/Rommes 2007:** Maaß, Susanne/Rommes, Els: Uncovering the Invisible: Gender-Sensitive Analysis of Call Center. In: Zorn, Isabel (Hrsg.)/Maass, Susanne (Hrsg.)/Rommes, Els (Hrsg.)/Schirmer, Carola (Hrsg.) /Schelhowe, Heidi (Hrsg.): Gender Designs IT. Construction and Deconstruction of Information Society Technology. Wiesbaden: VS Verlag für Sozialwissenschaften, 2007, S. 97–108
- Maaß/Wiesner 2006:** Maaß, Susanne/Wiesner, Heike: Programmieren, Mathe und ein bisschen Hardware... Wen lockt dieses Bild der Informatik? In: Informatik Spektrum 29 (2006), Nr. 2, S. 125–132
- Maaßen 2009:** Maaßen, Harald: LPIC-1. Sicher zur erfolgreichen Linux-Zertifizierung. Bonn: Galileo Computing, 2009
- do Mar Castro Varela/Dhawan 2005:** Mar Castro Varela, María do/Dhawan, Nikita: Postkoloniale Theorie. Eine kritische Einführung. Bielefeld: transcript, 2005
- Marcus 1986:** Marcus, George E.: Contemporary Problems of Ethnography in the Modern World System. In: Clifford, James (Hrsg.)/Marcus, George E. (Hrsg.): Writing Culture. The Poetics and Politics of Ethnography. Berkeley, Los Angeles, London: University of California Press, 1986, S. 165–193
- Marcus 2005:** Marcus, George E.: The refashioning of Ethnography. In: Soziale Welt Sonderheft 16 (2005), S. 387–396
- Margolis/Fisher 2002:** Margolis, Jane/Fisher, Allan: Unlocking the Clubhouse. Women in Computing. Cambridge, Massachusetts: MIT Press, 2002
- Mauss 1994:** Mauss, Marcel: Die Gabe: Form und Funktion des Austauschs in archaischen Gesellschaften. Frankfurt am Main: Suhrkamp, 1994 [1925]
- May 2000:** May, Christopher: A Global Political Economy of Intellectual Property Rights. London/New York: Routledge, 2000
- May 1992:** May, Timothy C.: The Crypto Anarchist Manifesto. Version: 1992. <http://www.activism.net/cypherpunk/crypto-anarchy.html>, Abruf: 26.10.2013
- Mayring 2007:** Mayring, Philipp: Qualitative Inhaltsanalyse. Weinheim/Basel: Beltz, 2007
- McCall 2005:** McCall, Leslie: The Complexity of Intersectionality. In: Signs 30 (2005), Nr. 3, S. 1771–1800

-
- McIntosh 2003:** McIntosh, Peggy: White Privilege and Male Privilege. A Personal Account of Coming to See Correspondences Through Work in Women's Studies. In: Kimmel, Michael S. (Hrsg.)/Ferber, Abby L. (Hrsg.): Privilege. A reader. Boulder, Colorado: Westview Press, 2003, S. 147–156
- Meier 2004:** Meier, Michael: Bourdieus Theorie der Praxis — eine ‚Theorie sozialer Praktiken‘? In: Hörning, Karl H. (Hrsg.)/Reuter, Julia (Hrsg.): Doing culture. Neue Positionen zum Verhältnis von Kultur und sozialer Praxis. Bielefeld: transcript, 2004, S. 55–69
- Meißelbach 2009:** Meißelbach, Christoph: Web 2.0 — Demokratie 3.0? Baden-Baden: Nomos, 2009
- Meretz 2000:** Meretz, Stefan: LINUX & CO. Freie Software — Ideen für eine andere Gesellschaft. Version: 2000. <http://www.kritische-informatik.de/index.htm?fsrevo.htm>, Abruf: 26.10.2013
- Meuser 2004:** Meuser, Michael: Nichts als alter Wein in neuen Schläuchen? Männlichkeitskonstruktionen im Informationszeitalter. In: Arbeit und Vernetzung im Informationszeitalter. Wie neue Technologien die Geschlechterverhältnisse verändern. Frankfurt am Main/New York: Campus Verlag, 2004, S. 73–93
- Meuser 2006:** Meuser, Michael: Geschlecht und Männlichkeit. Soziologische Theorie und kulturelle Deutungsmuster. Wiesbaden: VS Verlag für Sozialwissenschaften, 2006
- Meuser 2008:** Meuser, Michael: Ernste Spiele. Zur Konstruktion von Männlichkeit im Wettbewerb der Männer. In: Baur, Nina (Hrsg.)/Luedtke, Jens (Hrsg.): Die soziale Konstruktion von Männlichkeit. Hegemoniale und marginalisierte Männlichkeiten in Deutschland. Opladen: Verlag Barbara Budrich, 2008, S. 33–44
- Mey/Mruck 2007:** Mey, Günter/Mruck, Katja: Grounded Theory Methodologie — Bemerkungen zu einem prominenten Forschungsstil. In: Historical Social Research 19 (2007), S. 11–39
- Meyer 2003:** Meyer, Peter B.: Episodes of Collective Innovation. Working Paper des U.S. Department of Labor, Bureau of Labor Statistics. Version: 2003. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=466880, Abruf: 26.10.2013
- Michlmayr 2007:** Michlmayr, Martin: Quality Improvement in Volunteer Free and Open Source Software Projects. Exploring the Impact of Release Management. 2007 <http://www.cyrius.com/publications/michlmayr-phd.pdf>

- Moody 2002:** Moody, Glyn: Rebel Code. The Inside Story of Linux and the Open Source Revolution. Cambridge, Massachusetts: Perseus Publishing, 2002
- Morner/Lanzara 2007:** Morner, Michèle/Lanzara, Giovan F.: Von Organisatoren, die keine Menschen sind: Koordination durch Artefakte in Open-Source-Software-Projekten. In: Lang, Rainhart (Hrsg.)/Schmidt, Annett (Hrsg.): Individuum und Organisation. Neue Trends eines organisationswissenschaftlichen Forschungsfeldes. Wiesbaden: DUV, 2007, S. 225–249
- Mörth/Fröhlich 1994:** Mörth, Ingo (Hrsg.)/Fröhlich, Gerhard (Hrsg.): Das symbolische Kapital der Lebensstile. Zur Kultursoziologie der Moderne nach Pierre Bourdieu. Frankfurt am Main/New York: Campus, 1994
- Müller 2007:** Müller, Hans-Peter: Max Weber. Eine Einführung in sein Werk. Köln/Weimar/Wien: Böhlau Verlag, 2007
- Nafus et al. 2006:** Nafus, Dawn/Leach, James /Krieger, Bernhard: Free/Libre and Open Source Software: Policy Support. FLOSSPOLs Deliverable D 16. Gender. Integrated Report of Findings. (2006). http://www.flosspols.org/deliverables/FLOSSPOLs-D16-Gender_Integrated_Report_of_Findings.pdf, Abruf: 26.10.2013
- Nassehi 2006:** Nassehi, Armin: Der soziologische Diskurs der Moderne. Frankfurt am Main: Suhrkamp, 2006
- Niedermayer 2010:** Niedermayer, Oskar: Erfolg und Erfolgsbedingungen neuer Parteien im Parteiensystem am Beispiel der Piratenpartei Deutschland. In: Zeitschrift für Parlamentsfragen (2010), Nr. 4, S. 838–854
- Nohlen 1998:** Nohlen, Dieter (Hrsg.): Wörterbuch Staat und Politik. Bonn: Bundeszentrale für politische Bildung, 1998
- Nuissl et al. 2002:** Nuissl, Ekkehard (Hrsg.)/Schiersmann, Christiane (Hrsg.) /Siebert, Horst (Hrsg.): Kompetenzentwicklung statt Bildungsziele? Bielefeld: Bertelsmann, 2002
- Nuss 2006:** Nuss, Sabine: Copyright & Copyriot. Aneignungskonflikte um geistiges Eigentum im informationellen Kapitalismus. Münster: Westfälisches Dampfboot, 2006
- Nuss/Heinrich 2002:** Nuss, Sabine/Heinrich, Michael: Warum Freie Software dem Kapitalismus nicht viel anhaben kann — aber vielleicht trotzdem etwas mit Kommunismus

zu tun hat. (2002). <http://erste.oekonux-konferenz.de/dokumentation/texte/nuss.html>,
Abruf: 26.10.2013

von Oertzen 2006: Oertzen, Jürgen von: Grounded Theory. In: Behnke, Joachim (Hrsg.)/Gschwend, Thomas (Hrsg.)/Schindler, Delia (Hrsg.) /Schnapp, Kai-Uwe (Hrsg.): Methoden der Politikwissenschaft. Neuere qualitative und quantitative Analyseverfahren. Baden-Baden: Nomos, 2006, S. 145–154

Offenberger 2010: Offenberger, Ursula: Technologie und Geschlecht: Vergeschlechtlichte Praktiken, Objektivierungen und Deutungsmuster an der Schnittstelle von Entwicklung und Nutzung von Energietechnologien. In: Biniok, Peter (Hrsg.): Technik, Wissenschaft und Politik. Neue Forschungsperspektiven. Frankfurt am Main: Peter Lang, 2010, S. 85–94

Offenberger/Nentwich 2009: Offenberger, Ursula/Nentwich, Julia: Home Heating and the Co-construction of Gender, Technology and Sustainability. In: Women, Gender and Research 18 (2009), Nr. 3-4, S. 83–91

Oksaar 1988: Oksaar, Els: Fachsprachliche Dimensionen. Tübingen: Gunter Narr Verlag, 1988

O'Mahony/Ferraro 2004: O'Mahony, Siobhán/Ferraro, Fabrizio: Hacking Alone? The Effects of Online and Offline Participation on Open Source Community Leadership. (2004). <http://flosshub.org/sites/flosshub.org/files/omahonyferraro2.pdf>, Abruf: 26.10.2013

O'Mahony/Ferraro 2007: O'Mahony, Siobhán/Ferraro, Fabrizio: The emergence of governance in an open source community. In: Academy of Management Journal 50 (2007), Nr. 5, S. 1079–1106

