

Software Libre: die konstruktive Revolution

Wir schreiben das Jahr 2010, und die Freie-Software-Bewegung wird universell als Erfolg gefeiert. Obwohl die Mehrheit der Computernutzer noch immer unfreie Software anwendet, hat sich freie Software von einer technischen, für das breite Publikum uninteressanten Kuriosität zu einem viel beachteten Phänomen entwickelt. Es gibt kaum eine Firma auf dem Computermarkt, die ihre Produkte und Dienstleistungen nicht mindestens zu einem Teil auf freie Software aufbaut. Und obwohl die Nutzer unfreier Software noch sehr zahlreich sind, wächst ihr Unmut, was wiederum die Firmen dazu bringt, immer restriktivere und unbehaglichere «lock-in»-Mechanismen zu verwenden.¹ Wenn Microsoft, der große Software-Monopolist, seinen jährlichen Bericht für die amerikanische Börsenaufsichtbehörde veröffentlicht, nennt er freie Software als die einzige Bedrohung für seine Übermacht.

Im Rückblick scheint diese Entwicklung eigentlich unabwendbar: Unabhängig von ihren technischen Merkmalen sind die sozialen, politischen und ökonomischen Stärken der freien Software dermaßen übermächtig, dass man sich kaum etwas vorstellen kann, was ihr Widerstand leisten könnte. So sieht es heute aus. Vor 25 Jahren waren die Aussichten der Freie-Software-Bewegung aber lange nicht so rosig.

Zum einen bestand die Bewegung damals aus einem einzigen Mitglied: Richard Stallman, dem ersten, der die sozialen Gefahren der unfreien Software erkannt hatte. Zum anderen waren die Umstände sehr nachteilig: Computer waren damals noch keine Alltagsgegenstände, und dementsprechend schwierig war es, die politische und soziale Dimension des Problems zu begreifen. Die Gegner der freien Software waren von den Leuten bewunderte Hightech-Großkonzerne, die häufig sogar als gesellschaftliche Wohltäter angesehen wurden. Alles hätte auf so vielfache Weise schief laufen können, dass es sich lohnt, die Faktoren dieser Bewegung zu untersuchen, die einen solchen Erfolg ermöglicht haben.

1 Unter «lock-in» versteht man jene Mechanismen, die Software-Hersteller verwenden, um zu vermeiden, dass ihre Kunden auf Konkurrenzprodukte umsteigen. Eine verbreitete «lock-in»-Strategie besteht darin, Computerdateien in Formate zu kodieren, die nur dem Hersteller bekannt sind. Wenn der Benutzer eines solchen Programms das Programm wechseln will, verliert er damit den Zugang auf seine eigenen Daten.

Es gilt als sicher, dass eine Strategie der direkten Opposition wenig Chancen gehabt hätte: Bis heute wird die Idee, unfreie Software zu meiden, auch von Nutzern freier Software skeptisch betrachtet. Selbst die Leute, die Computernetze tagtäglich benutzen, sind schwer davon zu überzeugen, auf die neueste technische Spielerei zu verzichten, auch wenn diese ganz offensichtliche negative Folgen aufweist, wie z.B. der Verlust der Zugangskontrolle auf die eigenen Daten. Eine Freie-Software-Bewegung, die sich darauf beschränkt hätte, die Nutzer über den Schaden aufzuklären, den unfreie Software anrichtet, hätte wenig Resonanz gefunden. Ebenso wenig wie eine, die schlicht verlangt hätte, die Praktiken der unfreien Software gesetzlich zu verbieten.

Vielleicht die interessanteste Eigenschaft der Bewegung ist die Strategie der «Opposition durch Konstruktion». Für viele Bewegungen wird eine bessere Welt erst möglich, wenn die alten Strukturen zerstört werden. Die Freie-Software-Bewegung erkannte früh, dass dies auf ihrem Spezialgebiet nicht unbedingt der Fall sein muss: die neue Konstruktion kann der Zerstörung vorausgehen.

