

Open Source Software im Unternehmen – Ist eine Umstellung sinnvoll?

als
Masterarbeit
an der
Wirtschafts- und Sozialwissenschaftlichen Fakultät
der Universität Bern

eingereicht bei
Dr. Matthias Stürmer
Institut für Wirtschaftsinformatik
Forschungsstelle Digitale Nachhaltigkeit

von
Roman Schmid
von Oberegg AI
im 10. Semester
Matrikelnummer: 10-053-106

Studienadresse:
Bahnhöheweg 30
3018 Bern
(Tel. +41796934758)
(E-Mail: roman.schmid@mynetwork.li)

Bern, 07.08.2015

Zusammenfassung

Obwohl Open Source Software (OSS) in den letzten Jahren zu einer gleichwertigen Alternative zu proprietärer Software geworden ist und sich stark entwickelt hat, wird sie dennoch vergleichsweise wenig und nicht in allen Bereichen eingesetzt. In dieser Arbeit werden OSS und OSSE vorgestellt, evaluiert und dadurch vorhandene Informationsdefizite verringert. Darauf aufbauen wird ein Framework, das drei Funktionen erfüllen soll, entwickelt. Das Framework soll Unternehmen bei der Entscheidung, ob sich eine Umstellung auf OSS lohnt, unterstützen. Weiter soll es über Vor- und Nachteile von OSS für das jeweilige Unternehmen informieren und aufzeigen, ob eine Organisation für die Umstellung auf Open Source bereit ist. Abschliessend wird kurz auf die Rolle der Regierung eingegangen.

Summary

Although Open Source Software (OSS) has become a respectable alternative to proprietary software and has developed and changed tremendously, it is still used relatively little and not exhaustively. The aim of this thesis is to introduce and evaluate OSS and OSSD and in this way reduce existing information deficits. Building on this, a framework, which serves three different purposes, is developed. First, the framework supports companies in their decision whether a change of software systems to OSS is beneficial. Further, the framework should inform what advantages and disadvantages a transition to OSS brings along for the respective company. Thirdly, companies should be provided with an evaluation of whether they are ready to and capable of a switch to OSS. In the last part of this thesis, the role of a government is discussed briefly.

Inhaltsverzeichnis

1	EINLEITUNG	1
2	ENTSTEHUNG VON OPEN SOURCE	5
3	MOTIVATION	9
3.1	Individuen	9
3.1.1	Intrinsische Motivatoren	10
3.1.2	Extrinsisch internalisierte Motivatoren	11
3.1.3	Extrinsische Motivatoren	13
3.2	Unternehmen	14
4	LIZENZIERUNG	22
4.1	Definition der Open Source Initiative	22
4.2	Lizenzarten	23
4.2.1	Sehr restriktive Lizenzen	23
4.2.2	Restriktive Lizenzen	25
4.2.3	Nicht-restriktive Lizenzen	26
4.3	Aktuelle Situation	27
5	OPEN SOURCE SOFTWAREENTWICKLUNG	30
5.1	Unterschiede zur klassischen Methode	30
5.2	Fazit	33
6	EVALUATION VON OPEN SOURCE SOFTWARE	37
6.1	Ethische Aspekte	37
6.2	Technische Aspekte	39
6.3	Ökonomische Aspekte	42
6.4	Fazit	44
7	FRAMEWORK	47
7.1	Grafische Darstellung	49
7.2	Entscheidungsebenen	50
7.2.1	Organisation	50
7.2.2	Personal	59
7.2.3	Umwelt	62
7.3	Erhebung der Kriterien	65
7.4	Bewertung	66
7.4.1	Punktevergabe	66

7.4.2	Gruppen	67
7.4.3	Softwarenutzer	69
7.5	Handlungsempfehlungen für kritische Faktoren	69
8	STAATLICHE FÖRDERUNG VON OPEN SOURCE	72
8.1	Staatliche Interventionen in der Softwareindustrie	72
8.1.1	Argumente für eine Intervention	72
8.1.2	Argumente gegen eine Intervention	73
8.2	Open Source in der Schweizer Bundesverwaltung	74
8.2.1	Interne Sicht	74
8.2.2	Öffentliche Haltung gegenüber Open Source	75
8.3	Das BIT	76
8.4	Open Source beim BIT	78
8.5	Fazit	81
9	ZUSAMMENFASSUNG UND AUSBLICK	86
	ANHANG A	90
A.1	Motivatoren in Open Source	90
A.2	Befunde der Open Source Studie	92
A.3	Fragenkatalog für Framework	94
	ABBILDUNGSVERZEICHNIS	110
	TABELLENVERZEICHNIS	110
	ABKÜRZUNGSVERZEICHNIS	110
	LITERATURVERZEICHNIS	111
	SELBSTÄNDIGKEITSERKLÄRUNG	118
	VERÖFFENTLICHUNG DER ARBEIT	119

1 Einleitung

Im Jahr 2004 verabschiedete die Stadt München das *Linux*-Projekt, in dem beschlossen wurde, den gesamten IT-Betrieb der Stadt auf Open Source umzustellen (Stadt München, 2015). Zur selben Zeit zogen Städte wie Berlin oder Wien (Dobusch, 2014), und verschiedene Landesregierungen eine Migration auf OSS in Erwägung oder hatten diese bereits vollzogen (Evans & Reddy, 2003). Auch Unternehmen wie IBM investieren riesige Beträge im mehrstelligen Millionen- oder sogar Milliardenbereich in die Förderung von Open Source (Lindner, 2013). Zusätzlich wächst die Open Source Entwicklercommunity beständig und rasch weiter. Nicht nur Privatanwender, bei denen eine Umstellung mit vergleichsweise geringen Kosten und Aufwänden verbunden ist, sondern auch Unternehmen oder öffentliche Institutionen verwenden vermehrt Open Source Produkte und helfen bei deren Weiterentwicklung. Verglichen mit proprietärer Software weist OSS höhere Wachstumsraten auf und verzeichnet grösser werdende Marktanteile. Gemäss Herrmann (2008) erreichten Open Source Anwendungen 2008 ein Wachstum von 32 Prozent. In den letzten Jahren setzte sich diese Entwicklung fort. So wuchs der Anteil von OSS alleine in den vier wichtigsten OSS-Gebieten¹ von 2012 bis 2015 um rund 20 Prozent (Open Source Studie Schweiz, 2015, S. 11). Im eher neuartigen Trendbereich des Cloud Computings weisen Open Source Produkte laut der Open Source Studie (2015) sogar ein Wachstum von 42 Prozent auf. All dies zeigt, dass Open Source kontinuierlich an Einfluss gewonnen hat.

Trotz breiter Unterstützung in der Privatwirtschaft und teilweise auch in öffentlichen Verwaltungen ist das Potential von Open Source noch längst nicht ausgeschöpft. Bis heute hat sich OSS insbesondere im Clientbereich noch nicht grossflächig durchsetzen können. Ein möglicher Grund hierfür könnte der geringe Bekanntheitsgrad von Open Source Anwendungen, deren Verwendungsbereichen und Vorteilen sein (Jähnert, 2012). Obwohl bereits ein beachtlicher Katalog an Fachliteratur zu Open Source und seinen Grundzügen

¹ Dazu zählen Web-Server, Programmiersprachen, Datenbanken und Server-Betriebssysteme (Open Source Studie Schweiz, 2015)

besteht und Autoren teilweise auch Handlungsempfehlungen abgegeben haben (z.B. Hars & Ou, 2002; Raymond, 2002; Lerner & Tirole, 2002; Henkel, 2008 oder von Krogh et al., 2012), setzten viele Studien einen zu kleinen Fokus. Sie beschränken sich lediglich auf Unternehmen oder Regierungen an sich, statt die gesamte Umwelt mit all ihren Einflussmöglichkeiten zu betrachten (z.B. Riehle, 2007; Hauge, Ayala & Conradi, 2010; Qu, Yang & Wang, 2011 oder Mijinyawa & Abulwahab, 2014). Nicht nur die Charakteristiken des Unternehmens selber beeinflussen eine Entscheidung für oder gegen OSS. Einflüsse aus dem Umfeld, wie Veränderungen am Arbeitsmarkt oder die rechtlichen Rahmenbedingungen müssen ebenfalls miteinbezogen werden. Gerade die Rolle der Regierung im Zusammenspiel mit dem Unternehmen darf nicht unterschätzt werden. Als einzige in den Softwaremarkt eingreifende Partei hat eine öffentliche Verwaltungsbehörde die Möglichkeit, Rahmenbedingungen zu bestimmen und gleichzeitig auch anzupassen. Die Haltung der Regierung gegenüber OSS hat einen wesentlichen Einfluss, ob Unternehmen OSS verwenden oder nicht. Mitunter deshalb sollte eine Regierung und ihre Behörden intern diejenige Philosophie leben, die auch im Softwaremarkt gewünscht wird. Gemäss Bitzer und Schröder (2005) steigert aufkommende Konkurrenz durch OSS die Innovationsaktivität einer Industrie. Ermöglicht die Regierung also durch das Festlegen der Rahmenbedingungen den Einsatz von OSS, könnte dadurch der Fortschritt in der Softwareindustrie beschleunigt werden.

Auch individuelle Motivatoren müssen betrachtet werden, wenn der Erfolg von OSS untersucht wird. Da Open Source Programmierer meist unbezahlte, private Einzelpersonen sind, die sich zu Entwicklergemeinschaften zusammenschliessen, ist die Frage nach deren Motivation untersuchenswert und sollte bei der Förderung von OSS miteinbezogen werden.

Weiterhin spielt auch die Motivation eines Unternehmens, OSS anzuwenden, zu entwickeln und zu fördern, eine wichtige Rolle für den Fortschritt von Open Source. Wie auch Privatpersonen, investieren Unternehmen bei OSS finanzielle Mittel und Aufwand in etwas, das keinen direkten Ertrag bringt. Im Gegensatz zu Privatpersonen verlangt eine potentielle Umstellung auf OSS bei einem Unternehmen meist eine gründliche Situationsanalyse und hat auf-

grund des strategischen Charakters der Entscheidung langfristige Konsequenzen. Deshalb ist es wichtig, abzuklären in welchem Geschäftsbereich und mit welchem Ziel OSS in und von einem Unternehmen eingesetzt werden soll.

Diese Arbeit soll darüber informieren, was die Vor- und Nachteile von OSS sind und wieso es für Individuen und Unternehmen, aber auch für die Schweizer Regierung sinnvoll ist, Open Source einzusetzen und zu unterstützen. Dabei sollen Informationsdefizite und die vorhandene Skepsis gegenüber Open Source abgebaut werden, die verhindern, dass mehr Open Source Produkte verwendet werden (siehe auch Open Source Studie Schweiz, 2012). Zusätzlich soll ein Framework entwickelt werden, mit dem Unternehmen mit relativ geringem Aufwand eruieren können, ob ihre Organisation für eine Umstellung auf Open Source geeignet und dazu bereit ist.