O'Neil 2009: O'Neil, Mathieu: Cyberchiefs. Autonomy and Authority in Online Tribes. London, New York: Pluto Press, 2009

van Oost 2003: Oost, Ellen van: Materialized Gender: How Shavers Configure the Users' Femininity and Masculinity. In: Oudshoorn, Nelly (Hrsg.)/Pinch, Trevor (Hrsg.): How Users Matter. The Co-Construction of Users and Technology. Cambridge/Mass.: MIT Press, 2003, S. 193–208

Oudshoorn et al. 2004: Oudshoorn, Nelly/Rommes, Els /Stienstra, Marcelle: Configuring the User as Everybody: Gender and the Design Cultures in Information and Communication Technologies. In: Science, Technology & Human Values 29 (2004), Nr. 1, S. 30–63

- Overwien 2005:** Overwien, Bernd: Stichwort: Informelles Lernen. In: Zeitschrift für Erziehungswissenschaft (2005), Nr. 3, S. 339–355
- Overwien 2010:** Overwien, Bernd: Zur Bedeutung informellen Lernens. In: Neuber, Nils (Hrsg.): Informelles Lernen im Sport: Beiträge zur allgemeinen Bildungsdebatte. Wiesbaden: VS Verlag für Sozialwissenschaften, 2010, S. 35–51
- Paoli/D’Andrea 2008:** Paoli, Stefano D./D’Andrea, Vincenzo: How artifacts rule web based communities: Practices of Free Software Development. In: International Journal of Web Based Communities 4 (2008), S. 199–219
- Patzelt 1987:** Patzelt, Werner J.: Grundlagen der Ethnomethodologie: Theorie, Empirie und politikwissenschaftlicher Nutzen. München: Wilhelm Fink Verlag, 1987
- Paulitz 2005:** Paulitz, Tanja: Netzsubjektivitäten. Konstruktionen von Vernetzung als Technologien des sozialen Selbst. Münster: Verlag Westfälisches Dampfboot, 2005
- Perelman 2002:** Perelman, Michael: Steal This Idea. Intellectual property rights and the corporative confiscation of creativity. New York/Hampshire: Palgrave, 2002
- Perlot 2008:** Perlot, Flooh: Deliberative Demokratie und Internetforen — Nur eine virtuelle Diskussion? Baden-Baden: Nomos, 2008
- Peters 2009:** Peters, Manuel: Zur sozialen Praxis der (Nicht-) Zugehörigkeiten. Die Bedeutung zentraler Theorien von Bourdieu und Goffman für einen Blick auf Migration, Zugehörigkeit und Interkulturelle Pädagogik. Oldenburg: BIS-Verlag der Carl von Ossietzky Universität Oldenburg, 2009
- Pietraß et al. 2005:** Pietraß, Manuela/Schmidt, Bernhard /Tippelt, Rudolf: Informelles Lernen und Medienbildung. Zur Bedeutung sozio-kultureller Voraussetzungen. In: Zeitschrift für Erziehungswissenschaft 8 (2005), Nr. 3, S. 412–426
- Pilarczyk 2011:** Pilarczyk, Hannah: „Ich will keine Asis als Freunde“. Soziale Spaltung im Netz. In: Spiegel Online, 26.04.2011 (2011). <http://www.spiegel.de/netzwelt/web/0,1518,757280,00.html>, Abruf: 26.10.2013
- Preisendörfer 1995:** Preisendörfer, Peter: Vertrauen als soziologische Kategorie. Möglichkeiten und Grenzen einer entscheidungstheoretischen Fundierung des Vertrauenskonzepts. In: Zeitschrift für Soziologie 24 (1995), Nr. 4, S. 263–272

-
- Pritzlaff 2006:** Pritzlaff, Tanja: Ethnographische Politikforschung. In: Behnke, Joachim (Hrsg.)/Gschwend, Thomas (Hrsg.)/Schindler, Delia (Hrsg.) /Schnapp, Kai-Uwe (Hrsg.): Methoden der Politikwissenschaft. Neuere qualitative und quantitative Analyseverfahren. Baden-Baden: Nomos, 2006, S. 125–132
- Psychologie 2005:** Psychologie, Forum K.: Lernverhältnisse: Kategoriale, theoretische und methodische Probleme des Lernens und Lehrens. Hamburg: Argument, 2005
- Rammert 1998:** Rammert, Werner (Hrsg.): Technik und Sozialtheorie. Frankfurt am Main: Campus, 1998
- Rammert 2006:** Rammert, Werner: Technik in Aktion: Verteiltes Handeln in soziotechnischen Konstellationen. In: Technografie. Zur Mikrosoziologie der Technik. Frankfurt am Main: Campus, 2006, S. 163–195
- Rammert et al. 1991:** Rammert, Werner/Böhm, Wolfgang/Olscha, Christian /Wehner, Josef: Vom Umgang mit Computern im Alltag. Fallstudien zur Kultivierung einer neuen Technik. Opladen: Westdeutscher Verlag, 1991
- Rammert/Schubert 2006:** Rammert, Werner (Hrsg.)/Schubert, Cornelius (Hrsg.): Technografie. Zur Mikrosoziologie der Technik. Frankfurt/New York: Campus, 2006
- Ratto 2003:** Ratto, Matthew V.: The pressure of openness: the hybrid work of linux free/open source kernel developers. University of California, San Diego, 2003
- Raymond 2003:** Raymond, Eric: The Jargon File. Version: 2003 [letzte Version]. <http://www.catb.org/jargon/html/>, Abruf: 26.10.2013
- Raymond 2000:** Raymond, Eric S.: The Cathedral and the Bazaar. Version: 2000. <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>, Abruf: 26.10.2013
- Raymond 2006:** Raymond, Eric S.: The Luxury of Ignorance: An Open-Source Horror Story. Version: 2006. <http://www.catb.org/~esr/writings/cups-horror.html>, Abruf: 26.10.2013
- Reckwitz 2003:** Reckwitz, Andreas: Grundelemente einer Theorie sozialer Praktiken. Eine sozialtheoretische Perspektive. In: Zeitschrift für Soziologie 32 (2003), Nr. 4, S. 282–301
- Reckwitz 2004:** Reckwitz, Andreas: Die Reproduktion und die Subversion sozialer Praktiken. Zugleich ein Kommentar zu Pierre Bourdieu und Judith Butler. In: Hörning, Karl H.

- (Hrsg.)/Reuter, Julia (Hrsg.): Doing culture. Neue Positionen zum Verhältnis von Kultur und sozialer Praxis. Bielefeld: transcript, 2004, S. 40–54
- Rommelspacher 1995:** Rommelspacher, Birgit: Dominanzkultur. Texte zu Fremdheit und Macht. Berlin: Orlanda Frauenverlag, 1995
- Rommes 2002:** Rommes, Els: Gender Scripts and the Internet. The Design and Use of Amsterdam's Digital City. Enschede: Twente University Press, 2002
- Rooksby et al. 2009:** Rooksby, John/Rouncefield, Mark /Sommerville, Ian: Testing in the Wild: The Social and Organisational Dimensions of Real World Practice. In: Computer Supported Cooperative Work 18 (2009), Nr. 5/6, S. 559–580
- Rosaldo 1986:** Rosaldo, Renato: From the Door of His Tent: The Fieldworker and the Inquisitor. In: Clifford, James (Hrsg.)/Marcus, George E. (Hrsg.): Writing Culture. The Poetics and Politics of Ethnography. Berkeley, Los Angeles, London: University of California Press, 1986, S. 77–97
- Rostock 2005:** Rostock, Petra: Reproduktionsarbeitsteilung und gesellschaftliche Geschlechterarrangements. „Hausarbeiter-innen“ als individuelle Vereinbarkeitsstrategie? Berlin: Diplomarbeit am Otto-Suhr-Institut für Politikwissenschaft der FU Berlin, 2005
- Roßnagel 2006:** Roßnagel, Alexander: Datenschutz im 21. Jahrhundert. In: Aus Politik und Zeitgeschichte (2006), Nr. 5/6, S. 9–15
- Ruffing 2009:** Ruffing, Reiner: Bruno Latour. Paderborn: Wilhelm Fink, 2009
- Said 1989:** Said, Edward W.: Representing the Colonized: Anthropology's Interlocutors. In: Critical Inquiry 15 (1989), Nr. 2, S. 205–225
- Saupe 2002:** Saupe, Angelika: Verlebendigung der Technik: Perspektiven im feministischen Technikdiskurs. Bielefeld: Kleine Verlag, 2002
- Schatzki 2001a:** Schatzki, Theodore R.: Practice mind-ed orders. In: The Practice Turn in Contemporary Theory. London/New York: Routledge, 2001, S. 42–55
- Schatzki 2001b:** Schatzki, Theordore R.: Introduction: practice theory. In: The Practice Turn in Contemporary Theory. London/New York: Routledge, 2001, S. 1–14
- Schatzki et al. 2001:** Schatzki, Theordore R. (Hrsg.)/Cetina, Karin K. (Hrsg.) /Savigny, Eike von (Hrsg.): The Practice Turn in Contemporary Theory. London/New York: Routledge, 2001

