



## Benjamin Mako Hill - Freiheit für Nutzer, nicht für Software

[Benjamin Mako Hill](#)

Richard Stallman begründete 1985 die Freie-Software-Bewegung und veröffentlichte ein Manifest, in dem er Computernutzer aufforderte, sich mit ihm gemeinsam für Software einzusetzen, die ihren Nutzern gewisse Freiheiten garantiert (Stallman 2002). Er publizierte eine Definition Freier Software, die Free Software Definition (FSD), die die Rechte eines jeden Nutzers aufzählte, nämlich die Freiheit,

- das Programm zu jedem Zweck auszuführen;
- die Funktionsweise eines Programms zu untersuchen und es an seine Bedürfnisse anzupassen;
- Kopien weiterzugeben und damit seinen Mitmenschen zu helfen;
- ein Programm zu verbessern und die Verbesserungen an die Öffentlichkeit weiterzugeben, so dass die gesamte Gesellschaft profitiert.[1. Siehe unter: <http://www.gnu.org/philosophy/free-sw.de.html> (Zugriff am 07.10.2011).]

Als Informatiker verstand Stallman, wie Programmierer Software so gestalten können, dass sie Einfluss auf die Benutzung ihrer Programme haben. Beispielsweise können Programmierer Software schreiben, die ihre Nutzer ausspioniert, die gegen sie arbeitet oder die Abhängigkeiten schafft. Da die Kommunikation und das Leben der Menschen mehr und mehr vom Computer beeinflusst wird, wird auch ihre Erfahrung zunehmend durch die entsprechende Technologie beeinflusst und infolgedessen durch diejenigen, die die Software kontrollieren. Wenn Software »frei« ist, können sich Nutzer von Funktionen befreien, die ihnen nicht dienen. Sie können zusammenarbeiten, um ihre Technologien zu verbessern und sie selbst in die Hand zu nehmen. Nach Stallmans Ansicht ist Freie Software essentiell für eine freie Gesellschaft.

Leider dachten viele Menschen, die »Freie Software« hörten, dieses »frei« bedeute vor allem kostenfrei. Das führte verständlicherweise zu Verwirrung, denn Freie Software kann ohne Erlaubnis oder Bezahlung weitergegeben werden, und so geschieht es meist auch. Im gemeinsamen Bemühen, die Verwirrung aufzuklären, wurde der Slogan »free as in ›free speech‹, not as in ›free beer‹« (»Frei wie in ›Freiheit‹, nicht wie in ›Freibier‹«) geprägt, der auf den Unterschied zwischen den lateinischen Worten »liber« und »gratis« verweist. Der Spruch haftet heute der Bewegung an wie ein Klischee. Eine Biografie von Stallman trägt den Titel Free as in Freedom (Williams 2002).

In den späten 1990er-Jahren schlug eine Gruppe von Enthusiasten für Freie Software einen neuen Begriff vor: Open Source. Wie Stallman war auch diese Gruppe von der Mehrdeutigkeit des Wortes »frei« frustriert. Allerdings war das Hauptanliegen der Open-Source-Befürworter, Freie Software auch für Unternehmen attraktiv zu machen. Sie meinten, gewinnorientierte Unternehmen würden den Begriff »Freie Software« abstoßend finden, beschrieben die technischen Vorzüge, die die »Offenheit« Freier-Software-Entwicklung durch die Zusammenarbeit der Nutzer in großen Netzwerken bringen könnte, und vernachlässigten die Betonung der »Freiheit«. Diese Aufrufe fanden bei High-Tech-Firmen um die Jahrtausendwende Anklang, als sich das auf Freier Software basierende GNU/Linux-Betriebssystem zunehmender Beliebtheit erfreute und der Apache-Webserver einen Markt dominierte, auf dem sich vor allem Wettbewerber tummelten, die mit proprietärer Software arbeiteten. Das »Open-Source«-Konzept erlebte 1998 einen weiteren Schub, als Netscape den Quellcode seines Browsers Navigator veröffentlichte.

Trotz der rhetorischen und philosophischen Unterschiede bezeichneten »Freie Software« und »Open Source« dieselbe Software, dieselben Communitys, dieselben Lizenzen und dieselben Praktiken in der Entwicklung. Die Definition von Open Source war eine fast wortgleiche Kopie der Richtlinien für Freie Software, die die Debian-Free-Software-Community herausgegeben hatte; und sie war sogar ein Versuch, Stallmans Definition Freier Software neu zu formulieren. Stallman hat die Spaltung zwischen Freier Software und Open Source als das Gegenteil eines Schismas beschrieben. Bei einem Schisma beten zwei religiöse Glaubensgemeinschaften aufgrund miteinander geringfügiger Unstimmigkeiten über Liturgie oder Doktrin getrennt. Im Fall von Freier Software und Open Source haben beide Gruppen fundamental unterschiedliche Philosophien, politische Einstellungen und Motivationen artikuliert – dennoch arbeiten sie weiterhin eng zusammen.

