

Rezension:

Matthias Groß

Niels C. Taubert (2006). Produktive Anarchie? Netzwerke freier Softwareentwicklung. Bielefeld: Transcript, 249 Seiten, ISBN: 3899424182 (paperback), EUR 27,80

Keywords: freie Software, Wissenschaftsforschung, Innovation, wissenschaftliches Ethos

Zusammenfassung: Unter freier Software werden Programme verstanden, deren Lizenz, Nutzung, Vervielfältigung, Verbreitung und Veränderung frei zugänglich ist. Sie wird allseits gelobt und gilt gewöhnlich als Alternative zur käuflichen Software. Wie kann es zu einem solchen Phänomen in einer Zeit kommen, in der fast alles nur noch käuflich erwerbbar erscheint? Das Buch *Produktive Anarchie? Netzwerke freier Softwareentwicklung* von Niels C. TAUBERT setzt sich zum Ziel, Voraussetzungen und Bedingungen der freien Softwareentwicklung soziologisch zu untersuchen. Ein Kernergebnis ist, dass der Entwicklungsprozess zum großen Teil als adaptiver, experimenteller Suchprozess verstanden werden muss, in dem die beständige Rückkopplung zwischen Produktions- und Anwendungskontext entscheidend für eine stabile und erfolgreiche Softwareentwicklung ist. Zu den erstaunlichen Ergebnissen zählt, dass zu den Voraussetzungen dieses Rückkopplungsprozesses eine Orientierung an Normen wie Neutralität, Universalität oder der allgemeinen Zugänglichkeit gehört, wie sie in den 1940er Jahren von Robert MERTON der Wissenschaft zugeschrieben wurden. Sollte TAUBERT mit seiner Beobachtung, dass sich diese Normen heute außerhalb der institutionalisierten Wissenschaft finden, Recht haben, kann seine Fallstudie als Indikator einer neuen Form der Wissensproduktion im 21. Jahrhundert verstanden werden, in der eine allgemeine soziale Relevanz und Verantwortung handlungsleitend ist. TAUBERT liefert damit zentrale Hinweise auf die Erklärung der erfolgreichen Organisation von Technikentwicklungsprojekten, die eine breitere Diskussion anregen sollte.

Inhaltsverzeichnis

- [1. Die Revolution des Internets und die Bedeutung der freien Software](#)
- [2. Stabilisierte Anarchie und das Ethos der freien Programmierung](#)
- [3. Ausblick: Wissenschaft und Softwareentwicklung im 21. Jahrhundert](#)

[Literatur](#)

[Zum Autor](#)

[Zitation](#)

1. Die Revolution des Internets und die Bedeutung der freien Software

Man kann heute getrost sagen, dass die Vermutungen über Revolutionen, die das Internet und alle damit verbundenen Neuerungen zur Veränderung der Welt noch Mitte der 1990er Jahre geschürt hatten, sich nicht erfüllt haben. Insbesondere hat sich die Angst, dass uns das Internet einsam machen würde, weil wir zunehmend vor der Mattscheibe sitzend uns zu Soziopathen entwickeln, vorerst als unbegründet erwiesen. Ebenso unwahrscheinlich scheint es nun, dass E-Mail und andere über den Computer ablaufende Kommunikationsformen uns

und unser Sozialleben grundsätzlich verändern, wie es noch in den frühen 1990er Jahren propagiert wurde (vgl. BAKARDJIEVA 2005; FRANZEN 2000; KATZ & RICE 2002). Eine der letzten großen Erwartungen auf radikale Änderung verbindet sich heute fast nur noch mit der so genannten "Open Source" Bewegung. Der Ausdruck Open Source wird meist auf Computer-Software angewandt und soll darauf verweisen, dass es jedem und jeder ermöglicht wird, nicht nur in den Quellcode eines Programms einsehen zu können, sondern diesen auch weiter zu geben und zu verändern. Mehr noch, der Quellcode muss allen offen stehen, die darauf zugreifen wollen. Dies ist das oberste Gebot. Mit diesem Gebot soll gewährleistet werden, dass die Software entsprechend der jeweiligen Interessen der Nutzer und Nutzerinnen verändert und verbessert werden kann. Die Nutzer und Nutzerinnen sind über den gesamten Globus verteilt und koordinieren ihre Beiträge zur Entwicklung über Mailinglisten. Diese Form der Entwicklung erfreut sich wachsender Beliebtheit. [1]