-
- Scheffer 1997:** Scheffer, Thomas: Der administrative Blick. Über den Gebrauch des Passes in der Ausländerbehörde. In: Amann, Klaus (Hrsg.)/Hirschauer, Stefan (Hrsg.): Die Befremdung der eigenen Kultur. Zur ethnographischen Herausforderung soziologischer Empirie. Frankfurt am Main: Suhrkamp, 1997, S. 168–197
- Schelhowe 1997:** Schelhowe, Heidi: Das Medium aus der Maschine. Zur Metamorphose des Computers. Frankfurt/Main, New York: Campus, 1997
- Schelhowe oJ:** Schelhowe, Heidi: Hat der Computer ein Geschlecht? Frauenforschung in der Informatik. Version: (o.J.). <http://www.ikoe.de/Fraforasc.htm>, Abruf: 26.10.2013
- Scheuch 2003:** Scheuch, Michael: Demokratie per Mausklick. Neue Informationstechnologien und ihre Auswirkungen auf Demokratietheorie. Stuttgart: ibidem-Verlag, 2003
- Schinzel 2004a:** Schinzel, Britta: Kulturunterschiede beim Frauenanteil im Studium der Informatik. (2004). <http://mod.iig.uni-freiburg.de/cms/fileadmin/publikationen/online-publikationen/Frauenanteil.Informatik.International.pdf>, Abruf: 26.10.2013
- Schinzel 2004b:** Schinzel, Britta: Kulturunterschiede beim Frauenanteil im Studium der Informatik. Teil II: Informatik in Deutschland. (2004). <http://mod.iig.uni-freiburg.de/cms/fileadmin/publikationen/online-publikationen/Informatik.Frauen.Deutschland.pdf>, Abruf: 26.10.2013
- Schinzel 2005a:** Schinzel, Britta: Kulturunterschiede beim Frauenanteil im Studium der Informatik. Teil III: Partikularisierung der Informatik Frauenbeteiligung. (2005). <http://mod.iig.uni-freiburg.de/cms/fileadmin/publikationen/online-publikationen/Frauenbeteiligung.Informatikstudien.pdf>, Abruf: 26.10.2013
- Schinzel 2005b:** Schinzel, Britta: Kulturunterschiede beim Frauenanteil im Studium der Informatik. Teil IV: Abschliessende Interpretation und Literaturangaben. (2005). <http://mod.iig.uni-freiburg.de/cms/fileadmin/publikationen/online-publikationen/Informatik.Kultur.Literatur1.pdf>, Abruf: 26.10.2013
- Schinzel/Ben 2004:** Schinzel, Britta/Ben, Esther R.: Softwareentwicklung als Profession? Professionalisierungstendenzen und Implikationen für die Beteiligung von Frauen. In: Informatik-Spektrum 27 (2004), Oktober, Nr. 5, S. 441–447
- Schmidt 2008:** Schmidt, Robert: Praktiken des Programmierens. Zur Morphologie von Wissensarbeit in der Software-Entwicklung. In: Zeitschrift für Soziologie 37 (2008), August, Nr. 4, S. 282–300

- Schulz-Schaeffer 1999:** Schulz-Schaeffer, Ingo: Technik und die Dualität von Ressourcen und Routinen. Zur sozialen Bedeutung gegenständlicher Technik. In: Zeitschrift für Soziologie 28 (1999), Nr. 6, S. 409–428
- Schulz-Schaeffer 2000:** Schulz-Schaeffer, Ingo: Akteur-Netzwerk-Theorie. Zur Koevolution von Gesellschaft, Natur und Technik. Version: 2000. <http://www.uni-due.de/imperia/md/content/soziologie/akteurnetzwerktheorie.pdf>, Abruf: 26.10.2013. In: Weyer, Johannes (Hrsg.): Soziale Netzwerke. Konzepte und Methoden der sozialwissenschaftlichen Netzwerkforschung. München: R. Oldenbourg Verlag, 2000, S. 187-209
- Schulz-Schaeffer 2004a:** Schulz-Schaeffer, Ingo: Regelmäßigkeit und Regelmäßigkeit. Die Abschirmung des technischen Kerns als Leistung der Praxis. In: Hörning, Karl H. (Hrsg.)/Reuter, Julia (Hrsg.): Doing Culture. Neue Positionen zum Verhältnis von Kultur und sozialer Praxis. Bielefeld: transcript, 2004, S. 108–126
- Schulz-Schaeffer 2004b:** Schulz-Schaeffer, Ingo: Technik als altes Haus und geschichtsloses Appartement. Vom Nutzen und Nachteil der Praxistheorie Bourdieus für die Techniksoziologie. In: Bourdieus Theorie der Praxis. Erklärungskraft — Anwendung — Perspektiven. Wiesbaden: VS Verlag für Sozialwissenschaften, 2004, S. 47–65
- Schwaberow 2001:** Schwaberow, Volker: Straffe Verwaltung. OpenLDAP-Praxis. In: Linux-Magazin (2001), Nr. 5. <http://www.linux-magazin.de/Heft-Abo/Ausgaben/2001/05/Straffe-Verwaltung>, Abruf: 26.10.2013
- Schwingel 2000:** Schwingel, Markus: Pierre Bourdieu zu Einführung. Hamburg: Junius, 2000
- Schöne 2005:** Schöne, Helmar: Die teilnehmende Beobachtung als Datenerhebungsmethode in der Politikwissenschaft. Methodologische Reflexion und Werkstattbericht. In: Historical Social Research 30 (2005), Nr. 1, S. 168–199
- Schöne 2006:** Schöne, Helmar: Sequenzanalyse. In: Behnke, Joachim (Hrsg.)/Gschwend, Thomas (Hrsg.)/Schindler, Delia (Hrsg.) /Schnapp, Kai-Uwe (Hrsg.): Methoden der Politikwissenschaft. Neuere qualitative und quantitative Analyseverfahren. Baden-Baden: Nomos, 2006, S. 335–344
- Schütte 1991:** Schütte, Wilfried: Scherzkommunikation unter Orchestermusikern. Interaktionsformen in einer Berufswelt. Gunter Narr Verlag, 1991

-
- Siefkes 2007:** Siefkes, Christian: From Exchange to Contributions. Generalizing Peer Production into the Physical World. Berlin: Edition C. Siefkes, 2007
- Siefkes et al. 1998:** Siefkes, Dirk (Hrsg.)/Eulenhofer, Peter (Hrsg.)/Stach, Heike (Hrsg.) /Stadtler, Klaus (Hrsg.): Sozialgeschichte der Informatik. Kulturelle Praktiken und Orientierungen. Wiesbaden: Deutscher Universitätsverlag, 1998
- Singh 1999:** Singh, Simon: The Code Book. The science of secrecy from ancient Egypt to quantum cryptography. New York: Anchor Books, 1999
- Smith 1987:** Smith, Dorothy E.: The Everyday World As Problematic. A Feminist Sociology. Boston: Northeastern University Press, 1987
- Snoek 2003:** Snoek, Johannes: Performance, Performativity, and Practice. Against Terminological Confusion in Ritual Studies. In: Paragrana. Internationale Zeitschrift für Historische Anthropologie 12 (2003), Nr. 1/2, S. 78–87
- Sowe et al. 2008:** Sowe, Sulayman K./Stamelos, Ioannis /Angelis, Lefteris: Understanding knowledge sharing activities in free/open source software projects: An empirical study. In: The Journal of Systems and Software 81 (2008), Nr. 3, S. 431–446
- Spradley 1980:** Spradley, James P.: Participant Observation. New York [u.a.]: Holt, Rinehart and Winston, 1980
- Stallman 1985:** Stallman, Richard: The GNU Manifesto. Version: 1985. <http://www.gnu.org/gnu/manifesto.html>, Abruf: 26.10.2013
- Stecher 2005:** Stecher, Ludwig: Informelles Lernen bei Kindern und Jugendlichen und die Reproduktion sozialer Ungleichheit. In: Zeitschrift für Erziehungswissenschaft (2005), Nr. 3, S. 374–393
- Strauss/Corbin 1996:** Strauss, Anselm/Corbin, Juliet: Grounded Theory: Grundlagen Qualitativer Sozialforschung. Weinheim: Beltz Psychologie Verlags Union, 1996
- Strauss 1994:** Strauss, Anselm L.: Grundlagen qualitativer Sozialforschung. Datenanalyse und Theoriebildung in der empirischen soziologischen Forschung. München: Wilhelm Fink Verlag, 1994
- Strübing 1992:** Strübing, Jörg: Arbeitsstil und Habitus. Zur Bedeutung kultureller Phänomene in der Programmierarbeit. Kassel: Wissenschaftliches Zentrum für Berufs- und Hochschulforschung der Gesamthochschule Kassel, 1992

- Strübing 2004:** Strübing, Jörg: Grounded Theory: Zur sozialtheoretischen und epistemologischen Fundierung des Verfahrens der empirisch begründeten Theoriebildung. Wiesbaden: VS Verlag für Sozialwissenschaften, 2004
- Strübing 2005:** Strübing, Jörg: Pragmatistische Wissenschafts- und Technikforschung. Theorie und Methode. Frankfurt am Main: Campus, 2005
- Strübing 2007:** Strübing, Jörg: Glaser vs. Strauss? Zur methodologischen und methodischen Substanz einer Unterscheidung zweier Varianten von Grounded Theory. In: Historical Social Research 19 (2007), S. 157–174
- Störel 2004:** Störel, Thomas: Der Witz in der Fachsprachenforschung — Gedanken zu einer vernachlässigten Textsorte. In: Baumann, Klaus-Dieter (Hrsg.)/Kalverkämper, Hartwig (Hrsg.): Pluralität in der Fachsprachenforschung. Tübingen: Gunter Narr Verlag, 2004, S. 423–433
- Suchman 1987:** Suchman, Lucy: Plans and situated actions. The problem of human-machine communication. Cambridge: Cambridge University Press, 1987
- Suchman 1994:** Suchman, Lucy: Do Categories Have Politics? The language/action perspective reconsidered. In: Computer Supported Cooperative Work (CSCW) (1994), Nr. 2, S. 177–190
- Suchman 2002:** Suchman, Lucy: Located Accountabilities in Technology Production. In: Scandinavian Journal of Information Systems 14 (2002), Nr. 2, S. 91–105
- Suchman 2007:** Suchman, Lucy: Agencies in Technology Design: Feminist Reconfigurations. Version: 2007. <http://www.lancaster.ac.uk/sociology/research/publications/papers/suchman-agenciestechnodesign.pdf>, Abruf: 26.10.2013
- Südostasien-Gesellschaft 1998:** Südostasien-Gesellschaft, SEACOM (Hrsg.): Zwischen Verklärung und Erwachen. Eine kritische Betrachtung ethnographischer Beiträge zur Südostasienskunde. Berlin: SEACOM Edition, 1998
- Tervooren 2001:** Tervooren, Anja: Körper, Inszenierung und Geschlecht. Judith Butlers Konzept der Performativität. In: Wulf, Christoph (Hrsg.)/Göhlich, Michael (Hrsg.) /Zirfas, Jörg (Hrsg.): Grundlagen des Performativen. Eine Einführung in die Zusammenhänge von Sprache, Macht und Handeln. Weinheim/München: Juventa, 2001, S. 157–180