«Als Schule wollten wir das von Anfang an nicht akzeptieren. Ohne das Konzept der freien Software zu kennen, begaben wir uns auf die Suche nach Möglichkeiten, wie wir aus diesem scheinbar unauflösbaren Konflikt herauskommen konnten: «Monopol – öffentliche Schule – angekettet sein – Illegalität». Dann kamen wir auf die freie Software, die zudem noch Stabilität, Sicherheit und kollektiv erarbeitetes Wissen, Ideale sowie ein hohes technisches Niveau mitbrachte, also sehr viel mehr, als wir ursprünglich erwartet hatten. Der ganze Kontext von Gemeinschaft, Zusammenarbeit, Solidarität, Forschung und der Gestaltung eines konstruktiven gemeinsamen Prozesses bot uns als öffentlicher Schule einen perfekten Rahmen. Wir haben umgehend versucht, diese Philosophie in unsere Institution zu integrieren. Und jetzt stören uns all die Verbote der Unternehmen nicht mehr, wir verwenden freie Software, wir teilen sie, tragen etwas zu ihr bei und verändern sie! Jeder gibt etwas von seiner Arbeitsleistung, von seinem Wissen und seinen Erkenntnissen. Er teilt auch seine Zweifel und Probleme mit, und alle gemeinsam gestalten wir die Ausbildung. In unserer Schule ist das Kopieren gern gesehen!»

Javier Castrillo

Der Startpunkt der Freien-Software-Bewegung war die Gründung des GNU-Projekts im Jahr 1984, das eine vollständige, freie Software-Umgebung erstellen wollte, um zu zeigen, dass Software auch auf diese Weise produziert und geliefert werden kann. Natürlich schien das Ausmaß des Unterfangens damals ein unüberwindbares Hindernis zu sein. Solch eine enorme Aufgabe kann man vermutlich nur beginnen eingedenk des Sprichworts: «Auch die weiteste Reise beginnt mit einem ersten Schritt.»

Eine Gefahr der konstruktiven Strategie besteht darin, dass unter Umständen auch die Gegner der Bewegung davon profitieren. Im Fall Software geschah dies sehr früh: Da die Software frei war, durften Vertreiber unfreier Software sie in ihre

Produkte einarbeiten und an ihre Kunden unter unfreien Bedingungen weitergeben. Das Ergebnis war ein ökonomischer Vorteil für den Vertreiber, während der Nutzer ein freies Programm bekam, das für ihn nicht mehr frei war.

Stallman fand eine bemerkenswerte Lösung für dieses Problem: Er entdeckte, dass dasselbe Urheberrecht, das gewöhnlich dazu verwendet wurde, das Kopieren von Software zu verbieten, auch auf kreative Weise angewendet werden konnte – und zwar um die Möglichkeiten anderer einzuschränken, die Software unter restriktiven Bedingungen weiterzugeben. Diese Idee, die heute als «Copyleft» bekannt ist, erwies sich aus mehreren Gründen als sehr effektiv und vorteilhaft: Der Wille der Programmierer bekam damit erstmals eine legale Grundlage. Durch den Einsatz eines gängigen Instruments des Urheberrechts, der Lizenz, konnten Programmierer den Wunsch, ihre Software solle für alle frei bleiben, rechtswirksam werden lassen. Da «Copyleft» im Urheberrechtsgesetz verwurzelt war, wagten es seine Gegner nicht, es vor Gericht herauszufordern, denn sollten sie Erfolg haben, würden sie vermutlich dem Urheberrecht Schaden zufügen. In der Tat: Obwohl Vertreter unfreier Software gerne Zweifel an der juristischen Wirksamkeit der GPL äußern, dauerte es mehr als zwanzig Jahre, bis einer von ihnen sie tatsächlich vor Gericht verklagte – erfolglos!