In den jeweiligen Communitys für Freie Software und Open-Source-Software überschatteten die Diskussionen zum Thema »liber« und »gratis« eine zweite, viel weniger diskutierte Ebene der sprachlichen Mehrdeutigkeit des Begriffs »Freie Software«: Er führte dazu, dass die vier Freiheiten als Aussagen über Eigenschaften verstanden wurden, die die Software selbst haben sollte. Natürlich ist es Stallman gleichgültig, ob die Software selbst frei ist; ihm geht es vielmehr um die Freiheit der Software-Nutzer. Die Slogans »Frei wie ›Freiheit‹« und »Nicht frei wie in ›Freibier‹« helfen bei der Auflösung dieser Art Zweideutigkeit nicht weiter. Sie können sogar noch mehr verwirren. »Frei wie in ›Freiheit‹« sagt einfach nichts darüber aus, was frei sein soll; während das Argument, es gehe um »Meinungsfreiheit, nicht Freibier« sogar noch ein Parallelproblem reproduziert: Befürwortern der Meinungsfreiheit geht es schließlich nicht um die Freiheit der Sprache, sondern um die Freiheit, zu sprechen. Wenn die Rhetorik der Freie-Software-Bewegung die Aufmerksamkeit auf die Eigenschaften von Software lenkt, hat das zur Folge, dass manche Menschen die Freiheit der Nutzer als Anliegen zweiter Ordnung betrachten – als folge dies automatisch aus der Freiheit des Codes.

### Wenn Software frei ist, nicht die Nutzer

Die Freiheit der Nutzer ergibt sich aber nicht immer aus der Freiheit der Software. Tatsächlich hat Freie Software mit ihrer wachsenden ökonomischen und politischen Bedeutung auch die Aufmerksamkeit einiger Akteure auf sich gezogen, die einerseits die Vorzüge Freier Software für sich nutzen möchten, während sie andererseits die Nutzer einschränken und in einem Abhängigkeitsverhältnis lassen.

Google, Facebook und andere Titanen der Internetökonomie haben ihre Unternehmen auf Freier Software aufgebaut. Und sie sind nicht bloß Trittbrettfahrer in der Art und Weise, wie sie diese Ressource verwenden; in vielen Fällen veröffentlichen diese Firmen zumindest einen Teil des Codes, der auf ihren Servern läuft, und sie investieren bedeutende Ressourcen in die Schaffung oder Verbesserung dieses Codes.

Jeder Nutzer eines auf Freier Software basierenden Netzwerkdienstes, auch Google und Facebook, kann eine Kopie der Software haben, die die vier Freiheiten der Definition für Freie Software ermöglicht. Aber die Nutzer bleiben der Willkür der Firma, die ihre Kopie einsetzt, unterworfen; es sei denn, die Nutzer betreiben den Internetdienst selbst – was vermutlich technisch oder wirtschaftlich nicht machbar ist.



Software als Dienstleistung (»Software as a Service«, SaaS) – das heißt Software, die über die »Wolke«, die Cloud, bereitgestellt wird – ist völlig kompatibel mit der Vorstellung von Freier Software. Aber dadurch, dass die Nutzer der Dienste »in der Wolke« die Software nicht ändern und nicht nutzen können, wie sie wollen, wenn der Anbieter des Dienstes es ihnen nicht erlaubt (und sie kontrolliert), sind SaaS-Nutzer mindestens so abhängig und verletzlich, als würden sie proprietäre Software nutzen.

Googles Chrome-OS ist ein Versuch, ein Betriebssystem zu bauen, das Nutzer online bringt und direkt mit Diensten wie Google Docs verbindet, so dass die Nutzer ihre Computernutzung überwiegend in dieser Weise gestalten. Als Google Chrome OS ankündigte, haben viele in der Freie-Software-Community gefeiert; Chrome OS ist auf GNU/Linux aufgebaut, ist fast ausschließlich Freie Software und hat die Unterstützung von Google. Aber es ist das Ziel von Chrome OS, den Ort zu verändern, an dem die Rechenleistung erbracht wird, indem Anwendungen, die ein Nutzer auf dem eigenen Rechner laufen lassen könnte, durch SaaS ersetzt werden. Jeder Schritt von einem Stück Freier Software auf dem eigenen Rechner hin zu einem SaaS-Dienst ist ein Schritt von einer Situation, in der Nutzer die Kontrolle über ihre Software hatten, hin zu einer Situation, in der sie kaum noch über Einfluss verfügen. Beispielsweise ermöglicht Googles Einsatz Freier Software in seinen SaaS-Diensten die Überwachung aller Nutzungen und das Hinzufügen oder Entfernen von Features nach Gutdünken. Indem sie sich auf die Freiheit der Software und nicht auf jene der Nutzer konzentrierten, haben viele Unterstützer Freier Software diese besorgniserregende Dynamik nicht ausreichend erkannt.

