

2 Free / Libre Open Source Software-Unternehmen

Produkte aus Open Source Communities sind frei verfügbar. In Kapitel zwei wird der Frage nachgegangen, wie diese in eine Strategie für profitorientierte Unternehmen passen. Zunächst wird deshalb eine Definition von FLOSS vorgestellt, Entwicklung und Prozesse werden aufgezeigt und schließlich werden die Themen Marktstrukturen, Wettbewerber und Strategien von FLOSS-Unternehmen bearbeitet.

2.1 Der FLOSS-Begriff

Free/libre Open Source Software hat sich zu einem strategischen Vorteil in der Softwareentwicklung etabliert. Bis 2016 haben bis zu 90% der wichtigsten IT-Unternehmen auf dem globalen Markt als strategisches Ziel definiert, freie Software in ihrer Unternehmensstrategie zu verankern.⁸

Freie Software ist auf alle gewöhnlichen Elemente angewiesen, die auch von hoch qualitativer, proprietärer Software erwartet werden: Design, Programmierung, Tests und Wartungen. Der Unterschied besteht lediglich in der Lizenzierung. Freie Software hat den Vorteil, dass keine Lizenzgebühren anfallen, wie dies für proprietäre Softwareprodukte wie beispielsweise Windows oder SAP der Fall ist.

Im Mittelpunkt von Freier Software steht der Nutzer, gleichgültig ob dieser eine Person, eine Organisation oder ein Unternehmen ist. Freie Softwarelizenzen basieren auf vier Grundfreiheiten, die jedem Endnutzer vom Urheber der Software garantiert werden:

- Freiheit 0: Die Freiheit das Programm zu betreiben, ohne dass ein bestimmter Zweck vorgegeben ist
- Freiheit 1: Die Freiheit zu lernen, wie das Programm arbeitet und es für die eigenen Bedürfnisse zu adaptieren. Dafür ist der Zugang zum Quellcode Voraussetzung.
- Freiheit 2: Die Freiheit den Code weiterzuverbreiten, so dass Nutzer sich gegenseitig helfen können.
- Freiheit 3: Die Freiheit das Programm zu verbessern und Neuerungen an die Öffentlichkeit weiterzugeben, damit alle davon profitieren.⁹

⁸ Driver, M., (2012).

⁹ Vgl. Stallmann, R.M., (1999).

Für die Abgrenzung von freier zu proprietärer Software spielen die Lizenzierungen eine wesentliche Rolle. Hierbei ist zu unterscheiden zwischen der Copyleft Lizenz für freie Software und der Copyright Lizenz für proprietäre Software. Die Copyleft Lizenz garantiert dem Nutzer die vier genannten Grundfreiheiten. Die Copyright Lizenz stellt das Softwareprodukt unter einen Kopierschutz und wird von proprietären Softwareunternehmen genutzt.¹⁰

Der Begriff „frei“ bezieht sich in der deutschen Sprache nicht auf kostenlos, sondern auf „frei“ im Sinne von freiem Zugang und freier Nutzung. Zu den Begriffen *freie Software*, *Open Source Software* und *Libre Software* gibt es verschiedene Ansichten.

Freie Software definiert sich über die genannten vier Grundfreiheiten, welche keine Pflichten beinhalten, sondern die Rechte der Nutzer darstellen. Jede Person kann wählen, ob sie diese Freiheit in Anspruch nimmt oder nicht. Diese Freiheit bezieht sich auch auf kommerzielle Anwendungen und Verbreitung der Software und bildet damit die Grundlage verschiedenster Geschäftsmodelle im Bereich der freien Software.¹¹