Niels C. TAUBERTs *Produktive Anarchie? Netzwerke freier Softwareentwicklung* analysiert als erstes soziologisches Buch die Organisation von Innovationsprozessen in solchen freien Softwareentwicklungsprojekten. Die Entwicklung von freier Software erscheint insbesondere wissenschaftssoziologisch interessant, da die Produktion von innovationsbezogenem Wissen, wie TAUBERT es nennt, unter widrigen Bedingungen stattfindet, denn die Entwicklung freier Software wird nicht von einer Organisation koordiniert, sondern zum größten Teil von Freizeitprogrammierer(inne)n betrieben. TAUBERT geht daher der Frage nach der sozialen Organisationsstruktur freier Softwareentwicklungsprojekte nach und verfolgt diese in zwei Richtungen. Zum einen wird nach der Entstehung der Struktur, zum anderen nach den sozialen Mechanismen gefragt, die eine Handlungskoordination der Projektmitglieder erlauben. Empirische Grundlage bilden dabei Dokumente aus der Entstehungsgeschichte der freien Softwareentwicklung, die Kommunikation auf einer Projektmailingliste ("archivierte Mailingliste"), sowie zwölf Interviews mit bekannten Softwareentwickler(inne)n. Diese Materialien wurden von TAUBERT mittels einer "qualitativ-hermeneutischen Interpretation" (S.122) analysiert. Er legt besonderen Wert auf drei Aspekte der Interpretationstechnik: 1, den zeitlichen Ablauf der Kommunikation, 2. die Einstiegssequenzen von Episoden sowie die Suche nach kontrastierenden Aussagen und 3. das Bestreben, einseitige Interpretationen auszuschließen. [2]

"Ziel dieser Analyse", so TAUBERT,

"ist es dabei, unterschiedliche Lesarten zu entwickeln und im Verlauf des weiteren Vorgehens Interpretationen auszuschließen und Evidenz für die übrig bleibende Lesart zusammenzutragen. Das Material wurde immer im Kontext seines Auftretens, also im Zusammenhang mit der vorhergehenden und folgenden Kommunikation interpretiert" (S.122). [3]

TAUBERTs Fokus auf die Anfangssequenzen von Episoden bedeutet, dass er sich auf neue Betreffzeilen als Hinweis auf neue Themen konzentriert. Dieses

Vorgehen leitete er von der Annahme ab, dass Eingangssequenzen für den weiteren Verlauf einer Kommunikation entscheidend sind. Zusätzlich versucht TAUBERT mit der gezielten Suche nach kontrastierenden Aussagen, die dem herausgearbeiteten Kommunikationsfluss zu widersprechen scheinen, der einseitigen Interpretation singulärer Ereignisse entgegenzuwirken. TAUBERT erwähnt in seiner Danksagung zwar die Mitarbeit in einer "Methodenwerkstatt" (S.9), aber das methodische Vorgehen bleibt alles in allem jedoch etwas undeutlich. Ein kurzer Verweis auf bekannte Methoden, an die er sich angelehnt hat, hätte genügt. Da dies nicht der Fall ist, hätte das methodische Vorgehen etwas genauer erläutert und begründet werden können, denn unter einer "qualitativ-hermeneutischen Interpretation" kann man sich sehr viel vorstellen. [4]

2. Stabilisierte Anarchie und das Ethos der freien Programmierung

Produktive Anarchie? gliedert sich in vier thematische Blöcke, wobei der erste auffälligerweise nicht als eigener Teil nummeriert wurde, sondern erst die drei Blöcke ab Seite 63. Im *ersten* nicht nummerierten Teil (S.11-62) werden die Grundlagen für die Untersuchung geliefert: Nach der begrifflichen Klärung zu "freier Software" und "freien Softwareentwicklungsprojekten" wird die Fragestellung der Untersuchung präzisiert, um daran anschließend das Konzept der Innovationsnetzwerke vorzustellen. Hierbei handelt es sich um ein Konzept aus der Bielefelder Schule der Innovationsforschung, das seit den 1980er Jahren entwickelt wurde, um interorganisationale Rückkopplungsprozesse zwischen Hersteller(inne)n und Anwender(inne)n zu erfassen (z.B. KOWOL 1998; ASDONK, BREDEWEG & KOWOL 1991; HERBOLD et al. 2002). Besonderer Fokus liegt hier auf der Generierung von innovationsrelevantem Wissen. Diese theoretische Rahmung bietet sich für das Beispiel der freien Softwareentwicklungsprojekte an, da es hier ebenfalls um die Organisation von Innovationsprozessen unter Beteiligung von heterogenen Akteuren geht. [5]