Im ersten Teil der Arbeit wird Open Source vorgestellt. Dabei werden einige Meilensteine der Entstehungsgeschichte aufgelistet. Anschliessend werden die Faktoren betrachtet, die Individuen und Unternehmen dazu motivieren, Open Source zu unterstützen und anzuwenden. Zusätzlich wird die rechtliche Situation der verschiedenen Open Source Lizenzen beschrieben, sowie die wichtigsten Vor- und Nachteile von OSS evaluiert.

Im zweiten Teil der Arbeit wird das oben erwähnte Framework entwickelt und die einzelnen Elemente genau beschrieben. Das Framework soll Unternehmen bei der Entscheidungsfindung, ob sich eine Umstellung auf OSS für sie lohnt, unterstützen. Hierbei werden nicht nur die wirtschaftlichen Aspekte einer Unternehmung berücksichtigt, sondern auch externe Einflüsse wie rechtliche Rahmenbedingungen, die Situation im Arbeitsmarkt oder die Haltung der Regierung miteinbezogen. Ausserdem werden technische Faktoren und individuelle Komponenten der Mitarbeiter und des Firmenknowhows beachtet. Unternehmen sollen durch dieses Framework über weitere, ihnen bisher vielleicht unbekannt Vorteile von OSS informiert werden. Das erstellte Werkzeug soll es ermöglichen, die aktuelle Situation in einem Unternehmen zu analysieren und anschliessend zu entscheiden, ob ein Umstieg auf OSS sinnvoll ist. Dabei werden in der Literatur beschriebene Konzepte zusammengefasst und durch eigene Kriterien ergänzt. Um die für das Framework benötigten Informationen zu erhalten, wird ein Fragenkatalog erarbeitet, der verwendet werden kann, um die wichtigsten Bewertungskriterien zu erheben.

Im letzten Teil der Arbeit wird gezeigt, dass neben Individuen und Unternehmen auch eine Landesregierung eine wichtige Rolle bei der Förderung von Open Source spielt. Dabei wird untersucht, wie die Schweizer Regierung mit dem Thema Open Source umgeht, was unternommen wird, um OSS zu fördern und ob die Regierung ihrer Rolle als Rahmenbedingungen bestimmende Partei gerecht wird. Danach werden Handlungsempfehlungen abgegeben, wie die Schweizer Regierung ihre Bemühungen zur Förderung von Open Source verbessern könnte. Abschliessend wird kurz diskutiert, wie erreicht werden kann, dass OSS in allen Bereichen als Alternative zu proprietärer Software wahrgenommen und eingesetzt wird.

2 Entstehung von Open Source

Der Begriff *Open Source Software* ist relativ jung; er wurde im Jahr 1998 durch die Open Source Initiative geprägt (Open Source Initiative, 2015a). Die Entstehung von OSS geht jedoch bereits in die frühen 1960er Jahre zurück, als Quellcode in erster Linie von akademischen Institutionen wie der Berkeley University in Kalifornien USA oder dem Massachusetts Institute of Technology (MIT) in Boston USA entwickelt und ausgetauscht wurde (Lerner & Tirole, 2002, S. 200ff). Ziel dieser universitären Zusammenarbeit war es, ein Betriebssystem zu entwickeln, das auf mehreren Computerplattformen verwendet werden konnte. Die Veröffentlichung des Betriebssystems *Unix* im Jahr 1969 durch die Bell Labs sollte die Entwicklung von OSS erheblich beschleunigen. Begünstigt durch die Entwicklung des Usenets² entstand eine schnell wachsende Community von Softwareentwicklern, die sich auf die Verbesserung und Weiterentwicklung von Unix konzentrierte. Zu dieser Zeit wurden Projekte auf einer äusserst informellen Basis durchgeführt und es wurden keinerlei Regeln bezüglich Eigentumsrechten oder der Weiterverwendung des Quellcodes aufgestellt. Diese Art der Projektführung stellte sich als problematisch heraus, als die Firma *AT&T* in den 1980er Jahren anfang zu behaupten, das Betriebssystem Unix gehöre rechtmässig ihr. Als Konsequenz wurde das US-amerikanische Patentgesetz verändert und der Quellcode des Betriebssystems Unix geschlossen (Khanjani & Sulaiman, 2011b, S. 647). Der darauf folgende Rechtsstreit führte dazu, dass erstmals Schritte zur Formulierung grundsätzlicher Regeln in der kooperativen Softwareentwicklung unternommen wurden.

Zwei weitere dieses Jahrzehnt bestimmende Entwicklungen in der Open Source Geschichte waren die Initiierung des *GNU-Projekts* im Jahre 1983, und die 1985 folgende Gründung der *Free Software Foundation* (Henkel, 2008, S. 58). Ziel des GNU-Projekts war es, ein Betriebssystem zu entwickeln, das dem Unix Betriebssystem zwar ähnelte, aber komplett frei zugänglich, also

² Usenet (ursprünglich Unix User Network) ist ein 1979 entwickeltes Computer Netzwerk ähnlich dem heutigen Internet, das die Vernetzung von Unix-Entwickler Communities ermöglichte, mit dem Ziel des schnelleren Austausches von Technologien (Lerner & Tirole, 2002, S. 201)

*free*³, war (Free Software Foundation, 2015a). Um diese versprochene Freiheit zu garantieren, wurde 1989 die *GNU General Public License* (kurz GPL) ins Leben gerufen⁴.

Die flächendeckende Verbreitung des Internets in Industrienationen während den 1990er Jahren führte zu einer weiteren Steigerung der Aktivitäten der Open Source Community. So wurde beispielsweise 1991 das unter der GPL laufende Betriebssystem *Linux* des Softwareentwicklers Linus Torvalds veröffentlicht. Der Begriff der „freien“ Software und die antikommerzielle Haltung der Free Software Foundation wurden aber von vielen Befürwortern quell-offener Software als Hindernis für eine breite Akzeptanz betrachtet (Moody, 2001, S. 166ff). Aus diesem Grund wurde vermehrt der Wunsch geäußert, die Lizenzbedingungen weniger restriktiv zu gestalten, als dies bei der sehr konservativen GPL der Fall war. 1995 veröffentlichte die Firma Debian die *Debian Free Software Guidelines*, die 1997 für die Formulierung der *Open Source Definition* verwendet wurden (Open Source Initiative, 2015a). Die Debian Free Software Guidelines gaben den Lizenzgebern eine grössere Flexibilität bezüglich der Nutzung der Software. Es war erstmals erlaubt, frei erzeugten mit proprietärem Quellcode zu kombinieren, was unter der GPL klar verboten war.

Im Jahr darauf machten gleich zwei Open Source Anwendungen Schlagzeilen. Zum einen entschloss sich das Unternehmen Netscape dazu, die fünfte Version des Webbrowsers *Netscape Communicator* unter dem Namen „Mozilla“ als Open Source Browser freizugeben (Moody, 2001, S. 195), zum anderen ersetzte IBM seine eigene Webserver-Software durch die Open Source Alternative *Apache* (Moody, 2001, S. 206). Allgemein hatte OSS das Interesse von Unternehmen zu diesem Zeitpunkt endgültig geweckt. So entschied sich IBM dazu, alleine im Jahr 2001 eine Milliarde US-Dollar in das Linux-Projekt zu investieren, um die Entwicklung eigener Produkte voranzutreiben (Lindner, 2013). Ein weiteres Beispiel eines Unternehmens, das OSS

³ Im Zusammenhang mit Open Source ist der Begriff „free“ nicht mit gratis gleichzusetzen, sondern wird mit der Freiheit, die der Benutzer mit der Verwendung der Software erlangt, assoziiert (siehe Kapitel 4).

⁴ Auf die Eigenschaften der GPL wird in Kapitel 4.2.1 eingegangen.

offiziell unterstützt, ist die US-amerikanische Detailhändlerkette *Walmart*, die sich 2014 dem Hapi-Projekt⁵ anschloss (Phipps, 2014).

Die Entwicklung von OSS wurde nicht nur von Universitäten und aus der Privatwirtschaft gefördert. Das bessere Abschneiden bezüglich Sicherheit weckte auch das Interesse staatlicher Betriebe. 2001 präsentierte beispielsweise die National Security Agency (NSA) der USA ein eigenes Betriebssystem, das auf einem Linux System basiert (National Security Agency, 2015). Auch die Entscheidung des deutschen Bundestages vom März 2002, auf rund 150 Servern Microsoft Windows durch Linux zu ersetzen, zeigt, dass auch Staaten und Regierungen an der Weiterentwicklung von OSS interessiert sind (Heise Medien GmbH, 2002).

Der Erfolg und die Qualität von Linux und anderer OSS erregten vor allem deshalb so viel Aufmerksamkeit, weil hinter der Entwicklung des Programmes meist kein einzelnes Unternehmen steht, sondern diese Software typischerweise von einer nur lose organisierten Gemeinschaft Freiwilliger entwickelt wird (Raymond, 2001, S. 29ff). Ein ähnliches Phänomen zeigte sich bei den beiden Online-Enzyklopädien *Wikipedia* und *Encarta*. Während sich Wikipedia, das von tausenden Freiwilligen entwickelt und gepflegt wird, durchgesetzt hat, wurde die Entwicklung des von Microsoft professionell geführten, proprietären Encartas 2009 eingestellt (Crocoll, 2011). Microsoft reagierte damit auf den Erfolg Wikipedias, mit dem Encarta nicht mithalten konnte (Murad, 2009).

OSS blickt auf ein bewegtes halbes Jahrhundert zurück. Erst durch die oben beschriebenen Ereignisse ist Open Source heute in einer Position, in der es mit proprietärer Software gleichgesetzt werden kann. Open Source hat sich von einem Hobby zu einem Beruf entwickelt, und wird als Alternative zur *klassischen* Softwareentwicklungsmethode⁶ betrachtet. Im nächsten Kapitel wird

⁵ Hapi (<http://hapijs.com>) ist eine internetbasierte Entwicklungsplattform, welche die Entwicklung von Open Source Anwendungen fördert.

⁶ Die klassische Softwareentwicklung wird im Rahmen dieser Arbeit jeweils mit einer Entwicklung gleichgesetzt, die innerhalb einer Unternehmung und ohne externe Einflüsse stattfindet (siehe Kapitel 5).

die Frage beantwortet, weshalb und wodurch Open Source private Entwickler und Unternehmen anlockt.

3 Motivation

Im folgenden Kapitel soll auf die Frage eingegangen werden, was Entwickler weltweit dazu bewegt, sich zu teilweise riesigen Communities zusammenschliessen und freiwillig Beiträge zur Entwicklung von Software zu leisten, die dann anderen nicht nur kostenlos zur Verfügung gestellt wird, sondern von diesen auch beliebig verändert und weitergegeben werden darf. Dieselbe Frage stellt sich auch bei Unternehmen, die trotz ständigem Konkurrenzkampf und Erfolgsabhängigkeit das Ergebnis ihrer Arbeit veröffentlichen und so auf sämtliche Erträge aus dem Verkauf von Lizenzen verzichten.