Die Begriffe *Open Source Software* und *Libre Software* sind dem Begriff *Freie Software* zuzuordnen. Die Free Software Community ist groß und vereinigt verschiedene Interessen. Die unterschiedlichen Begriffe beschreiben verschiedene Sichtweisen auf *Freie Software*. Während der Begriff *Open Source Software* darauf abzielt, dieses Softwareentwicklungsmodell auch in der Wirtschaft zu verbreiten und somit technische Neuerungen voranzutreiben (technische Motivation) wurde der Begriff *Open Source Software* als Teil einer Marketingstrategie gewählt. Dagegen bezeichnet der Begriff *Freie Software* eher die sozio-ökonomische Bewegung der Entwicklung von Grundfreiheiten, die hinter der rein technischen Innovation steht.¹² Damit sollen die wirtschaftlichen Aktivitäten um Open Source und die Open-Source-Bewegung mit ihren gesellschaftlichen und politischen Forderungen miteinander verknüpft werden.

Libre Software ist ein von Europäischen Union geprägter Begriff, der im Rahmen ersten Studien über die Wirkung von *Freier Software* auf die Wirtschaft entwickelt wurde, um einen einheitlichen Begriff zu finden, der alle Interessengruppen und dahinter stehende Motivation einschließt.

Für die vorliegende Masterarbeit und die Gespräche mit den Experten zu dem Thema wurde der Begriff *Free / Libre / Open Source Software*, in Abkürzung *FLOSS* gewählt, um alle Interessengruppen und Motivationen hinter der Freien Software Bewegung abzubilden.

¹⁰ Vgl. Schießle, B. (2012). [Online].

¹¹ Vgl. Open Source Initiative (2002). [Online].

¹² Ebenda.

2.2 Entstehung von FLOSS-Projekten

Schon in den 1930er Jahren wurden von IBM Nutzergruppen eingerichtet, um mit den Kunden Informationen über die Datenverarbeitung zu teilen. In den 1970er Jahren kam zudem eine quelloffene Software hinzu, die es den Nutzern von IBM-Software ermöglichte, daran mitzuarbeiten. Parallel dazu entwickelte sich seit den 1960er in US-Universitäten eine Kultur des Verbesserns, Anpassens und Teilens von Software unter den Programmierern. Immer mehr Quellcodes wurden öffentlich zur Verfügung gestellt und mit dem Nutzer geteilt. Mit dem wachsenden Softwaremarkt nahm die Konkurrenz unter Anbietern wie Microsoft, Apple oder Comodor zu, so dass die Unternehmen Softwarelizenzen einführten, um ihre Produkte zu schützen. Der Quellcode für diese Softwareprodukte wurde nicht mehr ausgeliefert und somit konnte nur noch der Anbieter selbst darauf zugreifen. Damit wurde Software proprietär.¹³

Der Umschwung von freier zu kostenpflichtiger Software betraf auch Bereiche des MIT (Massachusetts Institute of Technology) und so begann Richard Stallmann Mitte der 1980er Jahre mit dem GNU-Projekt¹⁴ Software unter freier Lizenz der *General Public License* zu entwickeln. An dieser Software konnte jeder Nutzer mitarbeiten und entwickeln. Dieser Gegenbewegung zu proprietären Software entspringen auch die oben genannten Grundfreiheiten als Grundlage der General Public Lizenzen, die den Softwaremarkt revolutionierten.¹⁵ Eine der ersten freien Softwareprodukte aus den 1990ern ist das Betriebssystem Linux von Linus Torvalds, das er als alternative zu Windows und Macintosh entwickelte.¹⁶

2.3 Communitys als Produzenten von FLOSS

Das Besondere an FLOSS ist die offene Entwicklung der Software. Der Produzent der Software ist letztlich die Community der individuellen Nutzer. Dabei macht es keinen Unterschied ob Nutzer individuelle oder gewerbliche Interessen beim Entwickeln oder Weiterverwenden der Software verfolgen.¹⁷ Jeder kann mitmachen und die Software zum allgemeinen Nutzen weiterentwickeln.

Die wichtigste Entscheidung, die die Community trifft, ist die Bestimmung der Lizenz, da nach dem Prinzip von FLOSS jeder mit dem Code arbeiten kann. Bei freien Softwareprojekten gibt es verschiedene Lizenzsierungsmodelle. Die Community tritt als Produzent

¹³ Vgl. Pugh, E.W., (2002), S.57-58.

