

## Einleitung

---

Erfolgsgeschichten gab es im IT- und Softwarebereich in den vergangenen Jahren viele – wenn auch mit durchaus unterschiedlichem Ausgang. Kinder des Internet-Booms traten in großer Zahl und mit viel Getöse auf, zogen nicht nur Risikokapital, sondern in den 1990er Jahren auch das Geld von Kleinanlegern an, um dann nach mehr oder minder lang anhaltendem Höhenflug der Aktienkurse kurz von dem Jahrtausendwechsel herbe Abstürze hinzulegen. Aus diesem Boom erwachsende Unternehmen waren dann doch selten, häufig endete der Erfolg in Karerstimung: Dem ‚rise‘ folgte – wie so häufig – ein noch prompterer ‚fall‘. Verlässlicher Lieferant guter Nachrichten im Bereich der Informations- und Kommunikationstechnologien bildete während der gesamten Zeit freie Software: Von der Verbreitung und Popularisierung des Entwicklungsmodells Anfang/Mitte der 1990er Jahre bis zum heutigen Zeitpunkt läßt sich eine Geschichte des Erfolgs schreiben, die derartige Einbrüche nicht kennt. Die Zahl freier Programme und deren Verbreitung steigt kontinuierlich.

Zum Beispiel Linux: Als prominentester Vertreter dieses Softwaretypus ist das Betriebssystem mittlerweile zu einer etablierten Erscheinung auf Rechnerarbeitsplätzen geworden. Gestartet als System von Programmierern für Programmierer, hat es stark an Benutzerfreundlichkeit gewonnen und kann auch ohne die Kenntnis kryptischer Befehle von jedermann bedient werden. So gereift, hat es seinen Weg vom Exosystem für Computerfreaks in die Unternehmen,<sup>1</sup> die Bundestagsver-

---

1 Siehe hierzu beispielsweise die Liste der Referenzkunden des Linux-Distributors Suse unter [http://www.suse.de/de/company/customer\\_references/index.html](http://www.suse.de/de/company/customer_references/index.html). Alle in diesem Buch angegebenen Internetseiten wurden, sofern nicht anders angegeben, zuletzt im Oktober 2004 gesehen.

waltung,<sup>2</sup> die Stadtverwaltungen,<sup>3</sup> öffentliche Einrichtungen,<sup>4</sup> das Militär<sup>5</sup> und auch den heimischen PC gefunden.

Zum Beispiel Apache: Dieser Webserver bietet von seiner Verbreitung her ein noch größeres Erfolgsbeispiel. Ebenfalls als freies Softwareentwicklungsprojekt Anfang 1995 in der Version 1.0 veröffentlicht, gilt er seit spätestens 1996 als der am weitesten verbreitete Webserver, vom dem nach jüngsten Umfragen mehr als 62 Prozent der Websites im WWW gehostet werden.<sup>6</sup> Die entsprechenden Konkurrenzprogramme der Unternehmen Microsoft, Zeus und Sun folgen weit abgeschlagen.

Zum Beispiel Firefox: Jüngstes Beispiel bildet dieser Webbrowser, ein Ableger von Mozilla. Mitte September 2004 wurde die Version 1.0 der Öffentlichkeit vorgestellt und zum Download freigeschaltet. 300.000 Downloads in den ersten 24 Stunden künden von einem riesigen Interesse an dem Programm und lassen die Entwickler darauf hoffen, innerhalb von einem Monat von Null auf zehn Millionen Nutzern zu kommen.<sup>7</sup>

Überwältigend ist neben der schnellen Verbreitung von freier Software auch der Ruf, der diesem Programmtypus vorausieht: Kurz nachdem das Phänomen öffentlich wahrgenommen wurde, war die Berichterstattung euphorisch und ist heute immer noch positiv-wohlwollend – kritische Anmerkungen und Kommentare sind nach wie vor spärlich. Freie Software gilt als die freundliche und begrüßenswerte Alternative zu dem sonstigen Einerlei, das durch einen weitgehend negativ etikettierten amerikanischen Softwarekonzern geschaffen wurde. In diesem Spiel David gegen Goliath sind die Sympathien klar verteilt: David gilt hier als die Partei, für die das Herz schlägt, auch wenn vorerst noch die Software von Goliath genutzt wird.

Diese positiven Zuschreibungen setzen sich auf technischer Ebene fort: Freie Software gilt als stabiler, verschont den leidgeplagten Nutzer

---

2 Vgl. hierzu den Bericht über die Entscheidung des Ältestenrats in der Neuen Zürcher Zeitung vom 14.03.04. Als Online-Dokument: <http://www.nzz.ch/2002/03/08/em/page-article80M1P.html>.