3.1 Individuen

Gemäss Osterloh, Rota und Kuster (2004) gibt es zwei alternative Erklärungen, warum Programmierer freiwillig zur Entwicklung eines öffentlichen Guts beitragen, obwohl sie meist keine direkte Bezahlung dafür erhalten. Sie können eigennützige, kalkulierende Individuen sein, die rational in ihre Reputation investieren (Lerner & Tirole, 2002) oder aus der Anpassung der Software an ihre eigenen Bedürfnisse einen Nutzen ziehen (von Hippel, 2001; zitiert in Osterloh, Rota & Kuster, 2004). Als zweite Erklärung nennen Osterloh, Rota und Kuster (2004) sowohl den Spass am Programmieren, als auch die Überzeugung, zu einer „guten Sache“ beizutragen (Kollock, 1999).

Diese beiden Erklärungen basieren auf zwei unterschiedlichen Arten von Motivation (Hars & Ou, 2002; Lakhani & Wolf, 2005; Roberts et al. 2006; Wu et al., 2007; zitiert in von Krogh et al. 2012). Programmierer, die der ersten Erklärung zugeordnet werden können, sind *extrinsisch* motiviert. Sie erbringen eine Leistung, weil sie dafür eine Belohnung wie beispielsweise Karrierevorteile erhalten, oder weil sie dadurch einen Nachteil wie individuell schlecht nutzbare Software umgehen können. Programmierer, auf welche die zweite Erklärung zutrifft, sind hingegen *intrinsisch* motiviert. Die Tätigkeit wird um ihrer selbst willen ausgeführt und hängt nicht von einer externen Belohnung oder Bestrafung ab (Osterloh, Rota, & Kuster, 2004, S. 4).

Von Krogh et al. greifen dieses Konzept in ihrer 2012 durchgeführten Studie auf und fügen die *extrinsisch internalisierte* Motivation hinzu. Sie analysierten 40 Studien auf die darin genannten Motivatoren, zu Open Source

beizutragen⁷ (von Krogh et al., 2012, S. 654). Dabei teilten sie die in den Einzelstudien erwähnten Motivatoren den drei Kategorien intrinsisch, extrinsisch und extrinsisch internalisiert zu. Im Folgenden werden diese kurz beschrieben.

3.1.1 Intrinsische Motivatoren

Wie in Abbildung 1 ersichtlich ist, werden in den berücksichtigten Studien die vier intrinsischen Motivatoren *Ideologie*, *Altruismus*, *Community* und *Spas* unterschieden, die zusammen in rund 35 Prozent der verwendeten Quellen behandelt wurden.

Die Open Source zugrundeliegende Ideologie und der Altruismus werden in rund 10 Prozent der untersuchten Studien genannt. Zu Ideologie gehören laut von Krogh et al. (2012, S. 679) der Wunsch nach freier Veränderbarkeit und Weiterverteilung von Software und die Überzeugung, OSS solle proprietäre Software ersetzen. Hemetsberger (2004; zitiert in von Krogh et al., 2012) berichtet, dass rund ein Fünftel der Programmierer Altruismus als wichtigen Beweggrund, zu Open Source beizutragen, nannten. Dabei spielt Altruismus für häufig beitragende Entwickler eine grössere Rolle als für Entwickler, die nur gelegentlich oder selten beitragen. Auch Hars und Ou (2002, S. 3) stellten fest, dass Altruismus ein wichtiger Motivator ist, sich für eine aktive Teilnahme in der Community zu entscheiden.

Auch die Identifikation mit der Community spielt eine wichtige, intrinsisch motivierende Rolle und wird in rund sechs Prozent der untersuchten Studien als Motivator genannt. Je stärker die Entwickler sich mit der Community und deren Mitgliedern identifizieren, desto mehr wollen sie den Mitgliedern dieser Community durch eigene Beiträge helfen. Grössere Verbundenheit mit der Community bewirkt auch, dass den Entwicklern ihre Tätigkeit mehr Spas macht (z.B. Kollock, 1999; Hars & Ou, 2002; Lakhani & von Hippel, 2003; Osterloh, Rota & Kuster, 2004; Henkel, 2008).

⁷ Die ausgewählten Studien können Anhang A.1 entnommen werden.

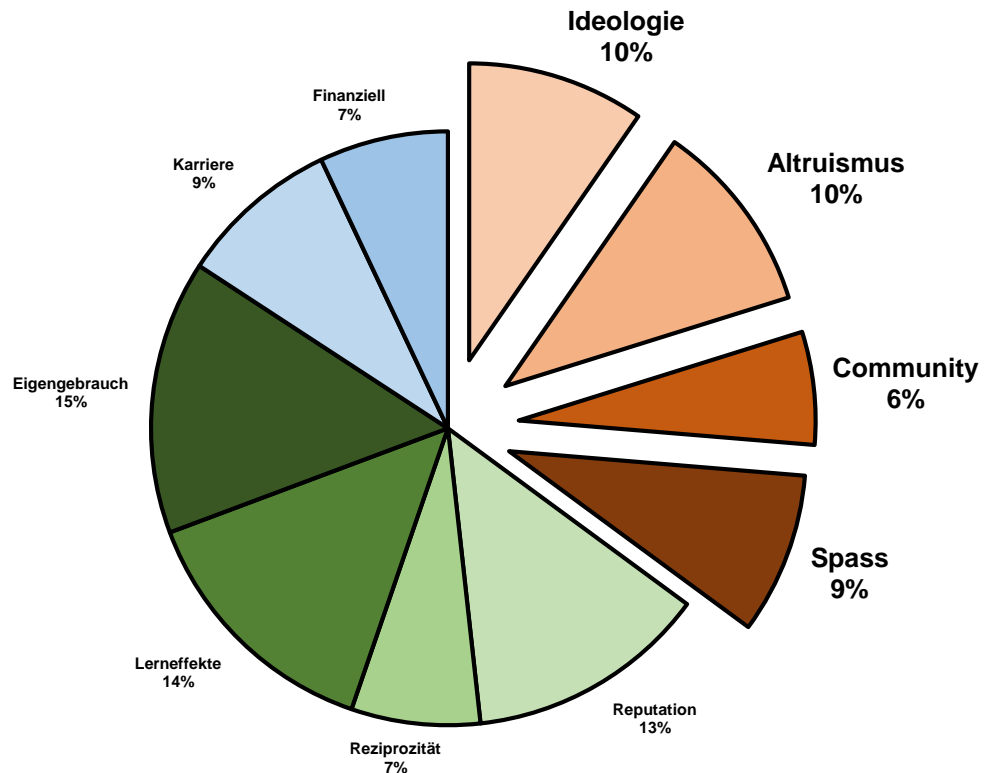


Abbildung 1 Intrinsische Motivatoren

In dieser Grafik wurden die intrinsischen Motivatoren (Rottöne) nach von Krogh et al. (2012) hervorgehoben.

3.1.2 Extrinsisch internalisierte Motivatoren

Bei der mit rund 50 Prozent grössten Motivationsart, der extrinsisch internalisierten Motivation, unterscheiden von Krogh et al. (2012) die Motivatoren *Eigengebrauch*, *Lerneffekte*, *Reziprozität* und *Reputation*, die in Abbildung 2 grün hervorgehoben sind.

Zum Einfluss, den die persönliche Reputation bei Arbeitskollegen oder Freunden und Verwandten hat, gehören die Wertschätzung geleisteter Entwicklungsarbeit und das Image, das Entwickler geniessen (z.B. Kollock, 1999; Hars & Ou, 2002; Lakhani & von Hippel, 2003; von Krogh et al., 2012). Die persönliche Reputation gilt als grundlegende Motivation, überhaupt zu Open Source beizutragen (z.B. Ghosh, 2005; Hars & Ou, 2002; Hemetsberger, 2004; Lakhani & Wolf, 2005; zitiert in von Krogh et al., 2012). Auf das Thema

der Reziprozität gehen unter Lakhani und von Hippel (2003) ein. Sie beschrieben die Kernaussage reziproken Verhaltens in ihrer Studie zur Apache Web Server Software wie folgt:

„I help today because I have been helped in the past and/or I expect to be helped in the future.“ (S. 924)

Während laut Raymond (1998) bei OSS das Prinzip von Geben und Nehmen auf dem Austausch von Ideen und Informationen beruht, liegt bei proprietärer Software eher eine klassische Transaktion vor, bei der ein Gut für einen bestimmten Geldbetrag gekauft wird. Im Anschluss erwartet der Käufer weitere Dienstleistungen wie Support und Updates, ohne selbst etwas zur Verbesserung und Weiterentwicklung beizutragen. Bei OSS hingegen motiviert der Erhalt von Beiträgen dazu, auch selbst zu programmieren.

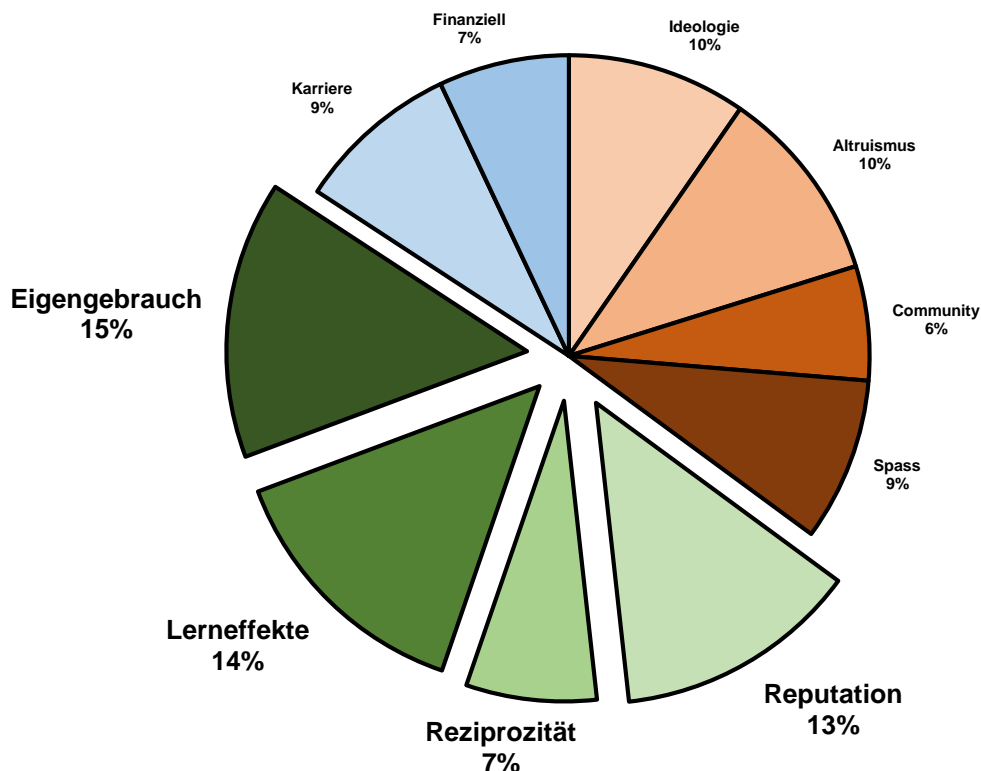


Abbildung 2 Extrinsisch, internalisierte Motivatoren

In dieser Grafik werden die durch von Krogh et al. (2012) beschriebenen extrinsisch internalisierten Motivatoren hervorgehoben (Grüntöne).