¹⁴ Das GNU Projekt ist ein Betriebssystem das freie Software ist und von Richard Stallman entwickelt wurde.

¹⁵ Vgl. Williams, S. (2002).

¹⁶ Vgl. Torvalds, L. / Diamon, D., (2002).

¹⁷ Vgl. Krishnamurthy, S. (2003), S.5.

auf. Sie schreibt den Quellcode und stellt ihn unter freier Lizenz. Sie entwickelt in den meisten Fällen nicht nur Quellcode sondern ein Produkt, wie ein Betriebssystem oder ein Browser beispielsweise. Zudem entwickeln die vielen Freiwilligen „How to“, „Learning Patterns“ oder stehen für Fragen zur Verfügung. Somit stellen Sie sicher, dass das FLOSS-Projekt auch weiterentwickelt wird. Dieser Prozess ist in der unten stehenden Abbildung dargestellt.

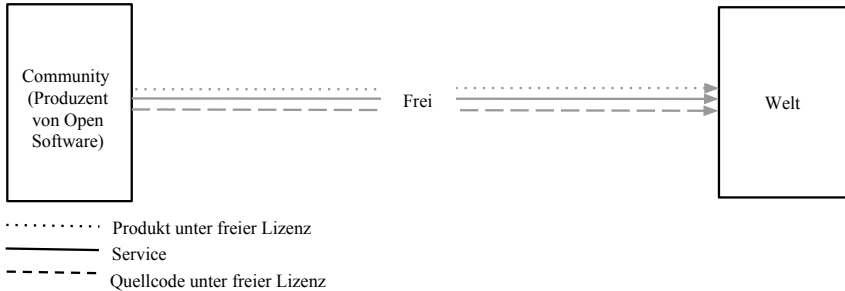


Abbildung 1: *Community als Produzent von Open Software, modifiziert nach Krishnamurthy (2003).*

Raymonds vergleicht Open Source Software Developer mit Händlern in einem Bazar, die selbst bestimmen, wann und wie sie ihren Beitrag leisten wollen. Im Gegensatz dazu vergleicht er nicht freie Software mit der Konstruktion von Kathedralen, die von einem verantwortlichen Architekten konstruiert werden.¹⁸

Die Metapher von Raymonds gilt vielen Autoren als unbefriedigend, da sie die Kernfunktionen, die den Prozess der Open Source Software Entwicklung ausmachen, nicht berücksichtigt.¹⁹ Die Funktionsweise eines offenen Softwareentwicklungsprozesses mit Freiwilligen lässt sich in verschiedenen Modellen beschreiben. Das hypothetische Modell von Crownston ist eines der Modelle, die der Praxis am nächsten kommen. Es beschreibt die FLOSS-Entwicklung anhand einer Teamstruktur. Im Zentrum stehen die Kernentwickler und die Co-Entwickler, welche verantwortlich sind für die Entwicklung und Organisation des gesamten Prozesses, vom Kontrollieren des Inhalts jedes Entwicklungsschrittes bis zur Entscheidung wie der Code in der Software aussieht.²⁰ Dieses Modell entspricht der folgenden Abbildung.

¹⁸ Vgl. Raymond, R.S., (1999), S.99.

¹⁹ Vgl. Crowston, K. et al., (2004), S.4.

²⁰ Vgl. Ebenda.