Sowohl «Copyleft» wie auch die Opposition durch Konstruktion stellten enorme Fortschritte dar, die es dem GNU-Projekt ermöglichten, eine große Anzahl freier Programme unter die Leute zu bringen, darunter auch Software, mit der Programme dezentral entwickelt werden können. Das heißt: Software-Entwicklung durch die unkoordinierte Zusammenarbeit zahlreicher Personen, die außer der Teilnahme am Projekt keine Verbindung zueinander haben, ganz unabhängig von ihren Standorten und ihrer Firmen- bzw. Organisationszugehörigkeit. Als sich das Internet Anfang der 90er-Jahre verbreitete, nahmen sowohl die Anzahl der Personen, die an freien Softwareprojekten arbeiteten, wie auch die Menge der Projekte mit einer Geschwindigkeit zu, die mit dem Wachstum des Netzes selbst vergleichbar war.

«Freie Software bedeutet zwangsläufig, auch die eigenen Methoden zu ändern, das Verhältnis zwischen Lehrern und Schülern und die Unterrichtsaktivitäten selbst, es geht um die wunderbare Erfahrung des Lernens um seiner selbst willen und nicht wegen irgendeines Punktesystems. Freie Software verändert in einer Lerngemeinschaft viel. Veränderungen, mit denen sich nicht alle Lehrer auseinandersetzen wollen. Aus vielerlei Gründen. Sie haben hauptsächlich damit zu tun, dass man sich nicht so engagiert und fremde Gewässer scheut. Unser pädagogischer Ansatz beruht auf der Kohärenz zwischen der Philosophie der freien Software und einem institutionellen Projekt, das auf Integration und Partizipation setzt und sich auf Werte wie Teilhabe, Solidarität, Teilen, Ethik, menschliche Entwicklung, Entfaltung der eigenen Fähigkeiten, Innovation, die Suche nach kreativen Problemlösungen, ständige Weiterbildung, politische Beteiligung und

soziales Engagement bezieht. Freie Software und Volksbildung mit sozialem Anspruch sind wie füreinander gemacht. Man kann nicht das Eine denken ohne das Andere im Blick zu haben, sie gehören zusammen, sind ein unteilbares Ganzes.»
Javier Castrillo

Dass die Freie-Software-Bewegung das Wachstum des Netzes so für sich nutzen konnte, lag an ihrer radikalen Dezentralisierung. Es gibt in der Bewegung überhaupt keine zentralisierten Strukturen zur Koordination oder Kontrolle. Um der Gemeinschaft beizutreten, braucht man nichts anderes zu tun, als freie Software zu installieren und anzuwenden. Um ein freies Projekt zu starten, braucht man es nur zu veröffentlichen. Jede Person, die freie Software verbreitet, ist ein Aktivist. Keine Erlaubnis ist notwendig, man braucht nicht einmal die Prinzipien der freien Software zu akzeptieren: Wichtige Beiträge zur freien Software, wie z.B. der Linux-Kernel, wurden von Leuten geschrieben, die sich nicht zur freien Software bekennen.

Radikale Dezentralisierung ist zwar wirksam, jedoch nicht unbedingt effizient: die Freie-Software-Gemeinschaft ist berüchtigt für ihre Heterogenität und ihre unendlichen internen Konfrontationen, die oft über technische, politische, philosophische und sogar persönliche Differenzen entflammen. Es fällt schwer zu argumentieren, dass es z.B. effizient ist, das GNU-System mit mehr als fünf miteinander konkurrierenden grafischen Benutzerschnittstellen auszustatten. Diese Aufwandsvervielfachung ist noch dazu die Regel und nicht die Ausnahme: Ob es um Kernels, Web-Browser bzw. Server, E-Mail-Programme, Datenbanken oder anderes geht, freie Software hat üblicherweise mehr als ein einziges Projekt anzubieten, das sich darum kümmert.