3 So zum Beispiel die Stadtverwaltung München. Siehe den Spiegel Online Bericht vom 02.06.04. Als Online-Dokument: <http://www.spiegel.de/spiegel/0,1518,251077,00.html>.

4 Beispiel hierfür bilden die Terminals der Universitätsbibliothek Bielefeld.

5 Der vom Pentagon bei IBM in Auftrag gegebene Supercomputer soll aus 1186 IBM-Rechnern zusammenschaltet werden, die US Navy mit aktuellen Wettervorhersagen versorgen und ein Linux-Betriebssystem besitzen. Siehe den Bericht vom 04.08.04 in Spiegel Online unter <http://www.spiegel.de/netzwelt/netzkultur/0,1518,311542,00.html>.

6 Vgl. die Netcraft-Umfragen vom Dezember 2002. Als Online-Dokument: <http://www.netcraft.com/Survey/index-200212.html>.

7 Vgl. Spiegel Online vom 16.09.04. Als Online-Dokument: <http://www.spiegel.de/netzwelt/technologie0,1518,druck-318189,00.html>.

von Systemabstürzen und verhält sich damit ressourcenschonend – zumindest was die Nerven der Anwender angeht. Freie Software gilt als weniger betroffen von Viren, Würmern, Trojanern und Spybots, also all jenen Begleiterscheinungen der vernetzten Gesellschaft, die dem Anwender das Fürchten lehren und ihn veranlassen, regelmäßig lästige Sicherheitsupdates durchzuführen, auch wenn er weiß, daß ihm diese eine nur äußerst kurzzeitige Atempause verschaffen – bis eben die nächste kritische Sicherheitslücke entdeckt wird, deren Größe mittlerweile die eines Scheunentors annehmen kann.<sup>8</sup> Selten war der Ruf eines bestimmten Typus von Technik so gut, wie es bei freier Software der Fall ist.

Worauf gründet sich nun dieser Erfolg, der sich in Ruf wie Verbreitung zeigt? Liegt es daran, daß freie Software in aller Regel kostenlos erhältlich ist, der Anwender sie also nutzen kann, ohne dafür zu bezahlen – frei nach dem Motto: ‚Einem geschenkten Gaul schaut man nicht ins Maul‘? Liegt es an der tatsächlichen Überlegenheit der Qualität des Programms, die aus dem Modell resultiert, mit dem es entwickelt wird und das dem elektronischen Wunderland direkt entsprungen zu sein scheint? Oftmals als chaotisch oder als anarchisch beschrieben, kann sich jedermann an der Softwareentwicklung beteiligen, das Programm nach seinen Wünschen modifizieren, was in der Summe dann wie von Zauberhand geleitet oftmals zu einem sehr brauchbaren Ergebnis führt.

Einen Teil des Erfolgs erklären wir in dieser Arbeit, indem wir der Frage nachgehen, wie freie Software innerhalb von offenen Programmierprojekten entwickelt wird. Das zumindest im Fall von erfolgreichen Projekten gelingende Zusammenwirken in Richtung eines Projektziels findet dabei unter Rahmenbedingungen statt, die auf den ersten Blick für das Entstehen von Handlungskoordination wenig günstig erscheinen: Freie Softwareentwicklungsprojekte sind öffentlich – jeder kann also mitmachen, aber genauso schnell wieder abspringen. Die Entwickler sind über den Globus verteilt und koordinieren ihre Arbeit ausschließlich über das Internet. Es findet sich bei näherer Betrachtung keine Organisation, welche die Koordinationsleistung in freien Softwareentwicklungsprojekten erbringt und die notwendigen Ressourcen für die Entwicklung bereitstellt. Die Existenz freier Software wird bei näherer Betrachtung zunehmend erklärungsbedürftig.

Die Studie gliedert sich wie folgt: In einem ersten Teil legen wir die Grundlagen für die Untersuchung. Am Anfang steht hier die Klärung der Begriffe ‚freie Software‘ und ‚freie Softwareprojekte‘, mit denen wir den hier interessierenden Gegenstand bestimmen. Hieran anschließend

---

8 Vgl. Spiegel Online vom 08.07.04. Als Online-Dokument: <http://www.spiegel.de/netzwelt/technologie/0,1518,druck-307447,00.html>.

werden wir im zweiten Kapitel die Fragestellung entwickeln. Abgeschlossen wird dieser Grundlagenteil mit der Darstellung des theoretischen Rahmens der Untersuchung, dem Konzept der Innovationsnetzwerke, mit dem Kooperationsprozesse zwischen heterogenen Akteuren im Bereich der Technikentwicklung analysiert werden.