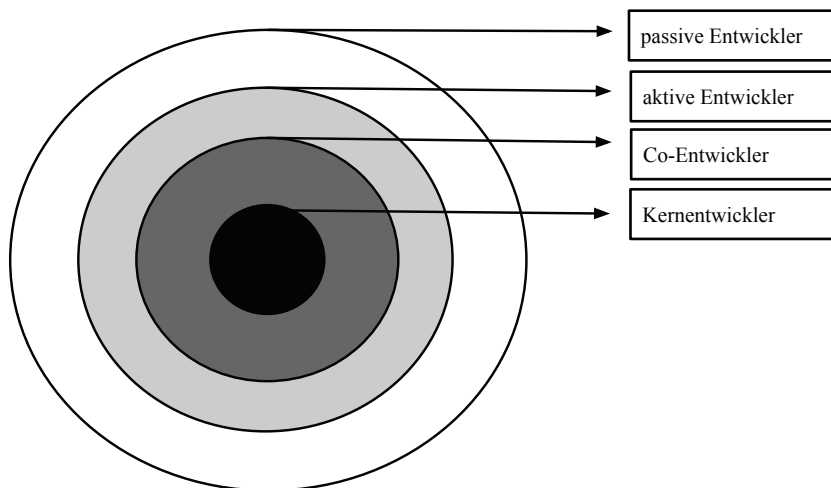


Abbildung 2: *Das hypothetische Modell der FLOSS-Entwicklung anhand einer Teamstruktur nach Crownston. Eigene Darstellung, nach Crownston, K. et al. (2004), S.4.*

Das entspricht den Kathedralen im Modell von Raynold. Die dritte und vierte Ebene sind geprägt von aktiven und passiven Nutzern, welche dem Bazaar aus Raynolds Modell entsprechen. Sie nehmen wann und wie sie wollen am Entwicklungsprozess des Projekts teil.²¹ Grundsätzlich sind Transparenz der Ziele, Entscheidungen oder die langfristigen Visionen des Projektes die Treiber von Open Source Softwareprojekten. Das Projektkernteam der ersten und zweiten Schicht muss dafür offen sein, also vielseitig kommunizieren, um damit den nachfolgenden Schichten die Weiterentwicklung des Projekts zu ermöglichen.²² Diese besteht aus Programmierern, die sich auf die Weiterentwicklung des Codes fokussieren. Das macht ungefähr 4% der Entwickler. Diese Schicht bildet somit die zweite Ebene. 96% der Beitragenden an Open Source Softwareprojekten sind Programmierer, die Fehler im Quellcode reparieren, was 70% aller Beiträge ausmacht.²³ Insgesamt werden zwei Drittel von FLOSS von Einzelpersonen geschrieben. Unternehmen beteiligen sich mit 15% und andere Institutionen mit 20%.²⁴

Insgesamt arbeiten 5% aller Entwickler bei mehr als zehn Open Source Software Projekten mit. Es kann also von einer hohen intrinsischen Motivation innerhalb der Open Source Software Bewegung gesprochen werden.²⁵

²¹ Vgl. Ebenda.

²² Vgl. Raymond, R.S., (1999), [Online].

²³ Vgl. Dempsey, J.B. et al. (1999), [Online].

²⁴ Vgl. EU FLOSS Impact Studie.

²⁵ Vgl. Hars, A., / Ou, S., (2002). S.25-39.

Ein wichtiger Bestandteil des Zusammenwirkens einer FLOSS Community sind die kollaborativen Lösungs- und Entscheidungsfindungsprozesse.²⁶ Ideen, Änderungen und Weiterentwicklungen finden in Abstimmungsprozess innerhalb der Community statt.²⁷ Somit werden anders als im Management von Unternehmen, Entscheidungen von „oben nach unten“ transportiert. Hier werden Entscheidungen gemeinsam beschlossen. Zwar dauern diese partizipativen Prozess länger, sind jedoch fester Bestandteil der FLOSS-Kultur und Indikator für Innovationen, da jede Idee eine Chance hat gewählt und umgesetzt zu werden.²⁸