-
- Tervooren 2006:** Tervooren, Anja: Im Spielraum von Geschlecht und Begehren. Ethnographie der ausgehenden Kindheit. Weinheim/München: Juventa, 2006
- Teubner 1995:** Teubner, Ulrike: Das Fiktionale der Geschlechterdifferenz. Oder: wie geschlechtsspezifisch ist die Kategorie Geschlecht? In: Wetterer, Angelika (Hrsg.): Die soziale Konstruktion von Geschlecht in Professionalisierungsprozessen. Frankfurt am Main/-New York: Campus, 1995, S. 247–262
- Thaler 2008:** Thaler, Anita: Laughing among engineers: About the interaction of gender, technology and humour. In: Karner, Sandra (Hrsg.)/Wieser, Bernhard (Hrsg.) /Getzinger, Günter (Hrsg.): Proceedings of the 7th Annual IAS-STC Conference: Critical Issues in Science and Technology Studies, 8th-9th May 2008, 2008
- Themelidis 2004:** Themelidis, Markos: Open Source. Die Freiheitsvision der Hacker. Norderstedt: Books on Demand GmbH, 2004
- Thürmer-Rohr 2010:** Thürmer-Rohr, Christina: Mittäterschaft von Frauen: Die Komplizenschaft mit der Unterdrückung. In: Becker, Ruth (Hrsg.)/Kortendiek, Beate (Hrsg.): Handbuch Frauen- und Geschlechterforschung: Theorie, Methoden, Empirie. Wiesbaden: VS Verlag für Sozialwissenschaften, 2010, S. 88–93
- Torvalds/Diamond 2001:** Torvalds, Linus/Diamond, David: Just for fun. Wie ein Freak die Computerwelt revolutionierte. München/Wien: Carl Hanser Verlag, 2001
- Tully 1994:** Tully, Claus J.: Lernen in der Informationsgesellschaft. Informelle Bildung durch Computer und Medien. Opladen: Westdeutscher Verlag, 1994
- Tully 2003:** Tully, Claus J.: Mensch — Maschine — Megabyte. Technik in der Alltagskultur. Eine sozialwissenschaftliche Hinführung. Opladen: Leske + Budrich, 2003
- Turkle 1986:** Turkle, Sherry: Die Wunschmaschine: Der Computer als zweites Ich. Reinbek bei Hamburg: Rowohlt, 1986
- Turkle 1998:** Turkle, Sherry: Leben im Netz. Identität in Zeiten des Internet. Reinbek bei Hamburg: Rowohlt Taschenbuch Verlag, 1998
- Villa 2007:** Villa, Paula-Irene: Wie Geschlecht gemacht wird. In: Hark, Sabine (Hrsg.): Dis/Kontinuitäten: Feministische Theorie. VS Verlag für Sozialwissenschaften, 2007, S. 19–26

- Wachendorfer 2001:** Wachendorfer, Ursula: Weiß-Sein in Deutschland. Zur Unsichtbarkeit einer herrschenden Normalität. In: Arndt, Susan (Hrsg.): AfrikaBilder. Studien zu Rassismus in Deutschland. Münster: Unrast, 2001, S. 87–101
- Wacquant 1989:** Wacquant, Loïc J. D.: Corps et âme. Notes ethnographiques d'un apprenti-boxeur. In: Actes de la recherche en sciences sociales (1989), Nr. 80, S. 33–67
- Wacquant 1996:** Wacquant, Loïc J. D.: Un mariage dans le ghetto. In: Actes de la recherche en sciences sociales (1996), Nr. 113, S. 63–84
- Wacquant 2003:** Wacquant, Loïc J. D.: Leben für den Ring. Boxen im amerikanischen Ghetto. Konstanz: UVK Verlagsgesellschaft, 2003
- Wagner 2006:** Wagner, Wolf: Kulturschock Deutschland — revisited. 2005. Version: 2006. <http://www.erato.fh-erfurt.de/so/homepages/wagner/Zuindex/Kulturschock%20Deutschland%202005.pdf>, Abruf: 26.10.2013
- Wajcman 1994:** Wajcman, Judy: Technik und Geschlecht. Die feministische Technikdebatte. Frankfurt am Main: Campus, 1994
- Wajcman 2000:** Wajcman, Judy: Reflections on Gender and Technology Studies: In What State is the Art? In: Social Studies of Science 30 (2000), Nr. 3, 447-464. <http://www.jstor.org/stable/285810>, Abruf: 26.10.2013
- Wajcman 2002:** Wajcman, Judy: Addressing Technological Change: The Challenge to Social Theory. In: Current Sociology 50 (2002), Nr. 3, S. 347–363
- Wajcman 2008:** Wajcman, Judy: Life in the fast lane? Towards a sociology of technology and time. In: The British Journal of Sociology 59 (2008), Nr. 1, S. 59–77
- Waugh/Metcalf 2008:** Waugh, Pia/Metcalf, Randy: The Foundations of Openness. Evaluating aspects of openness in software projects. (2008). <http://flosshub.org/sites/flosshub.org/files/Foundations-of-openness-V2-release.pdf>, Abruf: 26.10.2013
- Weber/Bath 2007:** Weber, Jutta/Bath, Corinna: 'Social' Robots & 'Emotional' Software Agents: Gendering Processes and De-Gendering Strategies for 'Technologies in the Making'. In: Zorn, Isabel (Hrsg.)/Maass, Susanne (Hrsg.)/Rommens, Els (Hrsg.)/Schirmer, Carola (Hrsg.)/Schelhowe, Heidi (Hrsg.): Gender Designs IT. Construction and Deconstruction of Information Society Technology. Wiesbaden: VS Verlag für Sozialwissenschaften, 2007, S. 53–63

-
- Weber 2006a:** Weber, Max: Politik und Gesellschaft. Frankfurt am Main: Zweitausendeins, 2006
- Weber 2006b:** Weber, Max: Politik als Beruf. In: Politik und Gesellschaft. Frankfurt am Main: Zweitausendeins, 2006 (Orig. 1919), S. 565–610
- Weizenbaum 1977:** Weizenbaum, Joseph: Die Macht der Computer und die Ohnmacht der Vernunft. Frankfurt am Main: Suhrkamp, 1977
- Weller/Meiszner 2008:** Weller, Martin/Meiszner, Andreas: Report on the effectiveness of a FLOSS-like learning community in formal educational settings. Version: 2008. <http://describd.com/doc/2285513/Report-on-the-effectiveness-of-a-Free-Libre-Open-Source-Software-like-FLOSSlike-learning-community-in-formal-educational-settings>, Abruf: 26.10.2013
- Wenger 1998:** Wenger, Etienne: Communities of practice. Learning, meaning, and identity. Cambridge/UK: Cambridge University Press, 1998
- West/Fenstermaker 1995:** West, Candace/Fenstermaker, Sarah: Doing Difference. In: Gender and Society 9 (1995), Nr. 1, S. 8–37
- West/Zimmerman 1987:** West, Candace/Zimmerman, Don H.: Doing Gender. In: Gender and Society 1 (1987), Nr. 2, S. 125–151
- Wetterer 1993:** Wetterer, Angelika: Professionalisierung und Geschlechterhierarchie. Vom kollektiven Frauenausschluss zur Integration mit beschränkten Möglichkeiten. Kassel: Jenior & Pressler, 1993
- Wetterer 1995a:** Wetterer, Angelika (Hrsg.): Die soziale Konstruktion von Geschlecht in Professionalisierungsprozessen. Frankfurt/New York: Campus, 1995
- Wetterer 1995b:** Wetterer, Angelika: Die soziale Konstruktion von Geschlecht in Professionalisierungsprozessen. Einleitung. In: Wetterer, Angelika (Hrsg.): Die soziale Konstruktion von Geschlecht in Professionalisierungsprozessen. Frankfurt/New York: Campus, 1995
- Wetterer 2002:** Wetterer, Angelika: Arbeitsteilung und Geschlechterkonstruktion. „Gender at Work“ in theoretischer und historischer Perspektive. Konstanz: UVK Verlagsgesellschaft, 2002

- Wieser 2004:** Wieser, Matthias: Inmitten der Dinge. Zum Verhältnis von sozialen Praktiken und Artefakten. In: Hörning, Karl H. (Hrsg.)/Reuter, Julia (Hrsg.): *Doing Culture. Neue Positionen zum Verhältnis von Kultur und sozialer Praxis.* Bielefeld: transcript, 2004, S. 92–107
- Willis 1997:** Willis, Sarah: The Moral Order of an Information System. In: Grundy, A. F. (Hrsg.)/Köhler, Doris (Hrsg.)/Oechtering, Veronika (Hrsg.) /Petersen, Ulrike (Hrsg.): *Women, Work and Computerization. Spinning a Web from Past to Future. Proceedings of the 6th International IFIP-Conference.* Berlin/Heidelberg/New York: Springer, 1997, S. 151–161
- Winker 2005:** Winker, Gabriele: Ko-Materialisierung von vergeschlechtlichten Körpern und technisierten Artefakten: Der Fall Internet. In: Funder, Maria (Hrsg.)/Dörhöfer, Steffen (Hrsg.) /Rauch, Christian (Hrsg.): *Jenseits der Geschlechterdifferenz? Geschlechterverhältnisse in der Informations- und Wissensgesellschaft.* München/Mering: Rainer Hampp Verlag, 2005, S. 157–178
- Winker/Degele 2009:** Winker, Gabriele/Degele, Nina: *Intersektionalität. Zur Analyse sozialer Ungleichheiten.* Bielefeld: transcript, 2009
- Wollrad 2005:** Wollrad, Eske: *Weißsein im Widerspruch. Feministische Perspektiven auf Rassismus, Kultur und Religion.* Königstein/Taunus: Helmer Verlag, 2005
- Wulf/Zirfas 2003:** Wulf, Christoph/Zirfas, Jörg: Anthropologie und Ritual. In: *Paragrana. Internationale Zeitschrift für Historische Anthropologie* 12 (2003), Nr. 1/2, S. 11–28
- Wächter 2003:** Wächter, Christine: *Technik-Bildung und Geschlecht.* München/Wien: Profil Verlag, 2003
- Wölbart 2008:** Wölbart, Christian: Unter Verschluss. In: *sueddeutsche.de Digital* (2008), Nr. 06.02.2008. <http://www.sueddeutsche.de/digital/2.220/internet-historie-unter-verschluss-1.274917>, Abruf: 26.10.2013
- Yeats 2006:** Yeats, Dave: *Open-Source Software Development And User-Centered Design: A Study Of Open-Source Practices And Participants.* Version: 2006. http://www.researchgate.net/publication/228420310_Open-source_software_development_and_user-centered_design_a_study_of_open-source_practices_and_participants/file/d912f512ccb50351f3.pdf, Abruf: 17.11.2013