Auch die internen Querelen und die häufigen Ego-Trips sind ineffizient. Aber sie sind der Preis, den man zahlen muss, um die Vorteile der Dezentralisierung zu genießen. Die Eingangsschwelle zur Gemeinschaft ist niedrig. In einer vernetzten Welt bedeutet das, dass viele Leute sie überwinden können, um «etwas» beizutragen. Dieser Beitrag wird automatisch einem nahezu darwinistischen Prozess unterzogen: Ist er gut, wird er wahrscheinlich von vielen Leuten adoptiert und weiterentwickelt, wenn nicht, dann wird er stagnieren und in Vergessenheit geraten. Zahlreiche kleine Beiträge, zusammen mit dem Copyleft-Konzept gleichen das Effizienzproblem aus: Wir brauchen gar nicht effizient zu sein, wenn wir nur zu viel Aufwand fähig sind und alle von der Arbeit aller profitieren.

Sogar die internen Auseinandersetzungen und die ständigen Streitigkeiten wirken sich für die freie Software positiv aus: Da es weder eine klare und allgemein anerkannte Führung noch eine strenge Koordinations- oder Vertretungslinie gibt, wird ein wirksamer institutioneller Angriff sehr schwierig. Es gibt keine Person, Organisation, kein Projekt, das für die Bewegung unabdingbar wäre. Es gibt nichts, was pleite machen, ein Konzern kaufen oder verklagen könnte, um die Kontrolle über freie Software zu gewinnen.

Das Erfolgsrezept der freien Software ist schwer auf andere Fachgebiete zu übertragen. Software hat ganz spezielle Eigenschaften: Anders als Land oder Wasser ist Software immateriell und einfach zu vervielfältigen; anders als in der Gentechnik sind Programmierkenntnisse relativ einfach zu lernen und Programmierfehler in der Regel leicht rückgängig zu machen; die Nicht-Rivalität von Software machte die Strategie der Konstruktion von Alternativen erst möglich.

Dennoch gibt es in der Freien-Software-Bewegung Elemente, die von anderen Bewegungen auf ihr Spezialgebiet angepasst und angeeignet werden können. Das versuchen derzeit die Bewegungen für freie Kultur und für freie Netze. Diesen Ansatz gibt es auch in den relativ neue Bemühungen, Konflikte unter dem Aspekt der Gemeingüter neu zu formulieren, wie es Teile der Ökologischen Bewegung oder Aktivist*innen für traditionelle Landwirtschaft und freien Zugang zu Medizin oder Wissen tun.

«Die Entscheidung zur Migration² an sich ist schon wichtig, und der Prozess der Migration kann recht langwierig sein. Den größeren Widerstand leisten die Erwachsenen, die sich an das Gegebene gewöhnt haben. Sie hinterfragen es nicht und denken auch nicht darüber nach, ob es Alternativen geben könnte. Es kommt häufig vor, dass sie vor einem Bildschirm mit Piktogrammen, die anders aussehen, als jene, die sie gewohnt sind, einfach sagen, sie verstünden das nicht. Sie haben gelernt, Programme einer bestimmten Marke zu nutzen und nicht darüber nachzudenken, welche Prozesse ausgelöst werden, wenn sie etwas Bestimmtes tun. Wenn man ihnen dann aber zeigt, wie lange ihr Computer [mit freier Software – Anm. d. Red.] schon ohne jegliche technische Unterstützung läuft, dann sind sie überrascht. Und trotzdem würden sie nie ihre privaten Computer migrieren, weil «das nicht alle Welt benutzt». Sie sind bequem, und aus dieser Bequemlichkeit heraus wird das, was man ändern kann, nicht angegangen. Für die Jungen hingegen gibt es diese Hürden nicht. Sie neigen dazu, die Herausforderungen anzunehmen, weil sie wissen, dass es mehr Spaß macht, und sie sagen einfach: «Wenn es den gleichen Zweck erfüllt, warum nicht?»

María Eugenia Nuñez

2 Der Wechsel von einem Softwaresystem zum anderen. [Anm. d. Red.]