Den zweitgrössten Motivator stellen die Lerneffekte dar. Durch das Beitragen zu Projekten in der Open Source Community können Entwickler von anderen, erfahreneren Mitgliedern Neues lernen und somit ihre eigenen Programmierfähigkeiten verbessern.

Die grösste und wichtigste extrinsisch internalisierte Triebkraft ist gemäss von Krogh et al. (2012) der Eigengebrauch, respektive die Nachfrage nach einer Anwendung, die in der gewünschten Form noch nicht existiert (siehe auch Hars & Ou, 2002; Lakhani & von Hippel, 2003).

3.1.3 Extrinsische Motivatoren

Den mit rund 16 Prozent geringsten Anteil der bei von Krogh et al. (2012) genannten Motivatoren machen die rein extrinsischen Faktoren aus. Dabei wird zwischen *finanziellen* und *karrierebezogenen* Motivatoren unterschieden.

Heutzutage wird viel Geld in OSS und deren Entwicklung investiert, was sich auch anhand der Investitionen von Grosskonzernen wie IBM zeigt (Henkel, 2008, S. 60). Werden Programmierer für ihre Arbeit bezahlt, gilt dies als extrinsischer Motivator. Nebst direkter Entlohnung werden Entwickler oft auch indirekt unterstützt, indem es ihnen z.B. gestattet ist, in ihrer Arbeitszeit für Open Source Projekte zu arbeiten (Lakhani & Wolf, 2005; zitiert in von Krogh et al., 2012).

Die Entwicklung von Karrieremöglichkeiten ist stark von anderen Faktoren abhängig. Die Mitarbeit an Open Source Projekten erhöht beispielsweise die eigenen Fähigkeiten, wodurch die Chancen auf dem Stellenmarkt verbessert werden. Durch das Veröffentlichen frei zugänglicher Software können OSS-Entwickler ihr Talent und Potential zeigen und so ihren Wert auf dem Arbeitsmarkt erhöhen (Lerner & Tirole, 2002).

Der in Abbildung 3 erkennbare, geringe Anteil der extrinsischen Motivatoren widerspiegelt die Dynamik, die hinter Open Source steht. Entwickler sind aufgrund der Ideologie von Open Source entweder intrinsisch motiviert oder internalisieren extrinsische Motivatoren. Rein extrinsische Motivatoren können keine entscheidende Rolle spielen, wenn Beiträge in erster Linie erzeugt werden, um eigene Bedürfnisse zu decken oder anderen zu helfen.

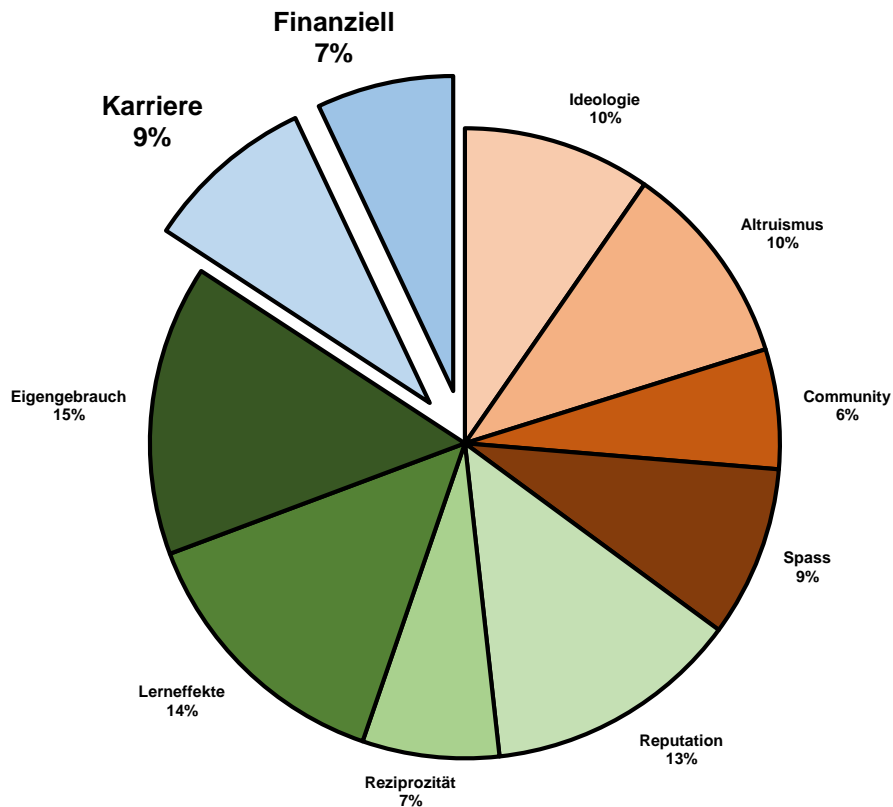


Abbildung 3 Extrinsische Motivatoren

Von Krogh et al. (2012) beschreiben zwei verschiedene extrinsische Motivatoren, die in dieser Grafik blau hervorgehoben sind.

3.2 Unternehmen

Ähnlich wie bei Einzelpersonen verhält es sich auch bei Firmen: es stellt sich die Frage, weshalb Unternehmen viel Zeit, Geld und Aufwand in etwas stecken sollten, das keinen direkten Gewinn bringt. Bill Gates (1976) ging in *An Open Letter to Hobbyists* auf dieses Problem ein, indem er sich fragte, wer es sich überhaupt leisten könne, jahrelange Arbeit in ein Projekt zu investieren, um es anschliessend gratis weiterzugeben. Dieselbe Frage ist auch heute noch aktuell. So fragten sich beispielsweise Lerner und Tirole (2001), wieso Unternehmen teilweise ihre besten Softwareentwickler einem Open Source Projekt zu teilten oder selbst Projekte lancieren, anstatt die gewünschte Software durch externe Unternehmen entwickeln zu lassen.

Einer nach wie vor grossen Anzahl Anbieter proprietärer Software steht mittlerweile dennoch eine stark und schnell wachsende Open Source Gesellschaft gegenüber. Da es unter der Verwendung von Open Source Lizenzen nicht möglich ist, Lizenzgebühren zu verlangen, kam es in den letzten zehn Jahren zu einem Wandel von Ertragsmodellen basierend auf Lizenzverträgen hin zu einem Modell, in dem Erträge vermehrt auf erbrachten Dienstleistungen basieren. Laut Lerner und Tirole (2002, S. 197) wurde die Entwicklung von OSS durch drei Faktoren begünstigt:

Die schnelle Verbreitung der Software. Als Lerner und Tirole 2002 ihren Artikel veröffentlichten, verzeichneten top Open Source Produkte wie Linux jährliche Wachstumsraten von bis zu 200 Prozent. Im Jahr 2012 betrug das Wachstum in den ersten neun Monaten immerhin noch 64 Prozent (Pingdom Royal, 2012). Im Vergleich dazu konnte Marktführer Microsoft im Jahr 2012 lediglich ein Wachstum von sechs Prozent verzeichnen (Gartner, 2014). Im Vergleich mit den Wachstumsraten proprietärer Software wächst der Markt für OSS deutlich schneller.

Die signifikanten Investitionen in Open Source Projekte. Seit längerem investieren namhafte Unternehmen wie IBM, Hewlett Packard oder Sun Microsystems (heute Oracle) regelmässig grosse Summen in externe und eigene Open Source Projekte. Andere Unternehmen wie Red Hat oder Sendmail bauten Geschäftsmodelle rund um OSS auf oder fanden Investoren zur Unterstützung ihrer Projekte.

Die neue organisatorische Struktur. Durch die Open Source Bewegung wurde die kollaborative Softwareentwicklung erst richtig bekannt. Auf diese Art der Softwareentwicklung und deren Vorzüge wird in Kapitel 5 genauer eingegangen.

Es gibt zwei Gründe, weshalb die steigenden Investitionen in Open Source aus ökonomischer Sicht nicht nachvollziehbar sind: Erstens ist es unter Verwendung von Open Source Lizenzen nicht möglich Lizenzgebühren zu verlangen, weshalb ein darauf aufbauendes Ertragsmodell nicht anwendbar ist. Zweitens sind mögliche Wettbewerbsvorteile gegenüber Konkurrenten bedroht, da die Konkurrenz, den offengelegten Quellcode nicht nur zu betrachten, sondern diesen auch in eigenen Entwicklungen weiterverwenden kann. Wieso es dennoch Sinn macht, eine Umstellung auf Open Source in Erwägung

zu ziehen, wird im Folgenden mittels einer Auflistung der wichtigsten Motive dargelegt.

Als Basis für die Gegenüberstellung der Vor- und Nachteile einer Investition in OSS in Tabelle 1 dienen die Befunde Henkels (2008, S. 83), die durch eigene Punkte ergänzt werden.

Vorteile	Nachteile
<ul style="list-style-type: none"> ▪ Entwicklungsunterstützung (Wissenstransfer) ▪ Kosteneinsparungen ▪ Einfachere Standardsetzung ▪ Erhöhte Mitarbeiterdisziplin und -motivation ▪ Erhöhte Attraktivität als Arbeitgeber ▪ Vergrößerung des Pools an potentiellen Mitarbeitern ▪ Reduzierte Abhängigkeit von proprietärer Software ▪ Druck auf Anbieter proprietärer Substitute ▪ Bessere Kontrolle über Software und Daten 	<ul style="list-style-type: none"> ▪ Risiko des Forkings ▪ Verlust von Wettbewerbsvorteilen ▪ Einsicht für Wettbewerber in Geschäftsprozesse ▪ Höhere Wartungskosten bei Open Source Projekts ▪ Sichtbarkeit von Sicherheitslücken ▪ Erträge durch Verkauf von Lizenzen ausgeschlossen ▪ Verlust von Bündelungsvorteilen ▪ Support möglicherweise nicht gesichert ▪ Hohe Trainingskosten für Personal

Tabelle 1 Vor- und Nachteile einer OSS-Migration für Unternehmen

Ein wichtiger Vorteil von Open Source ist, dass ein Unternehmen durch die Verwendung von OSS Zugang zu einer grossen Entwicklercommunity erlangt. Dabei kann die Unternehmung vom Wissen der Mitglieder profitieren. Zusätzlich können Unterhaltskosten gespart werden, da ein Teil der anfallenden Aufgaben auf ebendiese, grösstenteils kostenlos arbeitende Entwicklercommunity übertragen werden kann.