Im *zweiten* Teil des Buches unter dem Titel "Die Genese des Entwicklungsnetzwerks" (S.63-114) steht die Frage nach der Entstehung der besonderen Projektstruktur freier Softwareentwicklungen im Mittelpunkt. Ihre Beantwortung findet TAUBERT in der vertraglichen Innovation von freien Softwarelizenzen und der organisatorischen Innovation der offenen Programmierprojekte. Freie Softwarelizenzen generalisieren das Urheberrecht, indem sie allen Interessierten unbeschränkte Anwendung, Vervielfältigung, Vertrieb und Modifikation des betreffenden Programms gestatten. Durch diese Vertragsbedingungen wird freie Software sozusagen zum Kollektivgut, und zwar nicht nur in der zum Zeitpunkt der Lizenzierung vorliegenden Form, sondern in sämtlichen künftigen Versionen. Weiterer Effekt dieser Innovation ist, dass Beiträge zur Entwicklung zwangsläufig die Form von "Gaben", so TAUBERT, annehmen. Für Weiterentwicklungen können von Lizenznehmer(inne)n keine Gegenleistungen eingefordert werden. Durch diese spezifische Struktur soll die eskalierende Dynamik einer wechselseitigen Überbietung verhindert werden, die für andere Formen der Gabenökonomie typisch ist, bei denen die Transaktion von Dingen nur zwischen Personen erfolgt. TAUBERT stellt fest, dass die Gabe hier in Bezug auf ihre(n) Geber(in) personalisiert ist, nicht aber hinsichtlich der

Empfänger(innen). Interessant ist nun, dass sich die organisatorische Innovation der offenen Programmierprojekte gegen die von Software Engineering her bekannten Leitvorstellungen über die Organisation der Programmentwicklung abgrenzt. Während nach dieser Perspektive der Softwareentwicklungsprozess weitgehend formalisiert werden muss, um die hoch komplexen Anforderungen des Prozesses zu bewältigen, zeigt sich anhand der Diskussion zur Gründung des LINUX-Projekts, dass hier die Idee einer selbstorganisierten Projektarbeit auftritt. [6]

TAUBERT kommt mit der Analyse dieser Diskussion zum Schluss, dass sich die Entwickler(innen) an einem von Robert K. MERTON propagierten wissenschaftlichen Ethos orientierten. Dieses Ethos setzt sich aus den Normen des "organisierten Skeptizismus", des "Desinteresses" (Neutralität), der Universalität und des "Kommunismus" (allgemeine Zugänglichkeit) zusammen. Das ist auf den ersten Blick erstaunlich, denn MERTON hatte seine Normen bekanntermaßen unter dem Eindruck der Gräueltaten unter dem Deckmantel der Wissenschaft im Zweiten Weltkrieg formuliert (vgl. MERTON 1942), und die Wissenschaftsforschung nach dem Zweiten Weltkrieg hatte diese Normen längst als überholt oder als empirisch nicht haltbar enttarnt bzw. Alternativvorschläge unterbreitet (vgl. BARNES & DOLBY 1972; STEHR 1978; GARFINKEL 1967). Interessanterweise scheinen MERTONS Normen – man sollte betonen, dass auch MERTON tatsächlich von Normen spricht und nicht von festen Eigenschaften wissenschaftlicher Aktivitäten – auch heute noch in der Öffentlichkeit als Idealtypus und Verhaltenserwartung an die Wissenschaft gelten. TAUBERT kann nun aber zeigen, dass für die Gruppe der freien Softwareentwickler(innen) MERTONS Normen tatsächlich handlungsleitend sind. Auch wenn ich als der Rezensent dieses Buches dieser Schlussfolgerung beim ersten Querlesen von *Produktive Anarchie?* äußerst skeptisch gegenüberstand, so führte das genauere Studieren der Darstellung des empirischen Materials jedoch dazu, dass es mir nicht ausgeschlossen scheint, dass TAUBERT und damit MERTON für den Bereich der freien Softwareentwicklung Recht haben könnten. Das wissenschaftliche Ethos lebt also? Für TAUBERT stellt es sogar den wichtigsten Mechanismus dar, der bei Programmierprojekten zur Softwareentwicklung führt (S.105). [7]

