



Fakultät IV
Fachgebiet Informatik & Gesellschaft

Information Rules 1

Open Source

Kai Sisterhenn,
Lukas Neumann,
Stefan Laufmann
Viet Phuong Nguyen

16. November 2012

Inhaltsverzeichnis

1	Einleitung	1
2	Firmenmodelle	1
2.1	Frischer Wind	1
2.2	Firmenunterstützung	1
2.3	Teilfinanzierung	2
3	Schluss	2
	Literaturverzeichnis	3

1 Einleitung

Die folgende Ausarbeitung konzentriert sich auf das Vermitteln einiger Grundfaktoren für Firmen, die mit Open Source Software (OSS) Gewinne erwirtschaften wollen. Dabei werden wissenschaftliche Quellen bekannter und anerkannter Autoren verwendet, aus welchen die Quintessenz der Zielpunkte für eine solche Firma extrahiert werden. Diese Ausarbeitung soll einen Einstieg in Open Source Software Geschäftsmodelle bieten und somit besonders für den ungelehrten Leser den Irrglauben, dass man mit Software nur durch möglichst teures Verkaufen Geld machen kann, beseitigen.

2 Firmenmodelle

Der Beginn von Open Source Software ist mit dem Beginn der Hackerkultur am MIT¹ in den 80er Jahren zusammenzulegen. Die damaligen Administratoren der Großrechner waren sehr restriktiv mit Rechenzeit und ihrem Code, was die damals nur durch eben diese Beschränkungen gehinderten Hacker schnell den Konsens aufstellen ließen, dass alle Information frei sein müsse [vgl. Clu]. Während in der kommenden Zeit Firmen und „professionelle“ Entwickler ihren Code fast ausschließlich proprietär verkauften, veröffentlichten Hobby-Programmierer, Software-Enthusiasten und Hacker ihren Code durchweg frei und mit minimalen Beschränkungen für die Nutzung und Weiterentwicklung. Spätestens seitdem Red Hat mit ihrem Linux-Betriebssystem als Firma die Grenze von Einnahmen in der Höhe von 1 Milliarde Dollar geknackt hat [vgl. McM12], ist klar, dass Open Source Software durchaus profitabel für Firmen ist. Dabei spielen die folgenden 3 Punkte die größte Rolle.

2.1 Frischer Wind

Viele Closed-Source Projekte sind sehr alt, seit sehr langer Zeit auf dem Markt, dadurch nicht schlecht, aber resultierend daraus meist stark in ihren Grundsätzen eingefahren. Neue Programmierer sollen nichts verändern, was irgendwie

funktioniert, oft fehlt auch das Verständnis einiger Funktionen aufgrund von bereits gekündigten Programmierern; das Projekt stagniert. Genau hier kann ein OSS Projekt den Markt neu aufrollen, da folgende Dinge vorteilhafter sind:

- Es gibt meist mehr Programmierer, da ein interessantes Projekt Menschen anlockt.
- Es gibt oft Programmierer, die ihr Handwerk wirklich verstehen. In einem proprietären Projekt müssen Funktionen von Mitarbeitern implementiert werden, obwohl es nicht unbedingt ihr Spezialgebiet betrifft. Ein offenes Projekt lockt die Menschen an, die genau in diesem speziellen Bereich Erfahrungen haben, wodurch diese dann wesentlich besseren Code beisteuern und die Qualität des Projektes insgesamt höher sein kann, wie Hecker [Hec99, S. 50] treffend formuliert:

„Geeignet organisiertes und koordiniert verteiltes Entwickeln kann mehr Produkte schneller und mit höherer Qualität produzieren, als es Einzelnen möglich wäre.“²

- Ein frisches OSS-Projekt kann von Anfang an auf den bestehenden Systemen aufbauen und dadurch besseren und schnelleren Code produzieren, als es eingefahrenen, proprietären Systemen möglich wäre, da diese auch in den wenigsten Fällen überhaupt das Geld hätten, einen kompletten Code-Rewrite zu finanzieren.

2.2 Firmenunterstützung

Open Source Software ist in erster Linie von vielen Menschen für viele Menschen gemacht. Sie weckt die Motivation des Einzelnen, etwas beizutragen, was von Vielen genutzt wird. Das Endprodukt ist meist aus diesen und oben schon genannten Gründen auf dem gleichen Stand wie proprietäre Software.