Bedingt durch die Offenheit des Quellcodes können Änderungen und Verbesserungen jederzeit sowohl von Vertretern der Unternehmung, als auch von der Community überprüft und überwacht werden. Dies sollte interne Programmierer dazu animieren, Quellcode von möglichst hoher Qualität zu generieren und erhöht nebenbei auch die Mitarbeiterdisziplin. Die dadurch verbesserte Produktequalität lockt mehr und grössere Kunden an, was wiederum zu

mehr Umsatz führt. Mit den zusätzlichen Mitteln können bessere Mitarbeiter eingestellt werden, was die Qualität der Produkte weiter erhöht und somit auch die Attraktivität als Arbeitgeber im Open Source Markt steigert.

Durch ein Umstieg auf OSS ist es für Unternehmen auch einfacher, die eigenen Produkte zu Branchenstandards zu machen. Durch die Veröffentlichung des Quellcodes ist eine Software für alle kostenlos und ohne Einschränkungen verfügbar. Dadurch erhöht sich die Chance, dass die veröffentlichte, freie Software künftig auch von anderen Entwicklern in deren eigenen Projekten eingebaut wird. Eine solche Integrierung der ursprünglichen Software in andere Anwendungen erhöht die Wahrscheinlichkeit, dass sich die anfängliche Software zu einem Branchenstandard etabliert.

Ebenfalls nicht zu unterschätzen ist die gewonnene Unabhängigkeit von proprietären Softwareanbietern, die mit einem Wechsel auf Open Source einhergeht. Unabhängiger zu sein, bedeutet für ein Unternehmen nicht nur, eine grössere Auswahl potentieller Vertragspartner, sondern auch eine erhöhte Verhandlungsstärke. Dadurch steigt die Aussicht auf Kosteneinsparungen (Open Source Studie Schweiz, 2012, S. 9). Eine grosse Anzahl an Dienstleistern erlaubt es einem Unternehmen den Vertragspartner zu wechseln, wenn die Verhandlungen nicht zufriedenstellend verlaufen. Um keine Kunden zu verlieren, müssen Dienstleister ihre Preise senken und Konditionen verbessern.

Durch die Offenheit des Quellcodes können sowohl Produzenten, als auch Nutzer einer Software jederzeit nachvollziehen, welche Prozesse wann ablaufen. Dies gibt ihnen die volle Kontrolle über sämtliche Daten, die durch den Gebrauch generiert oder verändert werden.

Eine Umstellung auf OSS kann auch einige Nachteile mit sich bringen. So kann freigegebener Quellcode von Dritten verwendet werden, um unabhängige Produkte zu entwickeln. Dies wird *Forking*⁸ genannt. Kritischer ist, dass der freigegebene Quellcode auch von Konkurrenten eingesehen werden kann. Mit den gewonnenen Informationen könnte ein Wettbewerbsnachteil

⁸ Bei Forking wird bestehender Open Source Quellcode verwendet, um eine komplett neue und vom Ursprungsprodukte unabhängige Anwendung zu entwickeln (WhatIs.com, 2015).

aufgeholt oder gar der Wettbewerbsvorteil des Entwicklerunternehmens übernommen werden.

Ein weiterer Nachteil ist die Schwierigkeit, geeignete Supportpartner zu finden, welche die gewünschten Dienstleistungen in der vom Unternehmen oder dessen Kunden geforderten Qualität anbieten. Teilweise existieren gar keine Anbieter von Supportdienstleistungen, da die entsprechende Software zu komplex oder noch nicht lange genug auf dem Markt ist. Dies ist insbesondere bei spezifizierten oder personalisierten Anwendungen der Fall.

Je nach Grösse des Unternehmens können kurzfristige Kosteneinsparungen durch anfallende Umschulungskosten aufgewogen werden, weshalb das Unternehmen längerfristig, nicht aber unmittelbar profitiert. Viele Unternehmen können sich den kurzfristigen Mehraufwand nicht leisten oder müssen ihre Strategie anpassen. Dies ist wiederum mit zusätzlichem Aufwand oder Mehrkosten verbunden und hält deshalb von einer Umstellung auf OSS ab.

In der 2015 durchgeführten Open Source Studie werden zusätzlich fehlende Schnittstellen oder eine einengende Abhängigkeit von proprietären Systemen als Nachteile von OSS genannt (Open Source Studie Schweiz, 2015, S. 7). Das Vorliegen eines Vendor Lock-Ins⁹ ist einer der Hauptgründe, wieso viele Unternehmen nicht auf OSS wechseln.

Trotz den signifikanten Nachteilen gibt es Gründe, weshalb sich eine Umstellung dennoch lohnt. So listete Volpi (2014) sechs Gründe auf, die für eine Investition in Open Source Projekte sprechen.

1. Open Source ist topaktuell

Früher wurden Open Source Produkte lediglich als günstige, aber qualitativ minderwertige Alternativen zu proprietären Produkten angesehen. Heute werden Produkte wie Linux oder MySQL zur Entwicklung neuer Anwendungen, auf denen diverse Start-Ups ihre Geschäftsidee basieren, verwendet. Open Source gilt als eigenständige und gleichwertige Softwareentwicklungsmethode.

⁹ Unter Vendor Lock-In wird eine Situation verstanden, in der ein Kunde aufgrund bestehender Verträge, usw. nicht ohne weiteres zu einem Produkt oder einer Dienstleistung eines Konkurrenten des aktuellen Anbieters wechseln kann (WhatIs.com, 2015).

2. Open Source hat Unternehmensniveau erreicht

Lange Zeit hatte Open Source den Ruf, nicht unternehmenstauglich zu sein. Oftmals gab es entweder nicht genügend Möglichkeiten, die Software an firmenspezifische Bedürfnisse anzupassen, oder die OSS war nicht genügend sicher gegen Angriffe von aussen. Firmen gaben deshalb Millionen für den Erwerb und die Verlängerung proprietärer Softwarelizenzen aus. Heutige Unternehmen sind nicht mehr gewillt, diese Beträge für Produkte, die sie günstiger und besser erwerben können, aufzubringen. Positiver Nebeneffekt ist die reduzierte Abhängigkeit von wenigen proprietären Grossanbietern. Aktuelle Open Source Anwendungen sind vom Funktionsumfang her oft sogar besser als proprietäre Software und bieten gut bedienbare Benutzeroberflächen. OSS ist also konkurrenzfähig geworden und gilt heute als ernstzunehmende Alternative mit vielen Vorzügen.

3. Profitable Open Source Geschäftsmodelle funktionieren

Die cloudbasierte Verteilung von Software hat es Open Source Geschäftsmodellen ermöglicht, rentabel zu sein. Die heutzutage oft von Softwareunternehmen und App-Entwicklern verwendeten *Freemium-Modelle* stammen ursprünglich aus der Open Source Community. Dabei wird die Software an sich meist kostenlos zur Verfügung gestellt und kann anschliessend durch kostenpflichtige weitere Funktionen oder Dienstleistungen ergänzt werden.

4. Systeme werden komplexer

Bei den anfänglichen Open Source Technologien handelte es sich grösstenteils um Betriebs- oder Datenbankmanagementsysteme, die auf einem einzelnen Server installiert und anschliessend lokal betrieben wurden (Beispiel MySQL). Die Verbesserung der Infrastruktur und der Technologie zwangen auch Open Source Anwendungen dazu, sich weiterzuentwickeln. Gleichzeitig öffneten sich ihnen neue Möglichkeiten. So wäre es z.B. noch vor zehn Jahren undenkbar gewesen, ohne

Breitbandinternet und Cloud-Computing rund 40'000 Server mittels Hadoop¹⁰ zu managen, wie dies bei Yahoo gemacht wird.

5. Open Source Start-Ups kontrollieren ihr Schicksal

Erste Open Source Unternehmen wie beispielsweise Red Hat verliesen sich bei der Softwareentwicklung vollständig darauf, dass diese Entwicklungsarbeit durch die Community gemacht wurde. Dementsprechend hoch war das Risiko, nicht das gewünschte Produkt zu erhalten. Neuere Open Source Unternehmen wie Pentaho oder HortonWorks entwickeln ihre Anwendungen grösstenteils intern¹¹, was ihnen eine grössere Kontrolle über Produktefeatures und den Entwicklungsprozess gibt. Dadurch können Unternehmen nicht nur einen personalisierten, qualitativ hochstehenden Support für ihre Kunden garantieren, sondern auch ihre Abhängigkeit von der Verfügbarkeit der Entwicklercommunity minimieren.

6. Die Umstellung auf Open Source hat eine kritische Masse erreicht

Begünstigt durch die ersten fünf hier genannten Gründe hat Open Source einen deutlich erhöhten Verwendungsgrad erreicht. In nicht-technischen Industriezweigen verwenden ca. 86 Prozent aller Unternehmen Open Source (Vaughan-Nichols, 2013; zitiert in Volpi, 2014). Die entsprechende Zahl ist bei Unternehmen aus Technikbranchen wahrscheinlich noch grösser. Weiter wurde die Open Source Entwicklung durch die Erfindung der Cloud beschleunigt. Probleme konnten mittels Cloud der Community zugänglich gemacht werden, was die zur Fehlerbehebung benötigte Zeit stark verkürzte.

Open Source ist heute nicht mehr eine qualitativ minderwertige Alternative zu proprietärer Software, die lediglich von ein paar Freiwilligen entwickelt und verwendet wird. Die Open Source Community wächst rasant an, die Soft-

¹⁰ Hadoop ist ein Open Source Framework für zuverlässiges, skalierbares und verteiltes Rechnen (reliable, scalable, distributed computing). Die umfangreichste Hadoop Installation betreibt Facebook, mit insgesamt 2'000 Rechnern und 21 PB Daten (Thomas Krenn, 2015).

¹¹ Ca. 80 Prozent des gesamten Aufwandes fallen intern an.

ware ist geschäftstauglich geworden, hat sich als eigenständige Softwareentwicklungsmethode etabliert, ist personalisierbar und individualisierbar und gilt als sehr sicher. Probleme wie die Inkompatibilität mit traditionellen Ertragsmodellen wurden durch einen langsamen Wandel zu auf Dienstleistungen beruhenden Modellen gelöst. Open Source ist als mächtige Bewegung mit grossem Potential wahrzunehmen, bei der es sich auch für Unternehmen lohnen kann, finanzielle und personelle Ressourcen einzusetzen.

Im folgenden Kapitel werden die verschiedenen Lizenzmodelle, die Open Source bietet, vorgestellt. Ein spezieller Fokus wird dabei auf die Rechte und Pflichten des Nutzers einer Lizenz gelegt, da deswegen oftmals eine grosse Unsicherheit vorliegt (BITKOM, 2015, S. 5).

4 Lizenzierung

In diesem Kapitel wird darauf eingegangen, welche Bedingungen eine Softwarelizenz erfüllen muss, um als Open Source Lizenz zu gelten. Dazu wird weiter werden rechtliche Aspekte betrachtet, die der Einsatz der entsprechenden Lizenz mit sich bringt. Zuvor wird die offizielle Definition der Open Source Initiative beschrieben, sowie ein kurzer Überblick über die aktuelle Lizenzsituation im OSS-Markt gegeben.