Im *dritten* Teil der Arbeit (S.119-209) taucht TAUBERT tiefer in die Details offener Programmierprojekte ein. Er beschäftigt sich mit der Analyse von Mechanismen der Handlungskoordination in offenen Projekten. Er konzentriert sich hier auf das Fallbeispiel KMail. KMail ist ein E-Mail Client für eine besondere grafische Benutzeroberfläche. TAUBERT ist besonders an den Gründen für eine Beteiligung an der Entwicklung dieses E-Mail Clients interessiert. Es erscheint wenig überraschend, wenn er zeigt, dass die Beteiligung von Faktoren wie intrinsischem Interesse, der Möglichkeit einer Verfolgung von fernen Entwicklungszielen, der Befriedigung von eigenen Anwendungsbedürfnissen, der Anerkennung durch Anwender und Anwenderinnen sowie von Faktoren wie Reputation und wiederum von normativen Überzeugungen abhängt (S.128-131). [8]

TAUBERT macht nun einen Sprung und fragt nach dem Beginn eines solchen Projekts. Anhand eines gescheiterten Gründungsversuchs zeigt er, dass zu dieser "Stunde Null" freier Softwareentwicklungsprojekte paradoxe Anforderungen bewältigt werden müssen: Zum einen stellt sich die Aufgabe der Entwicklung eines konsistenten Programmentwurfs, zum anderen die der Integration von interessierten Programmierer(inne)n. Da dieses Interesse auf der Verfolgung von "eigenwilligen" Entwicklungszielen basiert, gelingt die Integration nur, wenn der Entwurf sich gegenüber weiteren Entwürfen als offen darstellt und das überraschende Auftreten von Fehlern geschickt absorbieren kann. Innerhalb von freien Softwareentwicklungsprojekten wird dieses Problem mittels der Strukturierungstechnik "Modularisierung" bewältigt. Geht die Entwicklung freier Softwareprojekte in den "Normalbetrieb" über, sind es zwei Mechanismen, die Einigungsdruck ausüben: Zum einen ist eine Ermüdung der Teilnehmenden festzustellen, zum anderen ein Übergang vom "Primat der Argumentation" zum "Primat der Implementation". Die (provisorische) Implementation einer Funktionalität durch die Protagonist(inn)en gilt jedoch nur dann als legitim, sofern die argumentativen Mittel vorher ausgeschöpft wurden. In Bezug auf die Einflussnahme der Anwender(innen) auf die im Projekt verfolgten Zielsetzungen ist zu bemerken, dass diese sich in einer schlechten strategischen Position befinden: Das für die Diskussion typische hohe Kompetenzniveau führt zu sozialen Ausschließungsprozessen von Personen, die dieses Kriterium nicht erfüllen. Die Versuche von Einflussnahmen der Anwender(innen) auf Zielsetzungen nehmen daher die Form von Wünschen an, die wenig Aussicht auf Erfolg haben. TAUBERT folgert daraus, dass trotz der anfänglichen Offenheit freier Entwicklungsprojekte, welche die Beteiligung von Anwender(inne)n fördert, diese Form der Entwicklung kein Beispiel für eine gelungene partizipative Technikentwicklung darstellt. [9]

Im letzten Teil (S.211-226) werden Innenansichten freier Softwareentwicklungsprojekte weiter beleuchtet und an die vorher im Buch bereits eingeführte Diskussion um "Realexperimente" angeknüpft. These ist hier, dass um die Stabilität und den Erfolg von freier Softwareentwicklung zu erklären der Rückkopplungsprozess zwischen Produktions- und Anwendungskontext entscheidend ist. Durch die Veröffentlichung von neuen Versionen gelangt das Programm in unterschiedliche Anwendungsfelder und wird dort erprobt. Diese Kontextualisierung stellt dabei nicht nur den Einsatz einer Technik zur Lösung von Anwendungsproblemen dar, sondern ist gleichermaßen Realexperiment zur Erzielung von Wissen über unvorhergesehene und unerwünschte Reaktionen des Programms. Dieses Wissen wird von den Anwender(inne)n erschlossen und in Form von Fehlerberichten in den Entwicklungskontext eingespeist. Es dient zur Konstruktion von dekontextualisierten Experimenten, mit denen Fehler lokalisiert und behoben werden. Mit diesen Innenansichten zeigt TAUBERT, dass die Gestalt der Technik von der sozialen Organisation des Entwicklungsprozesses selbst geprägt wurde. Dies gilt nicht nur für die Modularisierung des Programmquellcodes, die sich aus den praktischen Anforderungen nach der Integration von neuen, eigenwillig interessierten Personen speist, sondern auch für die typischen Kompromisse bei den Entscheidungen über Zielsetzungen sowie die Stabilität des Programms. [10]