2.4 FLOSS-Projekte und Organisationen

Die Beteiligung an Open Source Software Projekten ist jedem freigestellt. Es ist ein auf Freiwilligkeit beruhendes Konzept. Da Open Source jedoch häufig die Grundlage kommerzieller Software ist, liegt vielen Beteiligten daran, die Projekte durch die Gründung von Vereinen oder Stiftungen zu unterstützen wie z.B. *Mozilla Stiftung*, *Linux Foundation*, *KDE e.V.* oder *Red Hat Foundation*.²⁹ Über diese Organisationsformen können Spenden generiert, Konferenzen und Treffen der Communities veranstaltet oder an Netzwerken und Arbeitsgruppen gearbeitet werden. Für FLOSS-Unternehmen ist diese Form der Unterstützung wichtig auch außerhalb von Unternehmensstrukturen, da Vereine oder Stiftungen nachhaltige Organisationsstrukturen sind, um die Communitys auch unabhängig vom Unternehmen zu fördern. Gesellschaftliche und politische Interessen können über Stiftungen, Vereine unterstützt werden und der rechtliche Rahmen eines Vereins bietet ähnliche partizipative Entscheidungsprozesse, wie in den Communities selbst.

2.5 FLOSS-Markt und Wettbewerb

Egal ob Webserver, Betriebssysteme, Desktop Oberflächen, Webbrowser, Datenbank oder Email-Systeme, FLOSS-Anwendungen spielen in fast allen Bereichen einer IT-Infrastruktur eine wichtige Rolle. Das folgende Diagramm zeigt das exponentielle Wachstum von Open Source Software Projekten in den letzten 20 Jahren.³⁰

²⁶ Vgl. Heinrich, H. (2008), S.99.

²⁷ Ebenda.

²⁸ Ebenda.

²⁹ Vgl. Open Source.org. [Online].

³⁰ Vgl. Deshpande, A. / Riehle, D. (2008), S.7.

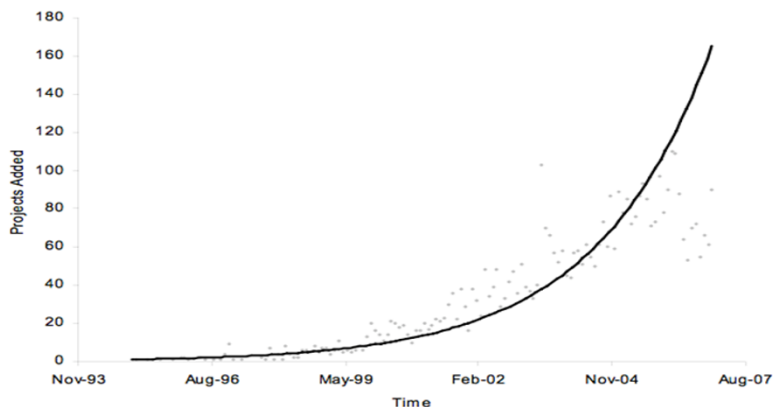


Abbildung 3: *Anzahl und Entwicklung von FLOSS-Projekten, entnommen aus: Deshpande, A. / Riehle, D. (2008), S.7.*

Bedeutung von Open Source nach Branchen			
Branche	unternehmenskritisch / wichtig	weniger wichtig / unwichtig	kein Einsatz
Gesamt	83,8	15,2	1,0
Forschung und Lehre	91,6	7,5	0,9
Software und Consulting	91,0	8,0	1,0
Gewerbe und Handwerk	90,6	7,1	2,4
Telekommunikation	89,6	9,1	1,3
Handel	83,9	16,1	0,0
Medien und Marketing	83,3	16,7	0,0
Sozialwesen, Körperschaften	75,0	25,0	0,0
Verwaltung	70,8	26,1	3,1
Banken und Versicherungen	69,7	27,3	3,0
Industrie	67,0	31,9	1,1

Abbildung 4: *Bedeutung von Open Source nach Branchen entnommen aus Diedrich, Dr. Oliver (2009), [Online].*

In Deutschland spielen FLOSS in fast 80% aller Firmen eine wichtige Rolle.³¹ Zudem finden sich Open Source Anwendungen in so gut wie jeder Branche.