Im Fokus des zweiten Teils steht dann die Frage nach den Ursachen für die Entstehung des Netzwerks freier Softwareentwicklung, das sich in die für das Feld typischen freien Softwareentwicklungsprojekte gliedert. Wie wir sehen werden, ist diese Struktur zwei Innovationen geschuldet: zum einen der vertraglichen Innovation freier Softwarelizenzen, zum anderen der organisatorischen Innovation der offenen Programmierprojekte. Im ersten Kapitel des zweiten Teils werden wir diese spezielle Form der Lizenz diskutieren und zeigen, daß diese sich reflexiv auf das Urheberrecht bezieht: Die dort monopolisierten Nutzungsrechte am Programm zugunsten des Werkschöpfers werden genutzt, um sie zu generalisieren: *All rights reversed*. Daran anschließend analysieren wir die zweite Innovation der offenen Programmierprojekte. Bezugspunkt dieser Innovation bildet das Software Engineering, eine Teildisziplin der Informatik, die ein Set von normativen Vorgehensmodellen bereithält, in denen sich die leitenden Vorstellungen über die Organisation von Softwareentwicklungsprozessen finden. Mit der Analyse der Diskussionen, die zum ersten offenen Programmierprojekt geführt haben, werden wir zeigen, daß sich die Innovation weitgehend negativ auf das Software Engineering bezieht und an die Stelle einer hierarchischen Projektsteuerung die Idee einer selbstorganisierten Projektabwicklung setzt.

Nachdem wir die Ursachen für das Entstehen dieser sozialen Struktur analysiert haben, werden wir uns im dritten Teil – Innenansichten offener Programmierprojekte – der Analyse der sozialen Mechanismen zuwenden, die für eine Handlungskoordination sorgen. Zunächst werden wir das Fallbeispiel KMail vorstellen, um daran anschließend in Form eines Exkurses die Gründe der Akteure für eine Beteiligung an der freien Softwareentwicklung zu analysieren. Im dritten Kapitel dieses Teils steht dann der Projektbeginn und die dabei von dem oder den Gründer(n) zu bewältigenden Aufgaben sowie eine im Bereich der freien Softwareentwicklung typischerweise gefundene Lösung im Mittelpunkt des Interesses. Nach der Analyse dieser Schlüsselsituation ‚Projektbeginn‘ werden wir uns für den ‚Normalbetrieb‘ von offenen Programmierprojekten interessieren, indem wir der Frage nachgehen, wie über Zielsetzungen entschieden wird. Hier werden wir den grundlegenden Mechanismus der Entscheidungsfindung sowie einige Konsequenzen für die Gestaltung von Software herausarbeiten und die Rolle der Benutzer diskutieren. Es wird sich zeigen, daß – zumindest dem Fallbeispiel zu-

folge – diese Programmierprojekte trotz ihrer Offenheit über Abschließungsmechanismen verfügen, die den Einfluß von Anwendern auf die Definition von Zielsetzungen beschränken. Freie Softwareentwicklung bildet daher nicht generell ein Beispiel für gelungene partizipative Technikentwicklung. Im fünften, die Innenansichten abschließendem Kapitel werden wir die Frage stellen, wie es zu dem häufig mit freier Software in Zusammenhang gebrachten Merkmal Stabilität kommt. Unsere Antwort hierauf lautet: Die Stabilität ist Resultat der Rückkopplung zwischen Herstellungs- und Anwendungskontext und der vielfachen experimentellen Erprobung der Programme in heterogenen Anwendungskontexten. Die Gestalt von Software – hier die Stabilität – resultiert also aus der organisatorischen Struktur des Softwareentwicklungsprozesses.

Die Arbeit schließt mit einem vierten Teil, in dem wir den theoretischen Ertrag zusammenfassen. Hier werden wir zeigen, daß das Konzept der Innovationsnetzwerke zwar geeignet ist, einige wesentliche Merkmale von offenen Programmierprojekten zu beschreiben, allerdings an anderen zentralen Stellen unserer Untersuchung an ihre Grenzen stößt. Wir werden daher statt einer weitgehend implizit bleibenden Fundierung des ursprünglichen Konzepts auf einer ‚Rational-Choice Theorie‘ eine institutionalistische Lesart, basierend auf einem normativen Handlungsmodell, entwickeln.