Das Phänomen der Stabilität ist ein zentraler Punkt des Buches. Die in der einen oder anderen Form immer wieder gestellte Frage lautet: "Warum hält sich die freie Softwareentwicklung so statthaft?" Stabilität der Entwicklung *und* des Programms resultieren, so TAUBERT, "aus einer Rückkopplung von Anwendungs- und Entwicklungskontext, mit der Wissen über Programmierfehler erzeugt wird, das die Grundlage zur Beseitigung dieser Fehler bildet" (S.203). Was könnte damit genau gemeint sein? Er führt dies so aus:

"Er [der Rückkopplungsprozess] beginnt mit der Veröffentlichung von neuen Programmversionen, mit der die Software von einer Vielzahl unterschiedlicher Anwendungskontexte überführt und dort erprobt wird. Da sich die Kontexte in Bezug auf die sachtechnische Umgebung wie Hardware, Betriebssystem und andere Applikationsprogramme stark unterscheiden, die Nutzer mit der Anwendung des Programms verschiedene Zwecksetzungen verfolgen und unterschiedliche Arten von Daten mit dem Programm bearbeitet werden, handelt es sich hierbei um eine realexperimentelle Erprobung des Programms, die auch der Gewinnung von Wissen über Programmierfehler dient [...] Entwickler werden so zu Experimentatoren, sind aber, was die systematische Aufzeichnung des Wissens angeht, auf die Kooperation mit Anwendern angewiesen" (S.203). [11]

Diese Form der Kooperation, so TAUBERT, zeigt sich in den oben angesprochenen normativen Erwartungen, die die Entwickler(innen) an die Fehlerberichte der Anwender(innen) stellen. Die Stabilität im Entwicklungsprozess wird, so ist TAUBERT wohl zu verstehen, erreicht, in dem das praktische Wissen um Fehler mit theoretisch-systematischem Wissen so integriert wird, dass es in iterativen Lernprozessen des 1. Ausprobierens neuer Funktionen, 2. der Erprobung der Software und 3. schließlich der Fehlerfindung durch kontrollierte Rahmenbedingungen entwickelt wird. Nach der Fehlerfindung werden wieder neue Funktionen ausprobiert. Es ist hier interessant anzumerken, dass TAUBERT diese Erprobung von Software als Realexperiment bezeichnet, um darauf zu verweisen, dass es sich hier um eine neue und überlegene "experimentelle" Form der Wissensgenierung außerhalb des überschaubaren Bereichs der "zertifizierten Experten" (vgl. COLLINS & EVANS 2002) der Institution Wissenschaft handelt. [12]

TAUBERT versucht in diesem Kapitel das Konzept des Realexperiments so zu gebrauchen, um es nicht mit anderen Formen risikobehafteten Handelns, mit Strategien des Versuchs-und-Irrtums oder mit aneinander gereihten Handlungsketten zu verwechseln. Dieser Gedanke ist richtig. Insbesondere von "instrumentellem Handeln" möchte TAUBERT realexperimentelle Strategien unterscheiden wissen, um das Problem der Analyse "infiniter Folgenketten" zu lösen (S.188-199). Was er außerdem vorschlägt, ist eine Kategorisierung unterschiedlicher Grade der Kontrolle der Variablen (S.192). Damit ein Realexperiment ein Experiment ist, müsse der Experimentator bzw. die Experimentatorin, so TAUBERT, die unabhängige Variable in jedem Fall kontrollieren, weil es sich sonst nicht um ein Experiment handle. Ist dies nicht der Fall, handle es sich um eine Beobachtung (S.193). Wie genau nun TAUBERT in einer realexperimentellen Anordnung verschiedene Grade der Kontrolle unterscheiden

will, bleibt mit dem nicht weiter ausgeführten Vorschlag für "die systematische Hervorbringung eines Phänomens" (S.193) leider unklar. [13]