Der TiVo – ein bahnbrechender digitaler Videorekorder – ist eine weitere Herausforderung dieser Art. Seine Software basierte auf GNU/Linux. In Übereinstimmung mit der »Copyleft«-Lizenz, unter der die meiste Freie Software vertrieben wird, veröffentlichte der TiVo-Konzern den vollständigen Quellcode. Aber TiVo setzte eine Verschlüsselung ein, um sein Gerät so einzustellen, dass es nur bestimmte genehmigte Versionen von Linux laufen ließ. TiVo-Nutzer konnten die TiVo-Software untersuchen und verändern, aber sie konnten die modifizierte Software auf ihrem TiVo nicht nutzen. Die Software war frei, nicht aber die Nutzer.

SaaS, Chrome OS und »TiVoisierung« sind Themen, die die Freie-Software- und Open-Source-Bewegungen weiterhin in Aufruhr versetzen und die philosophischen Bruchlinien aufdecken. Es überrascht nicht, dass Open-Source-Befürworter kein Problem mit SaaS, Chrome OS und TiVoisierung sehen; schließlich fühlen sie sich nicht der Freiheit der Nutzer der Software verpflichtet. Aber jedes dieser Beispiele hat sogar unter Menschen Zwietracht gesät, die die Überzeugung teilen, dass Software frei sein soll. Die Free Software Foundation (FSF) hat zwar gegen jede der oben genannten Themen Position bezogen; aber auch sie erkannte jede Gefährdung nur langsam und hat diese Positionen nur mühsam an ihre Unterstützer kommunizieren können. Heute sieht es ganz danach aus, dass das dienst-orientierte Geschäftsmodell von Google eine größere Bedrohung für die künftige Freiheit von Computernutzern darstellt als das Geschäftsmodell von Microsoft. Da Google aber die Lizenzbedingungen Freier Software peinlich genau einhält und Freie-Software-Projekten riesige Mengen Code und Geld spendet, haben die Befürworter Freier Software die bestehende Bedrohung erst sehr spät erkannt und darauf reagiert.

Sogar die FSF hat weiterhin Mühe mit ihrem eigenen softwarebezogenen Auftrag. Stallman und die FSF haben in den letzten Jahren daran gearbeitet, unfreien Code, der in der Regel auf kleineren Subrechnern läuft (das heißt eine drahtlose Schnittstelle oder ein Grafikauteil innerhalb eines Laptops), von der zentralen Festplatte des Rechners auf die Subprozessoren selbst zu verschieben. Der Zweck dieser Bemühungen ist es, unfreie Software zu eliminieren, indem sie in Hardware verwandelt wird. Doch sind die Nutzer von Software tatsächlich freier, wenn in ihrem Rechner Technologien, die sie auf Grund des Eigentumsrechts nicht verändern können, eine andere Form annehmen, aber dennoch bestehen bleiben?

Der Schlüssel zur Beantwortung dieser und anderer Fragen liegt in der Konzentration auf die Unterscheidung zwischen »frei« und »offen«. Befürworter Freier Software müssen zu ihrem großen Ziel zurückkehren, die Menschen zu befreien, nicht die Software. Die grundlegende Innovation von Stallman und der Freie-Software-Bewegung war es, Fragen individueller Autonomie und Freiheit mit Bereichen zu verbinden, in denen die meisten Menschen die Relevanz von Freiheit und Autonomie nicht erkannten. Da sich die Natur der Technologie permanent ändert, wird sich auch die Art und Weise verändern, die Freiheit der Nutzer zu verteidigen. Und wenn andere die Prinzipien Freier Software auf neue Bereiche übertragen und anpassen, werden sie mit ähnlichen Problemen konfrontiert sein. Nur in dem Maße wie unsere Communitys in der Lage sind, zwischen der »Offenheit« von Artefakten, also von Menschen gemachten Dingen einerseits und den Fragen der Kontrolle, der Politik und der Macht andererseits, zu unterscheiden, wird die Freie-Software-Philosophie in den Diskussionen über neue und klassische Commons – gleich ob Software oder nicht – relevant bleiben.

#### Literatur

- Stallman, Richard M. (2002): Free Software, Free Society: Selected Essays of Richard M. Stallman, Free Software Foundation.
- Williams, Sam (2002): Free As in Freedom: Richard Stallman's Crusade for Free Software, Sebastopol.