Young 1961: Young, Michael: Es lebe die Ungleichheit. Auf dem Wege zur Meritokratie. Düsseldorf: Econ-Verlag GmbH, 1961

Zboralski 2007: Zboralski, Katja: Wissensmanagement durch Communities of Practice. Eine empirische Untersuchung von Wissensnetzwerken. Wiesbaden: Deutscher Universitäts-Verlag, 2007

Zorn et al. 2007: Zorn, Isabel (Hrsg.)/Maass, Susanne (Hrsg.)/Rommes, Els (Hrsg.)/Schirmer, Carola (Hrsg.) /Schelhowe, Heidi (Hrsg.): Gender Designs IT. Construction and Deconstruction of Information Society Technology. Wiesbaden: VS Verlag für Sozialwissenschaften, 2007

Zürcher 2007: Zürcher, Reinhard: Informelles Lernen und der Erwerb von Kompetenzen. Theoretische, didaktische und politische Aspekte. Wien: Bundesministerium für Unterricht, Kunst und Kultur. Abteilung Erwachsenenbildung V/8, 2007 http://erwachsenenbildung.at/downloads/service/nr2_2007_informelles_lernen.pdf

Materialverzeichnis

KSP 2007: Beobachtungsprotokoll 1. Keysigning Party. 2007

LUG 2007a: Beobachtungsprotokoll LUG 1. 2007

LUG 2007b: Beobachtungsprotokoll LUG 1. 2007

LUG 2007c: Beobachtungsprotokoll LUG 1. 2007

LUG 2007d: Beobachtungsprotokoll LUG1. 2007

Int 2007a: Interview mit Debian-Maintainer. 2007

Int 2007b: Interview mit Freifunker. 2007

KSP 2008a: Beobachtungsprotokoll 2. Keysigning Party. 2008

KSP 2008b: Beobachtungsprotokoll 3. Keysigning Party. 2008

KSP 2008c: Beobachtungsprotokoll 4. Keysigning Party. 2008

Sta 2008: Beobachtungsprotokoll eines Vortrags von Richard Stallman. Berlin, 2008

LUG 2008a: Beobachtungsprotokoll LUG 1. 2008

LUG 2008b: Beobachtungsprotokoll LUG 1. 2008

LUG 2008c: Beobachtungsprotokoll LUG 1. 2008

LUG 2008d: Beobachtungsprotokoll LUG 1. 2008

LUG 2008f: Beobachtungsprotokoll LUG 1. 2008

LUG 2008g: Beobachtungsprotokoll LUG 1. 2008

LUG 2008i: Beobachtungsprotokoll LUG 1. 2008

LUG 2008j: Beobachtungsprotokoll LUG 1. 2008

LUG 2008k: Beobachtungsprotokoll LUG 1. 2008

LUG 2008l: Beobachtungsprotokoll LUG 1. 2008

- KSP 2008d:** Beobachtungsprotokoll meines Co-Beobachters (3. Keysigning Party). 2008
- A10 2008:** Gedächtnisprotokoll nach einem Besuch in einer Privatwohnung. 2008
- Fir 2008a:** Gedächtnisprotokoll von der Arbeit in einer IT-Abteilung. 2008
- Fir 2008b:** Gedächtnisprotokoll von der Arbeit in einer IT-Abteilung. 2008
- Fir 2008c:** Gedächtnisprotokoll von der Arbeit in einer IT-Abteilung. 2008
- Fir 2008d:** Gedächtnisprotokoll von der Arbeit in einer IT-Abteilung. 2008
- Fir 2008e:** Gedächtnisprotokoll von der Arbeit in einer IT-Abteilung. 2008
- Fir 2008f:** Gedächtnisprotokoll von der Arbeit in einer IT-Abteilung. 2008
- A9- 2008:** Gesprächsprotokoll. 2008
- Int 2008a:** Interview zum Keysigning. 2008
- Int 2008b:** Interview zum Keysigning. 2008
- LUG 2009b:** Beobachtungsprotokoll LPI-Vorbereitung. 2009
- LUG 2009c:** Beobachtungsprotokoll LPI-Vorbereitung. 2009
- LUG 2009d:** Beobachtungsprotokoll LPI-Vorbereitung. 2009
- LUG 2009e:** Beobachtungsprotokoll LUG 1. 2009
- LUG 2009f:** Beobachtungsprotokoll LUG 1. 2009
- LUG 2009g:** Beobachtungsprotokoll LUG 1. 2009
- LUG 2009h:** Beobachtungsprotokoll LUG 1. 2009
- LUG 2009i:** Beobachtungsprotokoll LUG 1. 2009
- LUG 2009j:** Beobachtungsprotokoll LUG 1. 2009
- LUG 2009l:** Beobachtungsprotokoll LUG 3. 2009
- Ch2 2009:** Chat mit LUG-Mitglied. 2009
- Int 2009:** E-Mail-Interview mit Teilnehmer eines Hacking-Contests. 2009
- Fir 2009:** Gedächtnisprotokoll von der Arbeit in einer IT-Abteilung. 2009
- Inf 2010:** Gespräch mit Informant. 2010

Bildnachweis

0.1. T-Shirt der <i>Free Software Foundation Europe</i> : https://fsfe.org/order/2010/images/tshirt-fsfs-orangefront.jpg , Copyright © 2001-2011 Free Software Foundation Europe Abruf: 11.12.2010	iv
2.1. Wohnsitze von FLOSS-Entwickler_innen: Country of Residence or Work of OS/FS Developers, Ghosh et al. (2002: 17), Copyright ©2002 International Institute of Infonomics, University of Maastricht	21
2.2. Teamstruktur eines FLOSS-Projektes: Crowston/Howison (2004: 7)	26
2.3. Maskottchen und Logos von FLOSS-Communities: (a) GNU: https://commons.wikimedia.org/wiki/File:Heckert_GNU.png , Autor_innen: Aurélio A. Heckert, basierend auf Peter Gerwinski, basierend auf Etienne Suvasa. Copyright: Free Art License (http://artlibre.org/licence/lal/en). (b) Tux: https://commons.wikimedia.org/wiki/File:Tux.svg , Autor_innen: Larry Ewing, Simon Budig, Anja Gerwinski. Copyright: The copyright holder of this file allows anyone to use it for any purpose, provided that the copyright holder is properly attributed. Redistribution, derivative work, commercial use, and all other use is permitted. (c) Debian: http://www.debian.org/logos/index.de.html , Autor: Raul Silva, Copyright: ©1999 Software in the Public Interest, Inc., Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 3.0 Unported (https://creativecommons.org/licenses/by-sa/3.0/deed.de), Abruf: 22.10.2013	38
4.1. Geburtstags Einladung in <i>Python</i> -Syntax: Screenshot aus einer E-Mail an die Autorin	70
4.2. Versteckter Humor in Software: eigener Screenshot	71
6.1. Textbasiertes Arbeiten: Eigener Screenshot	148
6.2. Logo eines Community-Events: Autor: Sirko Kemter. Mit freundlicher Genehmigung der Pressestelle der Technischen Universität Chemnitz (http://idw-online.de/pages/de/image85252), Abruf: 17.11.2013	153

6.3. Textbasierte und grafische Editoren: eigener Screenshot	154
6.4. Humor und Ästhetik: Kühe in ASCII-Art: eigener Screenshot	169
6.5. Suche nach WLAN-Netzen: eigener Screenshot	172
7.1. GnuPG-Schlüssel: eigener Screenshot	213
7.2. Teilnehmer_innenliste einer Keysigning-Party: eigener Screenshot	215
7.3. Ansicht eines signierten Schlüssels auf einem öffentlichen Keyserver: eigener Screenshot	216
7.4. Keysigning-Party 2007: eigenes Foto	226
7.5. <i>Web of Trust</i> vor und nach einer Keysigning-Party: Autor: Jens Kubieziel, Quelle: Website einer Keysigning-Party, Link bleibt wegen Anonymisierung verborgen.	235

A. Glossar

Access Point ist ein Gerät, das den Zugang zu einem Funknetzwerk (WLAN) ermöglicht.

apt-get ist eine Software, mit der man Programme installieren und deinstallieren kann (siehe auch → *Paketverwaltung*).

ASCII-Art ist die Kunst, Bilder mit den Zeichen darzustellen, die die Computertastatur zur Verfügung stellt. Sie stammt aus der Zeit, in der es noch keine grafischen Oberflächen für PCs gab. ASCII ist der Name des Zeichensatzes; er steht für *American Standard Code for Information Interchange*.

Backdoor „Hintertürchen“, durch das sich Unbefugte die Kontrolle über einen Computer verschaffen können

Bootloader Software, die beim Anschalten eines Rechners eine Auswahl der installierten Betriebssysteme anbietet und den Betriebssystemkern des ausgewählten Systems lädt

BSD steht für *Berkeley Software Distribution* und ist eine Version des Betriebssystems Unix. Sie wurde an der Universität Berkeley entwickelt. Die BSD-Lizenz unterscheidet sich von der → *GNU General Public License*: Entwickler_innen sind weder dazu verpflichtet, den → *Quellcode*, den sie verändert haben, wieder mit zu veröffentlichen, noch dazu, ihr → *Derivat* wieder unter eine BSD-Lizenz zu stellen. Eine proprietäre oder kommerzielle Weiterverwendung dieser Software ist damit möglich.

BSSID steht für *Basic Service Set Identifier*. Die BSSID ist in diesem Zusammenhang die eindeutige Bezeichnung einer WLAN-Funkzelle.

Bug Jam ist ein Termin für die Fehlerbehebung in freier Software. Die Idee ist, dass sich möglichst viele Personen gleichzeitig an die Arbeit machen und dabei über das Internet kooperieren.

Bugreport technischer Fehlerbericht, der an Softwarewickler_innen geschickt wird

Bugtracker Software, mit der in der Softwareentwicklung der Entwicklungsstand eines Projektes dokumentiert wird. Programmfehler, Verbesserungsvorschläge und geplante „Meilensteine“ werden dort zentral erfasst. Bugtracker freier Software sind oft als Websites öffentlich sichtbar, so dass nachvollziehbar ist, auf welchem Stand die Entwicklung ist und wer wofür zuständig ist.