Um die Randbedingungen besser in den Griff zu bekommen, gehören zu TAUBERTs realexperimentellem System dann jedoch "sämtliche in die Konzeption des Experimentators fallenden Gegenstände und Phänomene, die vom Experimentierenden eingesetzt, kontrolliert oder hervorgebracht werden und die er seinem eingreifenden Handeln zurechnet" (S.196). Dies ist eine nicht ganz glückliche Rahmung für den Fall der Erprobung neuer Software. Denn es ist nicht recht zu verstehen, warum damit nun andere Konzeptionen von Realexperimenten, auf die sich TAUBERT bezieht, verbessert werden sollen. Wie verschiedene Autor(inn)en aufzeigen, gehört zu einem realexperimentellen Setting, um es von anderen Formen der Veränderung zu unterscheiden, an zentraler Stelle 1. ein Design, 2. klar identifizierbare Beobachter(innen), 3. beobachtbare Effekte, 4. ein Monitoring oder zumindest eine Form der Protokollführung, 5. eine ausgewiesene Akteurskonstellation und 6. die Rückfütterung der Ergebnisse in den nächsten Schritt eines Planungsprozesses (vgl. KROHN & WEYER 1989; KROHN 2003; HOFFMANN-RIEM 2003; allgemein zu diesem Thema auch KOHLER 2002 oder LATOUR 2006). Es lassen sich in der Literatur weitere Bedingungen finden, die Realexperimente von anderen Formen des Handelns unterscheiden. Wichtiger ist, dass mit TAUBERTs Verweis auf die auf den Experimentator/die Experimentatorin zurechenbaren Phänomene ein entscheidender Faktor ausgebremst wird: Sind es nicht gerade die überraschenden und abweichenden Ereignisse, auf die Experimentator(inn)en warten? Es sind doch meist die nicht klar zurechenbaren Phänomene, die nicht antizipiert waren und oft von außerhalb des Versuchssystems kommen, die im Realexperiment – und gelegentlich auch im Laborexperiment – für neue und innovative Anstöße sorgen. Zudem startet TAUBERT mit dieser Definition keine Neuerung für das Verständnis von Realexperimenten, sondern versucht mit seiner Variante wieder möglichst nahe an den Idealtypus eines Laborexperimentes heranzureichen – ein unmögliches Unterfangen. Damit betrachtet er das Realexperiment aber implizit als minderwertige Form des "echten" Experimentes. Dies ist strategisch unglücklich, denn TAUBERT will ja, wie seine Ausführungen deutlich zeigen, gerade auf den andersartigen und für die Entwicklung der Software überlegenen *Eigencharakter* dieser Form des Experimentierens hinweisen. Ob er daher mit dieser wieder an das Laborexperiment angelehnten Variante des Realexperimentes die kritischen Kommentator(inn)en früherer Konzepte des Realexperimentes (z.B. WEHLING 2006; WESTERMAYER 2006; WYNNE 2002) eher überzeugen kann als dies bis jetzt gelungen ist, scheint eher unwahrscheinlich. [14]

3. Ausblick: Wissenschaft und Softwareentwicklung im 21. Jahrhundert

Nur einige wenige Aspekte des materialreichen Buches konnten hier beleuchtet werden. Alles in allem ist TAUBERTs Buch eine gut strukturierte und mit hilfreichen Leseanleitungen gespickte Arbeit, die zentrale Hinweise auf die Erklärung der Organisation von Technikentwicklungsprojekten liefert. [15]

Wenn nun tatsächlich, wie verschiedene Autoren und Autorinnen propagieren, heute die Naturwissenschaften die Ausbreitung der Forschung aus dem Labor in die Gesellschaft selbst zugelassen haben und weiter vorantreiben (vgl. KROHN & WEYER 1989; LATOUR 2004; LEVIDOW & CARR 2007), dann hat auch die Institution Wissenschaft ihre exklusive Beurteilung dessen, was als Wissenschaft gilt, verspielt. Mit dem Ergebnis, dass die Koordination der freien Softwareentwicklung als Wissensproduktion außerhalb der akademischen Wissenschaft auf MERTONs Wissenschaftsethos fußt, stellt TAUBERT eine These auf, die sicherlich zu Widerspruch einlädt. Wenn dann aber, wie TAUBERT glaubhaft zeigt, neue Formen der Innovation und Wissensproduktion außerhalb der institutionalisierten Wissenschaft die ethisch "bessere" Wissenschaft hervorbringen können, dann könnte dies durchaus als Hinweis darauf gelten, dass die Gesellschaft des 21. Jahrhunderts vor einem neuen Paradigma der Wissensproduktion steht (vgl. NOWOTNY, SCOTT & GIBBONS 2004). [16]