Für Betriebssysteme ist der Marktanteil von FLOSS in Europa höher als in den USA. Die Marktanteile von FLOSS wuchsen in den letzten Jahren enorm an, das belegt eine IDC Studie von 2005 in Westeuropa, in der 625 Unternehmen befragt wurden.³²

40% der Befragten bewerteten den Einsatz von FLOSS-Betriebssystemen als „signifikant, etwas oder beschränkt“ und fast 60% bestätigten die Nutzung von FLOSS-Datenbanken. Gleichgültig, ob im privaten Anwendungsbereich, bei kleinen und mittelgroßen Unternehmen oder im öffentlichen Sektor, Europa ist die führende Nation, wenn in der globalen Zusammenarbeit FLOSS eingesetzt wird. FLOSS wird an erster Stelle in Unternehmensdienstleistungen, an zweiter Stelle bei Finanzdienstleistungen und an dritter Stelle im Gesundheitswesen genutzt. Der Einsatz von FLOSS ist branchenunabhängig somit kann nicht von einem bestimmten Markt gesprochen werden.

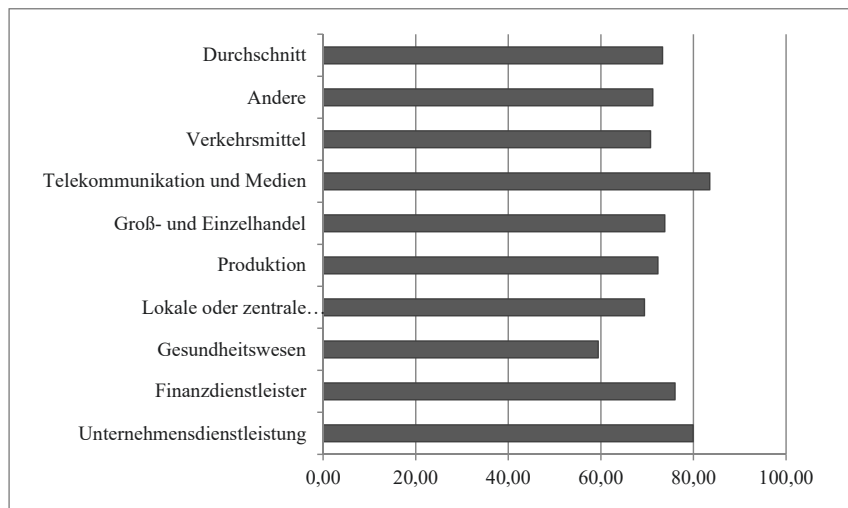


Abbildung 5 *FLOSS-Anwendungen in Europa nach Branche, modifiziert nach Rishab, A.G.(2006), S.26.*

Auch der Einsatz von diversen FLOSS-Produkten ist branchenunabhängig. Im digitalen Zeitalter ist jedes Unternehmen angewiesen auf gute Prozesse zur Effizienzsteigerung.

³¹ Vgl. Diedrich, Dr. Oliver (2009), [Online].

³² Vgl. Rishab, A.G.: (2006), S.26.