Client ist das Gegenstück zum → *Server*, nämlich ein Computer, der über das Netzwerk Dienste anfordert, die von Servern angeboten werden. Die Bezeichnung wird auch für einzelne Anwendungen benutzt, die Dienste abrufen: Ein E-Mail-Client ruft E-Mails vom Server ab.

Content-Management-System (CMS) Software für die Präsentation von Inhalten im Internet. Die meisten CMS haben ein Redaktionssystem, das das Einstellen und Gliedern von Inhalten für Personen ohne besondere technische Kenntnisse erleichtert. Bekannte freie CMS sind *Typo3*, *Joomla* oder *Drupal*.

Dateisystem die Art, wie die Ablage von Daten auf einem Datenträger (z. B. einer Festplatte) organisiert ist

debconf ist das jährliche Treffen von → *Debian*-Entwickler_innen. Es findet jedes Jahr an einem anderen Ort auf der Welt statt.

Debian ist eine der größten, ältesten und verbreitetsten → *Distributionen*. Das Projekt wurde 1993 von Ian Murdock gegründet und bekennt sich klar dazu, nur freie Software in *Debian* aufzunehmen.

Derivat bedeutet „Ableitung“. In Bezug auf Software sind Derivate getrennte Weiterentwicklungen, die aus dem → *Quellcode* eines Programmes entstehen.

Desktop hat drei verschiedene Bedeutungen: Erstens bezeichnet der Begriff einen *Personal Computer* in Abgrenzung zu mobilen Endgeräten oder → *Servern*. Zweitens ist es eine generelle Bezeichnung für die grafische Benutzeroberfläche. Drittens ist der Desktop ein Ordner mit persönlichen Dateien, der auf dem Bildschirm als Hintergrund einer grafischen Benutzeroberfläche dargestellt wird.

Digital Rights Management bezeichnet Verfahren, mit denen Hersteller die Nutzung und Verbreitung von digitalen Medien kontrollieren können.

Distribution Die Anzahl freier Programme ist unglaublich groß. Eine Linux-Distribution

ist eine ausgewählte Zusammenstellung freier Software z. B. auf einer Installations-CD. Jede Linux-Distribution beruht auf dem Linux-Kernel und beinhaltet eine Standardbasis an Programmen. Was darüber hinaus in einer Distribution enthalten ist, hängt davon ab, für welche Zielgruppe sie bestimmt ist. Auf jeder Distribution kann auch jede nicht enthaltene freie Software nachinstalliert werden.

DOS steht für *Disk Operating System*. DOS-Betriebssysteme wurden ab den 1960er Jahren entwickelt. Sie konnten im Unterschied zu den früheren *Tape Operating Systems* Informationen auf Festplatten und Disketten verwalten.

Feature Request Bitte an Programmierer_innen, eine gewünschte Funktionalität in ein Programm einzubauen

Fingerprint eine eindeutige Prüfsumme zur Identifikation einer Datei

Flamewar beleidigendes, unsachliches Redeverhalten im Internet, z. B. auf Mailinglisten

Funktionstaste die meistens zwölf Tasten (F1-F12) über der Haupttastatur, die je nach Programm unterschiedliche Funktionen haben oder selbst belegt werden können

Geek ist eine umgangssprachliche Bezeichnung für Menschen, die sich im besonderem Maße für ein Thema interessieren, im Zusammenhang dieser Arbeit für Computer oder Linux

Git ist eine Software zur → *Versionskontrolle*.

Gnome ist eine grafische Benutzeroberfläche.

GNU Das GNU-Projekt wurde 1983 von Richard Stallman gegründet und hatte die Entwicklung des ersten vollständig freien Betriebssystems zum Ziel. Es steht unter der → *GNU General Public License*. GNU steht für *GNU's not Unix*.

GNU General Public License (GPL) Die GNU GPL ist die am häufigsten verwendete Lizenz für freie Software. Sie wurde 1989 von Richard Stallman und ab 1991 von der *Free Software Foundation* entwickelt. Sie legt fest, dass die mit ihr lizenzierten Programme für jeden Zweck verwendet, kopiert und verändert werden dürfen und einschließlich des Quellcode weitergegeben werden müssen. → *Derivate* müssen wieder unter die GPL gestellt werden, damit sie frei bleiben.

grml ist eine → *Distribution*, die vor allem als → *Live-CD* für Datenrettungsaktionen verwendet wird.

Hacking-Contest Bei dem Hacking-Contest auf dem Linuxtag messen sich Teams darin, auf Linux-Rechnern Sicherheitslücken einzubauen und die Lücken der anderen Teams zu finden und zu beseitigen.

History → *Kommandozeilen* wie die *bash* erstellen eine History; das sind die Befehle, die zuletzt eingegeben wurden. Nutzer_innen können sie mit Tastenkombinationen schnell wieder aufrufen, anstatt ganze Befehle nochmals einzutippen.

IP-Adresse Adresse eines Rechners bzw. seiner Netzwerkkarte in einem Netzwerk

IRC *Internet Relay Chat* ist eine Kommunikationsform im Internet, bei der die Teilnehmer_innen sich über Textnachrichten unterhalten, die sofort übermittelt werden.

Jabber ist eine Form von Instant Messaging/Chat.

Kernel ist der Kern eines Betriebssystems, das Grundsystem, das direkt auf die Hardware zugreift.

Keysigning Im Internet verwenden Kommunikationspartner_innen digitale Schlüssel, um sich zu authentifizieren oder um verschlüsselt zu kommunizieren. Beim Keysigning bestätigen sie sich die Echtheit der Schlüssel durch digitale Signaturen. Keysigning stellt damit die Verbindung zwischen Person, E-Mail-Adresse und Schlüssel her.

Knoppix-CD eine → *Live-CD* mit der → *Distribution* Knoppix, die zur Datenrettung und zur Reparatur beschädigter Linux-Installationen verwendet wird

Kommandozeile Software, mit der man den Rechner direkt durch die Eingabe von Befehlen steuern kann

Kommentar Konfigurationsdateien, Quellcode oder auch (Shell-)Skripte können Kommentare enthalten, die für Leser_innen der Dateien bestimmt sind, ohne dass der Rechner sie auswerten soll. Sie werden durch bestimmte Zeichen am Zeilenanfang gekennzeichnet, oftmals durch eine Raute (#). Man spricht von aus- bzw. einkommentierten Zeilen.

Konsole Kommandozeile, in die Befehle eingegeben werden können. Auf Linuxrechnern wird zwischen (den nicht-grafischen) virtuellen Konsolen und (grafischen) X-Terminals

unterschieden. Nach dem Login in einer Konsole wird dort automatisch eine → *Shell* gestartet.

Kubuntu ist eine → *Distribution*, die → *Ubuntu* sehr ähnlich ist, aber eine andere grafische Benutzeroberfläche anbietet.

Linux Professional Institute (LPI) Das *Linux Professional Institute* bietet Zertifizierungen für Linux-Systemadministrator_innen an. In Deutschland werden die Prüfungen vom Verein *LPI e. V. German* abgenommen

Live-CD eine CD mit einer Linux-Distribution, von der man den Rechner starten kann, ohne dass Linux auf der Festplatte installiert werden muss

MAC-Adresse die eindeutige Hardware-Adresse einer Netzwerkkarte

Mailserver Rechner, der E-Mails empfängt und in die Postfächer der Empfänger_innen weiterleitet

man page kommt von *manual page*. *Man pages* sind textbasierte Anleitungen zu Unix- und Linux-Programmen.

Mandriva ist eine → *Distribution*.

Monitoring -Software überprüft automatisiert, ob andere Software funktioniert und über das Netzwerk erreichbar ist. Sie verschickt Benachrichtigungen, wenn dies nicht der Fall ist.

Network-Manager Software, mit der die Netzwerkverbindungen eines Computers verwaltet werden können

OpenSolaris ist ein freies Betriebssystem der Firma *Sun Microsystems*.

OpenSUSE ist eine Distribution.

Pair Programming bezeichnet eine Programmierpraxis, bei der zwei Softwareentwickler_innen gemeinsam und gleichzeitig nach dem Vier-Augen-Prinzip an der Software arbeiten.

Paketverwaltung Software, die die installierte und installierbare Software für ein Linuxsystem organisiert. Sie überprüft Abhängigkeiten zwischen Programmen und installiert Aktualisierungen.

Partition Eine Festplatte kann in mehrere Partitionen (Abschnitte) unterteilt werden, auf denen dann verschiedene Betriebssysteme installiert werden können.

Patch Ein Patch („Flicken“) ist eine nachgelieferte Reparatur-Komponente oder Ergänzung für Software. Damit werden Fehler behoben, Sicherheitslücken geschlossen oder in kleinem Umfang neue Funktionen eingebaut.

ping ist ein Kommandozeilenprogramm zur Überprüfung, ob Rechner über das Netzwerk miteinander verbunden sind.

Quellcode (oder Quelltext, Programmcode) ist Text in einer Programmiersprache, in dem Computerprogramme geschrieben sind. Maschinen können Quellcode nicht verarbeiten. Er wird deshalb kompiliert und liegt danach in binärer Form vor (Nullen und Einsen), die Menschen nicht mehr entziffern können. Wenn Programme ausschließlich als Binärpakete weitergegeben werden, ist ihre Machart für Menschen nicht transparent.

Repository In Bezug auf Software ist ein Repository (zu deutsch: Paketquelle) ein zentrales Verzeichnis auf einem Server, in dem Software zum Download bereitgestellt wird. Die großen → *Distributionen* enthalten eine Standard-Liste mit Repositories, zu denen bei Bedarf andere hinzugefügt werden können, von denen Software bezogen werden soll.

Request for Comments (RFC) sind Dokumente, in denen technische und organisatorische Fragen bezüglich des Internet spezifiziert sind. Ursprünglich waren sie tatsächlich als „Bitten um Kommentare“ zu einem Vorschlag gemeint. Inzwischen sind in den Dokumenten Standards definiert, der Name ist jedoch geblieben.

root ist der Administrationsaccount auf einem Linuxrechner.