4.1 Definition der Open Source Initiative

Damit eine Software als Open Source veröffentlicht werden kann, muss dies unter Verwendung einer entsprechenden Lizenz erfolgen. Die Bezeichnung *quelloffen* (open source) bedeutet nicht nur, dass der Zugang zum Quellcode für alle frei sein muss, es werden zusätzlich an die Weitergabe der Software gewisse Bedingungen geknüpft. Die Open Source Initiative führt auf ihrer Homepage folgende Kriterien auf, die eine Lizenz erfüllen muss, um als Open Source Lizenz verwendet werden zu können (Open Source Initiative, 2015a):

1. *Freie Weitergabe.* Die Software darf uneingeschränkt und ohne zusätzliche Lizenzgebühren weitergegeben werden.
2. *Quellcode.* Das Programm muss den Quellcode beinhalten, oder dieser muss gratis, beziehungsweise zu Selbstkosten verfügbar gemacht werden.
3. *Abgeleitete Software.* Die Lizenz muss von der Basissoftware abgeleitete Arbeiten und deren Distributionen unter derselben Lizenz erlauben.
4. *Integrität des Quellcodes des Autors.* Die Lizenz darf die Möglichkeit, den Quellcode in veränderter Form weiterzugeben, nur dann einschränken, wenn sie vorsieht, dass zusammen mit dem Quellcode sogenannte *Patch Files*¹² weitergegeben werden dürfen, die den Programmquellcode bei der Kompilierung verändern.

¹² Unter einem *Patch* (wörtlich: Flicker) versteht man eine Softwarekomponente zur Korrektur einer oder mehrerer Fehlfunktionen bei einem bereits installierten Softwareprodukt (Vogel, 2015). Ein *Patch File* ist demnach die Datei, die effektiv versendet wird.

5. *Keine Diskriminierung von Personen oder Gruppen.* Die Lizenz darf keine Personen oder Gruppen aufgrund deren Ethik, Nationalität, etc. benachteiligen.
6. *Keine Einschränkung bezüglich des Einsatzfeldes.* Die Lizenz darf den Verwendungszweck der Software nicht einschränken.
7. *Weitergabe der Lizenz.* Die Rechte an einem Programm müssen auf alle Personen übergehen, die diese Software erhalten.
8. *Die Lizenz darf nicht auf ein bestimmtes Produkt beschränkt sein.* Die Rechte an einem Programm dürfen nicht davon abhängig sein, ob dieses Teil eines bestimmten Softwarepaketes ist. Wird Software getrennt von diesem Softwarepaket weitergegeben, haben die Empfänger dieselben Rechte, wie die Nutzer des ursprünglichen Softwarepakets.
9. *Die Lizenz darf die Weitergabe zusammen mit anderer Software nicht einschränken.* Die Lizenz darf z.B. nicht verlangen, dass sämtliche verwendeten Softwarekomponenten OSS sein müssen.
10. *Die Lizenz muss technologieneutral sein.* Sie darf beispielsweise den Absatzkanal nicht einschränken, indem sie verlangt, dass nur CDs oder DVDs verwendet werden dürfen.

4.2 Lizenzarten

Um einen Überblick über die gängigsten Open Source Lizenzen zu geben, werden in diesem Kapitel die drei zuvor kurz erwähnten Kategorien vorgestellt, die Rechte und Pflichten der Nutzer beschrieben, sowie auf ausgewählte Lizenzen genauer eingegangen. Als Grundlage dient dabei der Leitfaden des BITKOMs (2015, S. 9ff).

4.2.1 Sehr restriktive Lizenzen

Die Kategorie der sehr restriktiven Lizenzen, auch Copyleft-Lizenzen genannt, wird am Beispiel der GPL vorgestellt. Obwohl die GPL zu den restriktivsten Open Source Lizenzen gehört, die aktuell verwendet wird, ist sie dennoch die am meisten verwendete Lizenz¹³.

¹³ Siehe Abbildung 4

Die GPL verpflichtet den Nutzer dazu, die Weiterentwicklung einer darunter veröffentlichten Software den für die Originallizenz geltenden Bestimmungen zu unterstellen, wenn für diese Weiterentwicklung der Quellcode verändert wurde. Diese Pflicht verleiht ihr einen viralen Charakter, da sämtlicher mit der Lizenz in Berührung gekommener Quellcode dieser auch automatisch unterstellt wird. Es ist also nicht möglich, Teile einer unter der GPL stehenden Software abzuspalten, unter eine proprietäre Lizenz zu stellen und dann zu verkaufen. Die proprietäre Software wird bei einer Verknüpfung mit unter der GPL stehenden Software zu OSS. Den Effekt der entsprechenden Lizenzbedingung nennt man *Copyleft*¹⁴.

Die GPL verlangt aber keineswegs, dass ihr jede Veränderung von Software unterstellt werden muss. Sie erlaubt sogar die Verbreitung freier Software zusammen mit proprietären Lizenzbedingungen. Wichtig ist jedoch, dass freie und proprietäre Software klar voneinander getrennt weitergegeben werden, kein abgeleitetes Werk entsteht und die unter der GPL stehende Software nicht verkauft oder gegen Vergütung lizenziert wird.

Der Nutzer ist bei sehr restriktiven Lizenzen berechtigt

- zur uneingeschränkten Weiterverbreitung der Software ohne Erhebung von Lizenzgebühren
- zur Nutzung der Software für jegliche Zwecke
- zur Vervielfältigung
- zur Bearbeitung
- zur Weitergabe unveränderter und veränderter Versionen der Software

wenn dabei folgende Pflichten eingehalten werden:

Bei unveränderter Programmversion

- muss der vollständige Lizenztext der GPL beigefügt werden
- muss ein *Copyrightvermerk* beigefügt und es dürfen keine Urhebervermerke und Gewährleistungsausschlüsse der GPL entfernt werden

¹⁴ Der Begriff Copyleft wurde erstmals im GNU-Projekt verwendet und dadurch auch geprägt (Free Software Foundation, 2015b).

- muss ein Hinweis auf den Haftungs- und Gewährleistungsausschluss der GPL beigefügt werden
- muss der Quellcode mitgeliefert werden
- dürfen die Pflichten aus der GPL nicht erweitert werden

Bei veränderter Programmversion

- muss das Copyleft beachtet werden
- muss ein Hinweis auf Modifikation und deren Datum eingefügt werden

4.2.2 Restriktive Lizenzen

Restriktive Lizenzen haben einen beschränkten Copyleft-Effekt. Sie werden am Beispiel der *Lesser General Public License* (LGPL) vorgestellt.

Die LGPL schreibt vor, dass Weiterentwicklungen der Software wie bei der GPL grundsätzlich den ursprünglichen Lizenzbedingungen unterstellt werden müssen (Copyleft). Werden Änderungen des Quellcodes jedoch in eigenständigen Dateien gemacht, können diese auch unter anderen, proprietären Lizenzbedingungen weitergegeben werden. Eine Kombination von Software unter mehreren Lizenzarten ist somit möglich, wenn auch nicht uneingeschränkt. Die LGPL hiess ursprünglich *GNU Library General Public License*, weshalb es naheliegt, ihre Funktionsweise am Beispiel einer Bibliothek zu beschreiben. Um zu bestimmen, unter welcher Lizenz eine Software gestellt wird, muss unterschieden werden, ob eine externe Software lediglich auf die unter der LGPL stehende Bibliothek zugreift oder ob sie mit der Bibliothek verlinkt ist. Greift die externe Software lediglich auf die Bibliothek zu, um beispielsweise Daten abzufragen, kann die externe Software ihre Lizenz behalten, da sie dabei unabhängig von der Bibliothekssoftware betrieben werden kann. Falls eine unabhängige Zusammenarbeit nicht möglich und die externe Software mit der Bibliothek verlinkt ist, tritt der Copyleft-Effekt ein und alle involvierten Softwarekomponenten fallen unter die Lizenzbestimmungen der Bibliothek. Wie dieses Beispiel zeigt, ist eine Softwarekombination aus verschiedenen Lizenzarten möglich, solange die beiden Teile unabhängig voneinander sind.

Bei restriktiven Lizenzarten ist der Nutzer berechtigt

- zur uneingeschränkten Weiterverbreitung der Software ohne Erhebung von Lizenzgebühren
- zur Nutzung für jegliche Zwecke
- zur Vervielfältigung
- zur Bearbeitung
- zur Weitergabe unveränderter und veränderter Versionen der Software

Grundsätzlich sind die Anforderungen geringer als bei der GPL. Folgende Mindestanforderungen oder Pflichten müssen aber zwingend erfüllt werden:

- Weitergabe der Lizenz
- Beibehalten des Copyrightvermerks
- Hinweis auf den Haftungsausschluss

4.2.3 Nicht-restriktive Lizenzen

Nicht restriktive Lizenzen haben keinen Copyleft-Effekt. Zu dieser Kategorie gehört z.B. die in Abbildung 4 genannte BSD-Lizenz.

Lizenzen ohne Copyleft-Effekt enthalten weniger Pflichten als die zuvor vorgestellten Kategorien. Insbesondere können Veränderungen der Software vorgenommen und dann unter beliebigen, auch eigenen, Lizenzbedingungen verbreitet werden. Dies ermöglicht die Kombination mit proprietärer Software, da keine Einschränkungen bezüglich der verwendeten Lizenz gemacht werden.

Nutzer erhalten ein unbeschränktes Nutzungsrecht der Software. Dies bedeutet, dass sie das Recht zur Vervielfältigung, Veränderung und zum Vertrieb veränderter und unveränderter Versionen als Quellcode oder im Objektcode erhält. Auch die Vermarktung von Software unter eigenen, auch proprietären Lizenzen ohne Offenlegung des Quellcodes ist möglich.

Die Mindestpflichten bei nicht-restriktiven Lizenzen sind

- die Weitergabe der OSS-Lizenz
- das Beibehalten des Copyright-Vermerks
- Hinweise auf den Haftungsausschluss (*Software as is*)
- das Beachten einer Werbeklausel

Diese Pflichten gelten nur für die Weitergabe von freigegebenem Quellcode. Wird Quellcode unter nicht-restriktiver Lizenzen laufender Software modifiziert und proprietär vermarktet, entfallen diese Pflichten.

4.3 Aktuelle Situation

Die Auswahl an Open Source Lizenzen ist gross. Alleine auf der Webseite der Open Source Initiative werden mehr als 70 von ihnen anerkannte Lizenzen angegeben (Open Source Initiative, 2015b). Gemäss Software Package Data Exchange (2015) gibt es aktuell 294 verschiedene Lizenzen, von denen sich viele aber nur geringfügig unterscheiden. Gemäss Lerner und Tirole (2005, S. 22) lassen sich OSS Lizenzen in drei Gruppen einteilen: *sehr restriktive* (mit Copyleft), *restriktive* (mit beschränktem Copyleft) und *nicht restriktive* Lizenzen (ohne Copyleft).