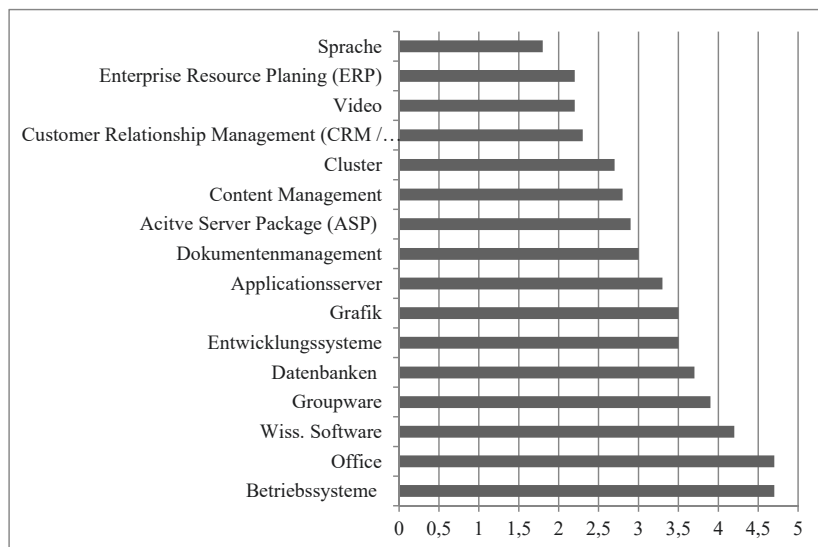


Abbildung 6: FLOSS-Produkte, modifiziert nach Renner, T. et al. (2005), S.32.

Die Ersparnis, die Unternehmen mit FLOSS-Anwendung erzielen können, ist beträchtlich.

“Organizations are saving millions of dollars on IT by using open source software. In 2004, open source software saved large companies (with annual revenue of over \$1 billion) an average of \$3.3 million. Medium-sized companies (between \$50 million and \$1 billion in annual revenue) saved an average \$1.1 million. Firms with revenues under \$50 million saved an average \$520,000.”

Nicht nur die Nutzungsrate von FLOSS hat sich erhöht, auch die Anzahl der Unternehmen und Organisationen, welche Services und Produkte anbieten, die auf FLOSS basieren. Die *Open Software Alliance*, der größte Verband für Open Source Software Unternehmen, zählt in Europa über 190 Mitglieder.³³ Somit kann die FLOSS-Branche als ein schnell wachsender Sektor in einem sich dynamisch verändernden Umfeld beschrieben werden.

³³ Vgl. OSB Alliance (2015), [Online].

2.6 Geschäftsstrategien von FLOSS-Projekten

Die Frage nach unterschiedlichen Geschäftsstrategien von Unternehmen, die im FLOSS-Umfeld tätig sind, ergibt sich aus der steigenden Anzahl von neuen FLOSS-Unternehmen.³⁴

Das grundlegende Geschäftsmodell für FLOSS-Unternehmen – unter der Voraussetzung einer freien Lizenz – ist im Folgenden dargestellt:

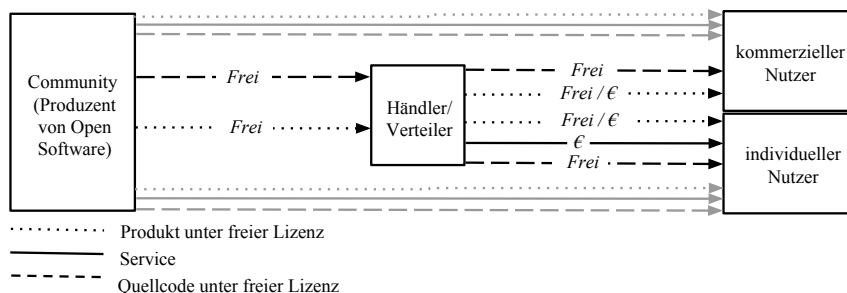


Abbildung 7: FLOSS-Unternehmen Geschäftsmodell modifiziert nach Krishnamurthy 2003.

In diesem Modell stellt die FLOSS-Community als Produzent Softwareprodukte und den Quellcode unter freier Lizenz her. Ein FLOSS-Unternehmen kann dann als Händler auftreten und Services für Unternehmen oder Einzelpersonen für dieses FLOSS-Produkt gegen Bezahlung anbieten, bspw. wenn ein Unternehmen Linux als Betriebssoftware nutzen möchte. Der Quellcode kann vom Unternehmen angepasst werden, wird aber auch unabhängig von den Modifizierungen des Unternehmens aus FLOSS-Projekt heraus über die freie Lizenz zur Verfügung gestellt. Natürlich muss der Endnutzer der freien Software nicht die Dienste des Unternehmens in Anspruch nehmen, um mit der Software arbeiten zu können, denn die Software, das Produkt und Anleitungen vom FLOSS-Projekt frei zur Verfügung gestellt.