Router sind Geräte, die Computernetzwerke verbinden oder voneinander trennen. Sie leiten den Datenverkehr an die Empfängeradressen weiter.

Routingtabelle Hier wird eingestellt, wohin ein → *Router* welche Daten senden soll.

screen In dem hier beobachteten Zusammenhang wird *screen* als Kooperationswerkzeug verwendet, damit jede Person auf ihrem eigenen Bildschirm sehen kann was die anderen auf dem Server gerade eintippen.

Server sind sowohl vernetzte Computer als auch Programme, die anderen Rechnern oder Programmen (→ *Clients*) Dienste zur Verfügung stellen, z. B. Websites, E-Mail-

Übermittlung oder Speicherplatz.

Shell textbasierte Software, die Befehle interpretiert, mit denen der Rechner gesteuert werden kann. Shells laufen innerhalb von → *Konsolen*.

Shellskript ein Programm, das in einer Skriptsprache geschrieben ist und kleinere Programmieraufgaben löst. Skripte werden z. B. von Systemadministrator_innen für die Automatisierung wiederkehrender Aufgaben programmiert.

Softkey Taste, die ihre Funktion ändert, je nach dem, was auf einem Bildschirm angezeigt wird. Ein Beispiel sind die Tasten an Geldautomaten: Dieselbe Taste kann zuerst für die Auswahl der Sprache, dann für die Auswahl der Tätigkeit, dann für den gewünschten Geldbetrag genutzt werden.

Softwareagenten sind Programme, die aufgrund bestimmter Ereignisse „eigenständig“ in Aktion treten. Ein Beispiel ist die Büroklammer in *Microsoft Word*, die Hilfe anbietet, nach der man nicht gefragt hat.

ssh steht für *secure shell* und ist sowohl ein Programm als auch ein Netzwerkprotokoll. Es dient dazu, sich über das Netzwerk verschlüsselt auf einem anderen Computer einzuloggen. Dabei kann man sich mit einem Passwort oder mit einem autorisierten privaten Schlüssel authentifizieren.

ssh-Daemon Ein Daemon ist ein Dienst, der auf einem Rechner im Hintergrund läuft und auf Arbeit wartet. Der ssh-Daemon wartet auf Login-Versuche über *ssh*.

SSL steht für *Secure Socket Layer* und bezeichnet ein Verschlüsselungsprotokoll. Viele Websites bieten SSL-Verschlüsselung an, damit Besucher_innen die Daten verschlüsselt abrufen können.

Switch Ein Switch ist ein Gerät, in das Netzkabel gesteckt werden können, um Computer oder Computernetze miteinander zu verbinden.

Theme Die grundsätzliche Gestaltung der grafischen Benutzeroberflächen wird durch Themes vorgenommen. Ein Theme beinhaltet meistens Elemente für das Aussehen der Fensterrahmen, Icons, evtl. Schriftarten.

Typo3 ist ein freies → *Content-Management-System*.

Ubuntu ist eine → *Distribution*.

Usability bedeutet Benutzerfreundlichkeit oder Bedienbarkeit von Software.

User In diesem Zusammenhang ist „User“ als das Gegenteil von root, dem Administrationsaccount, gemeint. Während root alle Rechte auf dem Rechner hat, sind die Zugriffsrechte von User_innen stark eingeschränkt. Man spricht deshalb von „unprivilegierten“ User_innen.

Versionskontrolle Ein Versionskontrollsystem stellt sicher, dass die Beteiligten bei gemeinsamer Arbeit an einer Software die Änderungen der anderen wahrnehmen und immer an der aktuellen Version einer Datei weiterarbeiten. Auf alle älteren Versionen kann dabei zurückgegriffen werden.

wakeonlan Software, mit der Rechner über das Netzwerk hochgefahren werden können (ohne sie per Schalter einzuschalten).

Webbrowser Ein Webbrowser, kurz Browser, ist ein Programm, mit dem man im Internet surfen kann, z. B. Firefox.

Weblog Ein Weblog, kurz auch Blog, ist eine Software, mit der Websites erstellt werden können, die (wie ein Logbuch) einen Schwerpunkt auf der chronologischen Reihenfolge von Einträgen haben. Oft werden sie deshalb als Online-Tagebücher bezeichnet.

Webserver Software, die auf einem → *Server* Websites zur Verfügung stellt. Der verbreitetste Webserver heißt *Apache*.

Wiki Ein Wiki ist eine Software, die meistens für die gemeinschaftliche Sammlung und Präsentation von Inhalten auf Websites genutzt wird. Das bekannteste Beispiel ist die freie Enzyklopädie *Wikipedia*.

X-Server Der X-Server stellt die grafische Benutzeroberfläche zur Verfügung.

X-Terminal Ein X-Terminal(emulator) ist eine → *Konsole*, die innerhalb von einer grafischen Oberfläche als „Fenster“ gestartet wird.

B. Forschungsmethoden

Eine Ethnografie ist keine fertige Methode, sondern eine Haltung gegenüber dem Forschungsfeld, die weiter oben erläutert wurde. Sie bedarf einer Präzisierung der gewählten Erhebungs- und Auswertungsmethoden.

B.1. Datenerhebung

Teilnehmende Beobachtung

„Ethnografische Verfahren werden (...) eingesetzt, um implizite Aspekte der Arbeitspraktiken ans Licht zu bringen. Sie zielen darauf ab, für selbstverständlich gehaltene Vorannahmen, Überzeugungen und gemeinsam geteilte Glaubensformen aufzudecken und zu problematisieren, und sie machen auf die Diskrepanz zwischen ‚offiziellen‘ Modellen und Sichtweisen der Praktiken und dem, was tatsächlich geschieht, aufmerksam“ (Schmidt 2008: 285).

Robert Schmidt fasst hier gut das Anliegen zusammen, das eine Untersuchung habitueller Praktiken bedeutet. In der qualitativen Sozialforschung wird davon ausgegangen, dass gewichtige Anteile dieser Selbstverständlichkeiten durch Interviews kaum zu erheben sind. Die Regeln, nach denen im Forschungsfeld kommuniziert wird, alltägliche Praktiken und Grundannahmen werden meistens in Interviews nicht benannt, sie müssen den Befragten nicht unbedingt bewusst sein (vgl. Schöne 2005: 194). Aus diesem Grund habe ich den größten Teil der empirischen Daten durch teilnehmende Beobachtung gewonnen. Dazu habe ich zwischen Herbst 2007 und Ende 2009 Community-Treffpunkte und -Events besucht. Die teilnehmende Beobachtung war offen angelegt, d. h. viele Anwesende waren über meine Forschung informiert. Bei Treffen mit großer Fluktuation war dies jedoch nicht laufend zu bewerkstelligen, so dass Personen, die einmalig oder mittendrin auftauchten oder nicht direkt in den beobachteten Kleingruppen waren, nicht unbedingt informiert waren.

Feldzugang

Meinen ersten Kontakt zum Feld hatte ich durch eine Stelle, bei der ich als Veranstaltungsorganisatorin in einem Community-Treffpunkt arbeitete. Dort entstand auch die Idee für diese Arbeit. Den Feldzugang zu den Linux User Groups habe ich über regelmäßige Teilnahme an den Treffen ab Herbst 2007 allmählich hergestellt. Zu Beginn der Beobachtungen blickte ich bereits auf eine vierjährige private Beschäftigung mit Linux zurück, in der ich ein Grundlagenwissen erworben hatte, das mir den Einstieg in die LUG mit ermöglichte. Nicht nur, um den Feldzugang herzustellen, sondern auch weil relevante Daten im Kontakt mit den Menschen im Forschungsfeld entstehen, war es nötig, regelmäßig die Beobachterinnenperspektive zu verlassen und mich in die auftauchenden technischen Fragen involvieren zu lassen. Ethnograf_innen beeinflussen demnach immer ihr Feld, was jedoch nicht als störender Effekt, sondern als Quelle von Beobachtungen und Erkenntnis gesehen wird (vgl. Emerson et al. 1995: 3 f.).

Beobachtungsprotokolle

Die ersten Beobachtungsprotokolle waren oftmals Gedächtnisprotokolle, die späteren basierten auf recht präzisen Notizen während der Beobachtung, die direkt im Anschluss ausformuliert wurden. Der Korpus besteht aus rund 50 Protokollen und Notizen sowie einzelnen Chat-Logs und E-Mails. Im gesamten Beobachtungszeitraum gab es Abende, an denen ich eher wie ein Gruppenmitglied dabei war und mich gemeinsam mit den Anwesenden mit der Lösung technischer Probleme beschäftigt habe. Zudem gab es Situationen, in denen ich es unpassend fand, ein Notizbuch herauszuholen und damit quasi einen sichtbaren Rollenwechsel zu vollziehen, der mich von den Anwesenden distanzieren würde. Auch solche Situationen wurden aus dem Gedächtnis protokolliert. Zudem habe ich in der frühen Erhebungsphase wertvolle Anregungen für das weitere Protokollieren beim Magdeburger Methodenworkshop (Februar 2008) erhalten, wo ich ein Protokoll zur Diskussion gestellt hatte.

Halboffene Leitfaden-Interviews

Zusätzlich habe ich zahlreiche Informationsgespräche mit Informant_innen und vier Leitfaden-Interviews geführt. Die ersten beiden Interviews — zu Beginn der Erhebung

gen — drehten sich inhaltlich stark um den Community-Begriff und dienten mir zur Aufspürung von Erhebungskontexten. Die beiden anderen Interviews habe ich mit Organisator_innen von Key signing-Parties geführt, um das Ritual „von innen heraus“ erklärt zu bekommen. Die Interviewten nahmen dadurch die Stellung von Expert_innen ein.

Eklektisches Hinzuziehen von Beobachtungen aus dem Internet

Einige Beobachtungen konnte ich nicht explizit protokollieren. Sie kamen mir zwar als wiederkehrende Praktiken oder Redewendungen vor, stellten sich jedoch während meiner Beobachtungen nicht ein. Stellenweise habe ich für entsprechende Zitate oder für Illustrationen deshalb Websites konsultiert, die das Gemeinte treffend wiedergeben.

Orte der Datenerhebung

Im Mittelpunkt stand die teilnehmende Beobachtung der Treffen einer größeren Linux User Group. Dort habe ich im Beobachtungszeitraum unterschiedlichen Vereinsaktivitäten beigewohnt (Einsteiger_innen-Treffen, Prüfungslerngruppe, Mitgestaltung eines Community-Events). Dazu habe ich punktuelle Beobachtungen in zwei anderen Gruppen hinzugezogen.