Nach TAUBERT findet sich MERTONs Ethos nicht hinter Laborwänden versteckt, sondern außerhalb dieser. TAUBERTs Fallstudie kann hier als Indikator einer neuen Form der Wissensproduktion verstanden werden, in der das wissenschaftliche Zielpublikum nicht nur in der weiteren Gesellschaft besteht, sondern ebenso eine allgemeine soziale Relevanz, gesellschaftliche Verantwortung sowie eine egalitär anmutende Organisation für die Wissensproduktion handlungsleitend sind, obwohl Unsicherheiten als elementare Bestandteile der Wissensproduktion aufgefasst werden. Realexperimente, so wie TAUBERT sie entworfen hat, sind eine eigene Form der Wissens- und Technikentwicklung, die unter besonderen Komplexitätsbedingungen stattfinden. Sie setzen sich umso stärker durch und werden robuster, je demokratischer ihre Prozesse sind und je mehr nicht nur zertifizierte Experten und Expertinnen Beteiligte am experimentellen Design sein dürfen, sondern quasi "jedermann". Auftretende Fehler können dann über ein experimentelles Rahmenwerk immer wieder behoben werden, um mit diesen Anpassungen das Gesamtprogramm Schritt für Schritt zu verbessern. TAUBERT ist mit seinem Buch und der Herausarbeitung dieses Prozesses ein beachtlicher Wurf gelungen, der verspricht, nicht nur in der Wissenschaftssoziologie Wellen zu schlagen. [17]

Literatur

- Asdonk, Jupp; Bredeweg, Udo & Kowol, Uli (1991). Innovation als rekursiver Prozess: Zur Theorie und Empirie der Technikgenese am Beispiel der Produktionstechnik. *Zeitschrift für Soziologie*, 20(4), 290-304.
- Bakardjewa, Maria (2005). *Internet society: The Internet in everyday life*. London: Sage.
- Barnes, Barry & Robert G.A. Dolby (1972 [1970]). Das wissenschaftliche Ethos: Ein abweichender Standpunkt. In Peter Weingart (Hrsg.), *Wissenschaftssoziologie Band I. Wissenschaftliche Entwicklung als sozialer Prozess* (S. 263-286). Frankfurt/M.: Fischer Athenäum.
- Collins Harry M. & Evans, Robert J. (2002). The third wave of science studies: Studies of expertise and experience. *Social Studies of Science*, 32(2), 235-296.
- Franzen, Axel (2000). Does the Internet make us lonely? *European Sociological Review*, 16(4), 427-438.
- Garfinkel, Harold (1967). *Studies in ethnomethodology*. Englewood Cliffs, NJ: Prentice Hall.
- Herbold, Ralf; Kämper, Eckard; Krohn, Wolfgang; Timmermeister, Markus & Vorwerk, Volker (2002). *Entsorgungsnetze. Kommunale Lösungen im Spannungsfeld von Technik, Regulation und Öffentlichkeit*. Baden-Baden: Nomos.
- Hoffmann-Riem, Holger (2003). *Die Sanierung des Sempachersees: Eine Fallstudie über ökologische Lernprozesse*. München: Ökom.
- Katz, James E. & Rice, Ronald E. (2003). *Social consequences of Internet use: Access, involvement, and interaction*. Cambridge, MA: MIT Press.
- Kohler, Robert E. (2002). *Landscapes and labscales: Exploring the lab-field border in biology*. Chicago: University of Chicago Press.
- Kowol, Uli (1998). *Innovationsnetzwerke: Technikentwicklung zwischen Nutzungsvisionen und Verwendungspraxis*. Wiesbaden: Deutscher Universitätsverlag.
- Krohn, Wolfgang (2003). Zukunftsgestaltung als Experimente von und mit uns selbst. Wer kann das verantworten? In Kirsten Mensch & Jan C. Schmidt (Hrsg.), *Technik und Demokratie* (S.157-177). Opladen: Leske + Budrich.
- Krohn, Wolfgang & Weyer, Johannes (1989). Die Gesellschaft als Labor: Die Erzeugung sozialer Risiken durch experimentelle Forschung. *Soziale Welt*, 40(3), 349-373.
- Latour, Bruno (2004). Ein Experiment von und mit uns allen. In Gerhard Gamm, Andreas Hetzel & Markus Lilienthal (Hrsg.), *Die Gesellschaft im 21. Jahrhundert* (S.185-195). Frankfurt/M.: Campus.
- Latour, Bruno (2006 [1983]). Gebt mir ein Laboratorium und ich werde die Welt aus den Angeln heben. In Andréa Belliger & David J. Krieger (Hrsg.), *ANThology: Ein einführendes Handbuch zur Akteur-Netzwerk-Theorie* (S.103-134). Bielefeld: Transcript.
- Levidow, Les & Carr, Susan (2007/im Druck). GM crops on trial: Technological development as a real-world experiment. *Futures*, 39(4).
- Merton, Robert K. (1942). Science and technology in a democratic order. *Journal of Legal and Political Sociology*, 1(1/2), 115-126.
- Nowotny, Helga; Scott, Peter & Gibbons, Michael (2004 [2001]). *Wissenschaft neu denken: Wissen und Öffentlichkeit in einem Zeitalter der Ungewissheit*. Weilerswist: Velbrück.
- Stehr, Nico (1978). The ethos of science revisited: Social and cognitive norms. In Jerry Gaston (Hrsg.), *Sociology of Science* (S.172-196). San Francisco: Jossey-Bass.
- Wehling, Peter (2006). *Im Schatten des Wissens? Perspektiven der Soziologie des Nichtwissens*. Konstanz: UVK.
- Westermayer, Till (2006). Rezension zu: Matthias Groß, Holger Hoffmann-Riem & Wolfgang Krohn (2005). Realexperimente. Ökologische Gestaltungsprozesse in der Wissensgesellschaft [28 Absätze]. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research* [Online Journal], 7(4), Art. 13. Verfügbar über: <http://www.qualitative-research.net/fqs-texte/4-06/06-4-13-d.htm> [Zugriff: 10. September 2006].
- Wynne, Brian (2002). Some observations on the notion of "experiment" which informs the idea of social experiments for robust design. Papier vorgestellt auf dem Workshop *Socially Robust Ecological Design*, Bielefeld, Oktober.