Die richtige Geschäftsstrategie zu entwickeln, ist eine Herausforderung für FLOSS-Unternehmen, verschiedene Aspekte müssen berücksichtigt werden. Greve hat dazu ein 3-Achsen-Modell entworfen.³⁵ Zu diesem Modell wird unterschieden zwischen dem Softwaremodell, das den Aspekt der Kontrolle beinhaltet, dem Entwicklungsmodell mit dem Aspekt der Zusammenarbeit und dem Geschäftsmodell mit dem Aspekt des Ertrages abbildet.

³⁴ Vgl. OSB Alliance (2014), [Online].

³⁵ Vgl. Greve, C.F.G. (2008), [Online].

- Die **Software-Modell** Achse ermöglicht es dem Nutzer Kontrolle über die Verwendung der Software zu erhalten, in dem er das Recht hat, die Software zu verstehen, da der Quellcode transparent ist. Darüber hinaus hat er das Recht, die Software zu verwenden, zu modifizieren oder weiterzuverbreiten.
- Die **Entwicklungsmodell** Achse bezieht sich auf den offenen Entwicklungsprozess, der im Abschnitt zur Arbeit der Community (2.3) erläutert wurde. Der Endnutzer kann somit an der FLOSS mit entwickeln, Fehler melden und reparieren. So wird das FLOSS-Projekt weiterentwickelt.
- Das **Ertragsmodell** für FLOSS- Unternehmen wird über die Geschäftsmodellachse beschrieben. Es geht um den Endnutzen beziehungsweise Kunden eine Dienstleistung anzubieten. Dies können Services, Schulungen, Integrationen, Wartungsverträge oder ein spezifisch für den Kunden entwickeltes Produkte sein. Entscheidend bei der Anwendung dieses Modells ist es, die richtige Kombination der gewünschten Elemente für die Geschäftsstrategie zu finden.

Insgesamt entscheiden sich viele FLOSS-Unternehmen für eine Strategie von Dienstleistungen bei der Softwareentwicklung und -anpassung. König hat verschiedene Geschäftsmodelle zusammengefasst³⁶:

Abonnement Strategie (Consulting, Anpassung, Wartung)	Consulting Strategie (Anpassung von FLOSS auf Unternehmensprozesse)
Patronage (Unternehmen betreut Plattform und zieht Vorteile aus der Entwicklung)	Duale Lizenzierung (zweite qualitativ hochwertigere Softwarelizenz wird für kommerzielle Nutzung angeboten)
Anbieten von Software Services (Dienstleistungen kostenpflichtig zur Verfügung stellen)	Optimierungsstrategie (Modularisierung der Software wird kostenpflichtig angeboten)
Eingebettete Strategie (Hardware wird verkauft, welche mit FLOSS z.B. als Betriebssystem läuft)	

Abbildung 8: Geschäftsmodelle FLOSS modifiziert nach König, J. (2004), S.2-6.

Dies geschieht dann meist über Schulungen und technische Unterstützung beim Kunden. Beispiele hierfür sind die *MySQL Datenbank*, *Red Hat* oder *Qt Development Frameworks*. Diese Unternehmen konkurrieren nicht mit FLOSS an sich, denn die FLOSS-Projekte stehen unter freier Lizenz und die Software ist somit öffentlich nutz-

³⁶ Vgl. König, J. (2004), S.2-6.

bares Gemeingut. Die Konkurrenz besteht vielmehr untereinander im Angebot von Dienstleistungen und Services. Für Anbieter proprietärer Software besteht der Wettbewerbsdruck dagegen eher im Lizenzierungsmodell für FLOSS.

Strategisches Management mit Free and Open
Software Communities

Erfolgsfaktoren für Unternehmen

Müllrick, S.

2017, XI, 62 S. 23 Abb., Softcover

ISBN: 978-3-658-17947-2