Linux User Groups

Die LUG im Zentrum der Erhebungen existiert als Gruppe seit Mitte der 1990er Jahre und ist seit den frühen 2000er Jahren als gemeinnütziger Verein eingetragen.¹ Der Verein hat rund 60 formale Mitglieder, von denen nach Aussage von Vorstandsmitgliedern nur ein Kern von schätzungsweise 10–15 Personen aktiv ist, während die anderen zahlende, stille Mitglieder sind. Die abendlichen Treffen zum technischen Support werden nicht nur von Vereinsmitgliedern besucht, sondern sie sind in der Szene eine bekannte Anlaufstelle. Demnach gab es zwischen den einzelnen Abenden eine deutliche Fluktuation von Teilnehmenden, sowohl was die Personen als auch was die Anzahl der Anwesenden angeht; letztere schwankte zwischen zehn und rund 30–40 Personen. Die Vereinsmitglieder waren nicht zu „erkennen“. Neben den Support-Treffen habe ich einzelne Vorträge beobachtet

¹Die LUG wird aufgrund der Anonymisierung des Materials nicht namentlich genannt. Zur Notwendigkeit der Anonymisierung von Datenquellen aus dem Internet vgl. Krieger (2004).

sowie sechs Treffen der LPI-Lerngruppe beobachtet. Diese Gruppe traf sich über einen längeren Zeitraum zur Vorbereitung auf die Zertifizierungsprüfungen des → *Linux Professional Institute*. Weitere Vereinsaktivitäten wie einen Linux-Einsteiger_innen-Kurs oder die eigenen Software-Projekte der LUG habe ich ausgespart.

Die zweite Gruppe ist ein wesentlich kleinerer nicht institutionalisierter Zusammenschluss von Interessierten. Dort fanden Treffen mit Kurzvorträgen und solche zur gemeinsamen Beschäftigung mit technischen Problemen oder Herausforderungen statt. Die dritte Gruppe war eine User Group einer speziellen → *Distribution*, die ebenfalls als Anlaufstelle fungiert. Dort habe ich jedoch einen so genannten → *Bug Jam*, ein Treffen zum gemeinsamen Beheben von Fehlern in Software, beobachtet ebenso wie eine *Release Party*. Release Parties sind Veranstaltungen, die zur Veröffentlichung einer neuen stabilen Version einer Distribution stattfinden. Auf diesem Community-Event gibt es Vorträge über die Neuerungen, Installations-Medien und Fanartikel.

Community-Events

Community-Events sind gruppenübergreifende Szenetreffen. Ich habe verschiedene Events besucht: den „großen“ viertägigen *Linuxtag* in Berlin (2007–2009), die *Chemnitzer Linuxtage*, die jeweils ein Wochenende dauern (2007–2009), den eintägigen *Brandenburger Linux-Infotag (BLIT)* in Potsdam (2008) und zwei Veranstaltungen des *Chaos Computer Clubs (CCC)*, nämlich den *Chaos Communication Congress* 2007 und das *Chaos Communication Camp* 2007.

Aus dem meist mehrtägigen Programm wurden gezielt einzelne Programmpunkte für die teilnehmende Beobachtung ausgewählt. Dazu zählten die *Social Events*, drei Keysigning-Parties und ein → *Hacking-Contest*. Die Keysigning-Parties und den Hacking-Contest habe ich offen teilnehmend beobachtet, also auch vor Ort Notizen gemacht, Gespräche und Interviews geführt. Auf eine Keysigning-Party begleitete mich ein Co-Beobachter, der ein eigenes Beobachtungsprotokoll verfasste. Die übrigen Beobachtungen auf Community-Events wurden quasi „verdeckt“ in der Masse durchgeführt und in Gedächtnisprotokollen festgehalten. Bei diesen Beobachtungen habe ich die Perspektiven dadurch gewechselt, dass ich teilweise als Besucherin oder als Helferin an einem Stand an den Events teilgenommen habe.

IT-Abteilung: Systemadministration im Arbeitsalltag

Zusätzlich zu der teilnehmenden Beobachtung an typischen Community-Treffpunkten habe ich von Januar 2008 bis Januar 2010 eine Nebentätigkeit in einer IT-Abteilung aufgenommen. Dort waren mehrere in Community-Projekten Engagierte tätig. Die Tätigkeit bot mir die Möglichkeit, die tägliche Arbeit von Linux-Systemadministrator_innen zu beobachten. Die Abteilung betreute ein Rechenzentrum für ein mittelständisches Unternehmen, in dem etwa 60 Linux-Server liefen, und gewährleistete Support für die Mitarbeiter_innen des Unternehmens. Aus diesen Beobachtungen entstanden Gedächtnisprotokolle.

B.2. Datenauswertung

Die Datenauswertung lässt sich in Ethnografien nicht streng von der Datenerhebung trennen, da bereits das Protokollieren durch den Explikationszwang oft Teil der Analyse ist (vgl. Amann/Hirschauer 1997: 28). Demnach hatte die Datenauswertung verschiedene Phasen.

Interpretation in Colloquien und Forschungswerkstätten

In der frühen Phase der Erhebungen habe ich verstärkt Colloquien und Forschungswerkstätten aufgesucht, um Anregungen zum Protokollieren, aber auch Interpretationsansätze anderer einzubeziehen. Dies waren die beiden Colloquien meiner Gutachter_innen, der Magdeburger Methodenworkshop (Februar 2008), die Nachwuchstagung des *Arbeitskreises Politik, Wissenschaft und Technik* in der *Deutschen Vereinigung für Politische Wissenschaft* (Oktober 2008) sowie der Methodenworkshop für Promovierende in der Rosa-Luxemburg-Stiftung (April 2009). Der Austausch auf diesen Foren ist aus verschiedenen Gründen wichtig: Was die Beobachtungsprotokolle angeht, habe ich im Sinne eines theoretischen Sampling nach ersten ausgewerteten Protokollen die Beobachtung fokussiert (vgl. z. B. Strübing 2004: 30 ff.). So rückte beispielsweise der Small Talk bei den Key signing-Parties erst bei den späteren Beobachtungen in den Mittelpunkt meines Interesses, während ich zuvor vor allem den Ablauf des Rituals dokumentiert hatte. Darüber hinaus führten mir die akademischen Gesprächspartner_innen meinen Prozess des „going native“ vor Augen und zwangen mich zu einer erneuten Übersetzung des Beobachteten, auf

das sie neue, feldfremde Perspektiven warfen (vgl. ebd.). Die gemeinsame Interpretation in der Forschungswerkstatt ist nicht zuletzt hilfreich, um die eigene Interpretation zu untermauern oder zu hinterfragen, obwohl Reliabilität in Ethnografien eher eine Anforderung an das „Gesamtporträt‘ einer Szenerie“ (ebd. 31) ist.

Offenes Kodieren und Veranschaulichung in Mindmaps

Das komplette Forschungsmaterial besteht aus rund 50 Beobachtungsprotokollen und Notizen, vier Interviews und zusätzlichem Material wie Weblinks oder Bilder. Es wurde „offen“ kodiert, also mit möglichst wenig theoretischen Vorannahmen. Das offene Kodieren ist ein Verfahren, das der Grounded Theory entlehnt ist. Es hat das Ziel, überhaupt erst einmal einen Zugang zu den Daten zu bekommen, der eher breit ausfällt (Strübing 2004: 19 ff.). Dazu habe ich die Software *MaxQDA* verwendet. Der Kodebaum enthielt vorab die Kategorien aus meinen Hypothesen: Herkunft, Klasse, Geschlecht, „soziale Beziehungen“, Sprache, Wissen und Technologien. Aus der Vielzahl von vergebenen Codes habe ich die Liste der Kategorien ergänzt z. B. um Alter, Körper, Zeit, aber auch Beobachterin. Eine Kategorie umfasst jeweils diverse Codes. Während Codes textnahe Schlagworte sind, dienen Kategorien der Theoriegenerierung (vgl. Berg/Milmeister 2007: 186 f.).

Zitatauswahl und Entwicklung der Storyline

Aus dem dabei entstandenen Kodebaum habe ich dann eine Auswahl der zu analysierenden Szenen getroffen: Zum einen begünstigt *MaxQDA* eine quantitative Perspektive auf das Material. Dadurch rückten die am häufigsten vergebenen Codes in den Vordergrund, etwa „Nutzung textbasierter Programme“, „Lehrer_in-Schüler_in“ (Kleingruppenarbeit), „Warten“ oder „Ausprobieren“. Um mir Zusammenhänge zwischen den vergebenen Codes zu vergegenwärtigen, habe ich kein weiteres Kodierverfahren angewendet, sondern — auch anstelle von Memos — Mindmaps aus dem Kodebaum erstellt, die von diesen Häufungen ausgingen. In die Auswahl gingen auch Codes bzw. Zitate ein, die nicht durch Häufungen auffielen, sondern durch eine von mir wahrgenommene Dichte der Situation oder durch eine besondere Nähe zu den Fragestellungen. So fanden Systematik und assoziative Kreativität und Offenheit gleichermaßen Raum in der Auswahl (vgl. Berg/Milmeister 2007: 204). Die Interpretation der ausgewählten Zitate erfolgte in Teilen in den genannten Diskussionszusammenhängen, in Teilen durch das Heranziehen von existierender Literatur

und immer im Bewusstsein, dass mehr als *eine* Interpretation möglich ist. Der offensive Umgang mit Überraschungs- oder Befremdungseffekten, den Klaus Amann und Stefan Hirschauer beschreiben, war dabei zentral: Die Unterschiede zwischen meinem Wissen, meinen Annahmen und denen der Beobachteten, Fehlschläge von Beobachtungen und Verständnisprobleme habe ich gezielt zur Aufspürung von Relevanzen im Forschungsfeld genutzt (vgl. Amann/Hirschauer 1997: 19 ff.). Dadurch rückten Themen wie die Maßstäbe für Expertenwissen, die explorative Funktion des „Abschweifens“, der Umgang mit Zeit oder die Herstellung von Vertrauen durch Keysigning erst in den Blick.