Zum Autor

Matthias GROSS (Jg. 1969) arbeitet als Soziologe am *Helmholtz-Zentrum für Umweltforschung*, UFZ in Leipzig. Er studierte Soziologie in Bielefeld und für ein Jahr als "graduate student" an der amerikanischen Westküste (Arcata, Kalifornien). Im Jahre 2000 war er DAAD-Doctoral Fellow an der University of Wisconsin, Madison (USA) und promovierte 2001 in Soziologie an der Universität Bielefeld. Zu seinen Arbeitsschwerpunkten gehören die Umweltsoziologie, Geschichte der Soziologie, Wissenschafts- und Technikforschung, sowie die Gestaltung und Sanierung postindustrieller und kontaminierter Landschaften. Er lehrte an den Universitäten Karlsruhe (TU), Leipzig und Bielefeld, sowie der Loyola University Chicago (USA). Zu seinen Büchern gehören *Inventing Nature: Ecological Restoration by Public Experiments* (2003), *Realexperimente: Ökologische Gestaltungsprozesse in der Wissensgesellschaft* (2005, zus. mit Wolfgang KROHN und Holger HOFFMANN-RIEM), sowie *Natur* (2006). Er ist Gründer und Mitherausgeber der Zeitschrift *Nature and Culture*.

Kontakt:

Dr. Matthias Groß

Department Stadt- und Umweltsoziologie
Helmholtz-Zentrum für Umweltforschung, UFZ
Permoserstr. 15
D-04318 Leipzig

Tel.: 0341-235 3315

Fax: 0341-235 2825

E-Mail: matthias.gross@ufz.de

URL: <http://www.ufz.de/index.php?en=5479>

Zitation

Gross, Matthias (2006). Rezension: Niels C. Taubert (2006). Produktive Anarchie? Netzwerke freier Softwareentwicklung [17 Absätze]. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 8(1), Art. 10, <http://nbn-resolving.de/urn:nbn:de:0114-fqs0701109>.