Eine sehr restriktive Lizenz stellt hohe Anforderungen an die Rechte und Pflichten der Benutzer betreffend der Weitergabe und Verwendung der Software¹⁵. Restriktive Lizenzen sind aber trotz strenger Bestimmungen besser geeignet, um neue Entwickler anzulocken, da sie sicherstellen, dass veröffentlichte Beiträge immer einem bestimmten Entwickler zugeordnet werden können. In der Softwareentwicklung werden auch weniger restriktive Lizenzen eingesetzt, da diese deutlich mehr Spielraum für Ideen lassen und somit die Motivation für Softwareentwickler, zu einem Projekt beizutragen, steigern (Lerner & Tirole, 2005b, S. 39). Privatpersonen → restriktive Lizenzen, Unternehmen → weniger restriktive Lizenzen → → Ausführen, wieso es sich nicht widerspricht

Abbildung 4 zeigt eine Rangliste der meistgebrauchten Lizenztypen (Black Duck Software Inc., 2013). Die Liste enthält Lizenzen aus allen drei in diesem Kapitel beschriebenen Restriktionsstufen.

¹⁵ Siehe Definition unter 4.2.1.

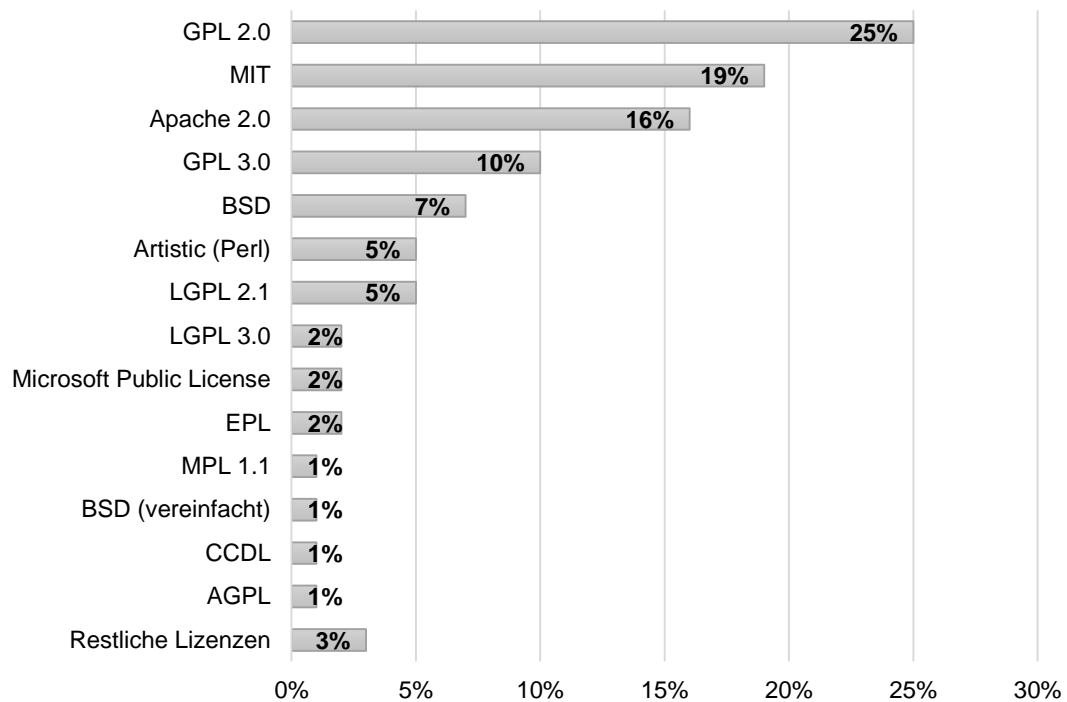


Abbildung 4 Meistverwendete Open Source Lizenzen

Auflistung der Firma Black Duck Software Inc. über die 15 meistverwendeten Open Source Lizenzen aus dem Jahr 2013

Laut den Befunden in Abbildung 4 wurden 2013 grösstenteils sehr restriktive (z.B. GPL, LGPL, MPL¹⁶) oder nicht restriktive (z.B. MIT, Apache, BSD, Artistic) Open Source Lizenzen eingesetzt. Die in Kapitel 2 beschriebene Verschiebung der Kräfteverhältnisse in den 1980er und 1990er Jahren ist auch in der obigen Abbildung erkennbar. Am Anfang waren die GPL oder ähnliche, restriktive Lizenzen dominant. Später veränderte sich die Open Source Branche zu einem Markt, in der neben diesen restriktiven Lizenzen eine in etwa gleichbedeutende Gruppe weniger restriktiven Lizenzen existierte. Diese beiden Lizenzgruppen werden, wie Abbildung 4 zeigt, etwa gleich häufig verwendet. Eine Einzelbetrachtung der Lizenzen zeigt, dass die GPL nach wie vor die meistverwendete Lizenz ist. Dies entspricht den Feststellungen aus Kapitel 2.

Durch die Forderung nach weniger restriktiven Lizenzen in den 1990er-Jahren und der daraus resultierenden Abschwächung der Bestimmungen konnten neue Anreize gesetzt werden, um neue Lizenzen zu generieren. Aus

¹⁶ MPL steht für *Mozilla Public License*

diesem Grund entstanden diverse Lizenzen wie die MIT- oder BSD-Lizenz, die in den folgenden Jahren einen erheblichen Marktanteil erlangten. Sie schafften es jedoch nicht, die GPL als meistverwendete Lizenz zu verdrängen, da der Fokus nach wie vor stark auf der Entwicklung und Weiterentwicklung neuer und bestehender OSS lag. OSS-Anbieter erhielten durch die neuen, weniger restriktiven Bestimmungen aber die Möglichkeit, Geschäftsmodelle zu entwickeln, mit denen Gewinne erwirtschaftet werden konnten.

Gemäss Abbildung 4 werden Lizenzen wie die *Microsoft Public License*, die *Eclipse Public License* (EPL) oder die MPL weniger häufig verwendet. Dies könnte durch deren Spezifität erklärt werden. Sie wurden jeweils für ein sehr spezifisches Produkt wie den Mozilla Firefox Internetbrowser, geschaffen und können auch nur dort eingesetzt werden. Die GPL oder die MIT-Lizenz hingegen können überall eingesetzt werden und werden wahrscheinlich deshalb öfters verwendet.

5 Open Source Softwareentwicklung

Die *Open Source Softwareentwicklung* (OSSE) hat sich in den letzten Jahren zu einer ernstzunehmenden Alternative zur klassischen Softwareentwicklung etabliert. Im folgenden Kapitel wird die OSSE vorgestellt und danach Unterschiede zur klassischen Softwareentwicklungsmethode aufgezeigt. Anschließend werden die wichtigsten Vor- und Nachteile und daraus resultierende Konsequenzen diskutiert.

5.1 Unterschiede zur klassischen Methode

Die OSSE wird oftmals als Revolutionierung der IT-Branche betrachtet, da die Entwicklungsarbeit nicht mehr in einem geschlossenen Team, sondern von einer beinahe unendlich grossen Community an Entwicklern geleistet wird. Die Community setzt sich aus Personen mit verschiedensten sozioökonomischen und kulturellen Hintergründen zusammen, da jeder und jede Interessierte ohne weiteres der Community beitreten kann. Dabei spielt weder Geschlecht, Alter, Ethnie, Bildung oder die Kultur der potentiellen Mitglieder eine Rolle.

Khanjani und Sulaiman (2011b, S. 647) verglichen die OSSE mit der klassischen Softwareentwicklung. Dabei listeten sie neun Punkte auf, in denen sich die beiden Entwicklungsarten grundlegend unterscheiden:

1. *Rechte der Kunden.* In der traditionellen Softwareentwicklung ist das Endprodukt durch ein Copyright geschützt und wird an Kunden verkauft, um Geld damit zu verdienen. Bei der OSSE ist die Software für alle frei zugänglich und kann beliebig verändert werden, sofern dies nicht durch eine entsprechende Open Source Lizenz anders geregelt ist¹⁷.
2. *Organisation.* Auf dem traditionellen Weg wird der gesamte Entwicklungsprozess systematisch und formell geplant, anschliessend von ausgebildeten Entwicklern umgesetzt, von einem zentralen Management-Team überwacht und gelegentlich mittels Updates und Ähnlichem

¹⁷ Siehe 4.2

aktualisiert. Hinter der OSSE steckt kein spezifisches Team und die beteiligten Entwickler können sich bezüglich ihren Fähigkeiten stark unterscheiden. Da bei OSS aber meist viel mehr Leute an der Entwicklung des Projekts beteiligt sind, werden häufiger Aktualisierungen veröffentlicht. Die Entscheidung, was in den Quellcode der Original-Software aufgenommen wird und was nicht, wird dabei in der Regel vom Projekttinitiator oder einer Gruppe der aktivsten Entwickler entschieden. Die Organisation und der Aufbau der beiden Entwicklungsprozesse unterscheiden sich also deutlich.

3. *Entstehung der Softwarequalität.* Die Qualität der proprietären Software stammt hauptsächlich aus der zentralen Führung und der damit einhergehenden Kontrolle. Die Qualität der OSS entsteht durch die ständige Kontrolle des Quellcodes durch die Community¹⁸ und die beinahe unendlich grosse Auswahl an zur Verfügung stehenden Softwarekomponenten, die in die entwickelte Software eingebaut werden könnten. Zu beachten ist aber, dass zu viele beteiligte Personen hinderlich für das Ergebnis sein können, da die Community aus vielen Laien besteht (Diffe, 2003). Die Qualität proprietärer und freier Software muss nicht zwingend verschieden sein, entsteht aber auf unterschiedliche Art und Weise.
4. *Softwareanforderungen.* Durch die Offenheit von OSS können die Anforderungen der Kunden deutlich besser identifiziert werden, als dies bei der traditionellen Softwareentwicklung der Fall ist. Darüber hinaus können die Benutzer bei OSS selber an der Entwicklung der Anwendung mitarbeiten und ihre Wünsche direkt einbringen. Bei proprietärer Software hingegen können Benutzer meist erst nach einer finalen Testphase oder gar erst nach der Veröffentlichung ein Feedback abgeben.
5. *Qualitätssicherung.* Während bei der traditionellen Methode nicht klar ist, wie viele Teams in der Qualitätssicherung arbeiten, kann dies in der Entwicklung von OSS anhand von Logs und den im Quellcode hinterlegten Verweisen klar nachvollzogen werden. Benutzern fällt es somit

¹⁸ Siehe Raymond (1998, 2001) der behauptet, dass es eigentlich keine Fehler gibt, wenn nur genügend Augen darüber schauen.

einfacher ein objektives Urteil über die Qualität der Projektführung zu fällen.