Wenn nun genau hier eine Firma ansetzt und Vollzeitprogrammierer, Hardware-Partnerschaften

¹Massachusetts Institute of Technology, <http://mit.edu>

²Eigene Übersetzung, im Original: *“Properly organized and coordinated, distributed development can produce more products faster and with higher quality than would be possible in an isolated effort.”*

oder Standards zum Projekt beitragen kann, heben es diese Beiträge komplett von der proprietären Software ab. Das führt zu einem Wechsel der Software beim Anwender, welcher nun potentiell die bessere Software benutzt, die zwar Open Source ist, aber dann durch „Vertrieb, Support und Integration“ [vgl. Val98] monetarisiert wird. Der finanzielle Vorteil der proprietären Software, Geld mit der Software selbst zu verdienen, trifft erstmal nur auf Neukunden zu, welche, noch ohne Wechselkosten, natürlich die bessere Software benutzen. Da die OSS-Firma neben dem Support natürlich nur ein Ziel hat, nämlich die Wechselkosten für Nutzer zu reduzieren, ist diese im Vorteil gegenüber den alteingesessenen Softwareschmieden und hat darüber hinaus noch weniger Geld für diesen Stand der Dinge ausgeben müssen. Ab diesem Punkt hat die proprietäre Softwarefirma nun massiv Geld auszugeben, um den Rückstand an Softwarequalität aufzuholen, verliert Geld durch fehlende Neu- und wechselnde Altkunden und die Konkurrenz. Das Fazit aus dieser Situation ist, dass die proprietäre Firma in einem Qualitätskampf mit der, durch eine Firma unterstützte, Open Source Software steht, in welchem OSS durch spezialisiertere Programmierer die Oberhand hat. Darüberhinaus profitiert der Kunde, egal, welchen Ausgang dieses ökonomische Spiel nimmt.

2.3 Teilfinanzierung

Software kann auch nur teilweise als OSS bereit gestellt werden. Es können je nach Zielgruppe (Privat oder Geschäftskunden) oder nach Funktionsumfang verschiedene Versionen vermarktet werden. Die Basiskomponenten können offen gelegt werden, aber für spezialisiertere Funktionen muss der User zahlen [vgl. Haf08]. Dadurch verliert die Firma den Verkaufspreis nicht komplett, kann aber gleichzeitig von den OSS-Faktoren profitieren, nämlich mehr und fachlich passendere Programmierer und mehr Aktivität im Sourcecode.

Unter dem Teilfinanzierungsaspekt kann man zusätzlich eine Firma betrachten, welche Hardware verkauft und dafür passende Open Source Software finanziell unterstützt oder selbst entwickelt, um eben diese Hardware attraktiver zu machen. Hecker [Hec99, S. 46] konstatiert dazu:

„Eine Firma kann Quellcode frei verfügbar machen und trotzdem den eigenen Geschäftsinteressen als profitorientierte Organisation gerecht werden.“³

Abseits davon kann eine Firma mit Komplementärgütern einiges an Geld machen, bspw. durch Bücher oder kostenpflichtigen Apps, die auf der Software basieren oder mit dieser interagieren.

3 Schluss

Wie nun klar zu sehen ist, ist Open Source Software keineswegs mehr eine Randerscheinung und Software braucht keineswegs eine große Firma, welche diese im Alleingang produziert. Als Zusammenfassung lässt sich sagen, dass Open Source Software etwa gleichauf in der Qualität mit proprietärer Software ist. Wenn nun eine Firma die Qualität mit Geld als Arbeitsstunden noch fördert, kann durch diese, relativ billig entstandene, Software viel Geld mittels Support und Integration erwirtschaftet werden.

Der Nutzer gewinnt darüber hinaus immer, da die Software besser, kostenlos und mit einem starken Service ausgestattet wird. Außerdem ist die Freiheit, die Software zu benutzen, zu verändern und so anzuwenden, wie es am besten passt, nicht nur aus ethischer Sicht hoch einzuordnen. Da der Nutzer die Software nun nach seinen individuellen Bedürfnissen kauft, kann er von einem teuren Komplettpaket differenzieren und aus Zusatzfeatures, Kundenservice oder spezieller Integration frei wählen.

³Eigene Überstzung, im Original "A company can choose to make source code freely available and still serve its own business interests as a for-profit organization."

Literaturverzeichnis

- [Clu] Chaos Computer Club. *hackerethics*. URL: <http://www.ccc.de/hackerethics> (besucht am 02.07.2012).
- [Haf08] Christian Hafner. *Open Source Software und innovative Möglichkeiten zur Kommerzialisierung*. Grin Verlag, 2008, S. 60. ISBN: 3-640-11666-6.
- [Hec99] F. Hecker. „Setting up shop: The business of open-source software“. In: *Software, IEEE* 16.1 (Jan. 1999), S. 45–51. DOI: [10.1109/52.744568](https://doi.org/10.1109/52.744568).
- [McM12] Robert McMilan. *Red Hat Becomes Open Source's First \$1 Billion Baby*. Online-Newseintrag. 28. März 2012. URL: <http://www.wired.com/wiredenterprise/2012/03/red-hat/> (besucht am 02.07.2012).
- [Val98] Vinod Valloppillil. *Halloween Document I*. Geleakte interne Kommunikation. 11. Aug. 1998. URL: <http://www.catb.org/~esr/halloween/halloween1.html> (besucht am 02.07.2012).