6. *Entwicklungsziele.* Ziel der traditionellen Softwareentwicklung ist es, durch die Anwendung der Software Marktanteile zu erlangen und Gewinne zu erwirtschaften. OSS hingegen wird oft zum Lösen eines subjektiven Problems entwickelt. Ob daraus finanzielle Erträge entspringen, spielt sehr oft eine eher nebensächliche Rolle.
7. *Fehleranfälligkeit.* Gemäss Koru, Zhang und Liu (2007; zitiert in Khanjani & Sulaiman, 2011b, S. 648) weist traditionell entwickelte Software tendenziell einen grösseren Quellcode auf als OSS. Dies führt dazu, dass der Aufwand für das Finden und Korrigieren aller Fehler bei proprietärer Software entsprechend grösser ist als in der OSSE. Auch sind bei der klassischen Softwareentwicklung die Programmierer-Teams im Vergleich zur OSS-Community eher klein sind. Zusammen erhöhen diese beiden Tatsachen die Wahrscheinlichkeit für das Auftreten von Fehlern, woraus Probleme betreffend Sicherheit, Benutzerfreundlichkeit, etc. entstehen können.
8. *Verteilung der Aufgabenlast.* In der traditionellen Softwareentwicklung fällt nach der Veröffentlichung der Software ein vergleichsweise grosser Aufwand an, da davor die Testmöglichkeiten sehr beschränkt sind. Bei der OSSE sind Entwickler und Benutzer oftmals ein und dieselbe Person, was eine parallele Entwicklung und Testung der Software ermöglicht. Aufgrund dieser Gegebenheit können sich Open Source Entwickler nach der Veröffentlichung der Software grösstenteils darauf konzentrieren, eine neuere, bessere Version der Software zu entwickeln. „Klassische“ Entwickler hingegen müssen nach der Veröffentlichung viel mehr Zeit in die Behebung von Fehlern der aktuellen Version investieren.
9. *Besitzverhältnisse.* Im Gegensatz zum traditionellen Ansatz, bei dem die Software zu jedem Zeitpunkt dem entwickelnden Unternehmen gehört, gibt es in der OSSE keine Ansprüche, die an die Software gestellt werden können: Sie befindet sich jederzeit im Besitz der Öffentlichkeit.

Auch wenn am Ende zwei Produkte vorliegen, die sich lediglich durch die verwendete Lizenz unterscheiden, gibt es doch gewaltige Differenzen in den Entwicklungsprozessen proprietärer Software und OSS. Grösster Unterschied ist das Verhältnis der Entwickler zueinander und die Motivation, wieso sie an einem Projekt mitarbeiten. Während sich bei der OSSE beliebige Individuen freiwillig zu einer Gemeinschaft formieren und Software entwickeln, ist die Variation in einem klassischen Projekt deutlich geringer. Die involvierten Entwickler sind Mitarbeiter des Unternehmens, welches das Projekt durchführt. Aus diesem Setting resultiert eine Einschränkung betreffend der Generierung neuer Ideen. Während bei der OSSE beinahe unendlich viele Personen Verbesserungsvorschläge einbringen können, setzt sich das Team in der traditionellen Softwareentwicklung meistens aus den gleichen Personen zusammen. Dadurch ist die Ideenfindung auf einige wenige Personen beschränkt.

Im folgenden Kapitel werden weitere Vor- und Nachteile der OSSE diskutiert, sowie eine Evaluation durchgeführt.

5.2 Fazit

Die Methode der OSSE erfreut sich einer grossen Anhängerschaft und hat sich den Ruf einer effektiven Entwicklungsmethode erarbeitet. Durch die grosse Flexibilität von Open Source, ist es möglich, Software zu entwickeln, die exakt den Bedürfnissen der Endbenutzer entspricht. Raymond (2001) argumentiert, dass Softwareentwicklung auf *flexiblen*, nicht *starr* Ansätzen basieren sollte. Flexibel ist in diesem Fall so zu verstehen, dass der Entwicklungsprozess bei der OSSE fortlaufend an die sich ändernden Bedürfnisse der Community angepasst wird. Bei der klassischen Softwareentwicklung hingegen werden die Anforderungen oftmals lediglich zu Beginn eines Projekts erhoben und anschliessend nicht mehr angepasst. Eine flexible Softwareentwicklung muss aber nicht zwangsläufig die beste Methode sein, und Open Source ist auch nicht die einzige Möglichkeit um eine flexible Entwicklung zu ermöglichen.

Wird die klassische Softwareentwicklungsmethode z.B. mit agilen Projektführungsmethoden wie Scrum¹⁹ kombiniert, wäre es aufgrund dessen sehr flexiblen Aufbaus ebenfalls möglich, ähnliche Resultate wie bei der OSSE zu erreichen.

Khanjani und Sulaiman (2011a, S. 549) untersuchten die Vor- und Nachteile der Open Source Entwicklungsmethode. Die Ergebnisse dieser Studie sind in Tabelle 2 dargestellt. Eine der grössten Stärken der OSSE ist ihre Flexibilität. Diese zeigt sich darin, dass bereits vorhandene Anwendungen problemlos in ein bestehendes System integriert oder zusammen mit anderen verwendet werden können. Es ist aber auch möglich, sehr schnell Fortschritte in der Programmentwicklung zu erzielen, da in den Communities viele Programmierer den Quellcode kostenlos auf Fehler überprüfen und weiterentwickeln können.

Das Internet macht es möglich, viel Wissen in Open Source Communities zu transferieren und daraus zu beziehen. Sowohl das Wissen der Community, als auch die entwickelte Software können weiterentwickelt und wiederverwendet werden. Diese Kosteneffizienz wird zusätzlich gesteigert, da die erstellten Anwendungen als Basis für weitere Entwicklungen verwendet werden können.

Die OSSE bringt aber auch Nachteile mit sich. So dauert es aufgrund der oftmals schlechteren Dokumentation der OSSE länger, bis sich neue Entwickler in einem Projekt zurechtgefunden haben. Diese Unstrukturiertheit von OSSE-Projekten spricht für den Gebrauch der klassischen Softwareentwicklungsmethode. Ebenso ist es einfacher ein Projekt zu kontrollieren, wenn die daran teilnehmenden Entwickler gewisse Verpflichtungen gegenüber dem Projektleiter haben und das Projekt nicht jederzeit verlassen können. Auch dies ist ein Vorteil herkömmlicher Softwareentwicklung gegenüber OSSE, bei der Entwickler nicht finanziell entschädigt werden, dafür aber auch keinerlei Leistungsverpflichtungen haben. Die Frage der Verantwortung stellt ein grundsätzliches Problem der OSSE dar. Meist ist der Initiator eines Projekts auch

¹⁹ Scrum ist ein Konzept der Projektleitung in der Softwareentwicklung, bei dem mehrere abhängige Teams am gleichen Projekt arbeiten und das andere Team auf täglicher Basis über den jeweiligen Fortschritt informieren und Feedbacks einholen.

dafür verantwortlich, oftmals gibt es aber auch keinen Hauptverantwortlichen, was zu Verzögerung oder Verwirrung führen kann.

Die Grösse und dadurch bedingte Unübersichtlichkeit vieler OSS-Projekte macht es schwierig abzuschätzen, wie viele Mitarbeiter benötigt werden. Das Fehlen eines formalen, zentralisierten Prozesses verschlimmert die Unübersichtlichkeit zusätzlich.

Mit der OSSE hat sich eine Softwareentwicklungsmethode etabliert, die neue Wege einschlägt. Software wird nicht mehr in einer festen hierarchischen Unternehmung entwickelt, sondern in einer über das Internet lose zusammengewürfelten Gruppe von Softwareentwicklern. Trotz der Nachteile, die mit dieser Konstellation einhergehen, überwiegen die oben genannten Vorteile. In Tabelle 2 werden auch deutlich mehr Vor- als Nachteile von OSSE aufgeführt.

Diese Art der Softwareentwicklung stellt neue Anforderung an Projektverantwortliche und alle Teilnehmer. Es reicht nicht mehr nur, eine Gruppe von Softwareentwicklern zu coachen. Da die involvierten Programmierer in der Regel keinerlei Verpflichtungen gegenüber der Projektleitung haben, müssen die Entwickler zuerst motiviert werden, überhaupt am Projekt mitzuarbeiten. Diese Motivation gilt es auch im Anschluss hoch zu halten, da Beitragende nicht nur jederzeit aus dem Projekt aussteigen, sondern auch den Quellcode oder Ähnliches mitnehmen können.

Obwohl die OSSE viele Vorteile hat, ist sie nicht für jedes Unternehmen geeignet. Ist ein Unternehmen beispielsweise nicht flexibel, kann es nicht vom Vorteil der OSSE-Flexibilität profitieren. Gleich verhält es sich mit dem Wissenstransfer. Besteht kein Interesse daran, mit der Community Wissen auszutauschen oder Ideen aus der Community zu berücksichtigen, kann die Offenheit des Quellcodes sogar ein Nachteil sein²⁰. Deshalb ist es für Unternehmen wichtig, im Vorfeld einer Umstellung auf OSSE nicht nur allgemeine Vor- und Nachteile dieser Softwareentwicklungsmethode zu betrachten, sondern auch die eigenen Absichten und Möglichkeiten zu klären.

²⁰ Siehe 0

	Benutzer	Entwickler	System
Vorteile	<ul style="list-style-type: none"> ▪ Flexibilität ▪ Hoher Stellenwert ▪ Mitwirken an interessanter Arbeit ▪ Verfügbarkeit des Quellcodes ▪ Möglichkeit den Quellcode zu verändern ▪ Wissenstransfers ▪ Gesteigerte Motivation ▪ Grössere Auswahl und Kontrolle 	<ul style="list-style-type: none"> ▪ Möglichkeit eigene Lösungen zu kreieren ▪ Wiederverwendung von Funktionalitäten ▪ Möglichkeit Probleme kostenlos überprüfen zu lassen ▪ Weniger Fehler zu Beginn eines Entwicklungsprozesses ▪ Höhere Motivation 	<ul style="list-style-type: none"> ▪ Fehlerentdeckung ▪ Open Source Werkzeuge ▪ Verlässlichkeit ▪ Veränderbarkeit ▪ Weitentwickelte Technologie ▪ Kosteneffizient ▪ Schnelle Entwicklung ▪ Tragbarkeit ▪ Möglichkeit der Erweiterung ▪ Wiederverwendbarkeit ▪ Geringe Kosten ▪ Viele Lizenzen
Nachteile	<ul style="list-style-type: none"> ▪ Unnütze Dokumentation ▪ Unstrukturierte Entwicklung ▪ Nicht verlässliche Individuen 	<ul style="list-style-type: none"> ▪ Mangel an Werkzeugen ▪ Zusammenarbeit mit neuen Entwicklern ▪ Oft sehr grosse Projekte (unübersichtlich) 	<ul style="list-style-type: none"> ▪ Fehlen eines formalen, zentralisierten Prozesses ▪ Schlechtes Design ▪ Abschätzen der benötigten Mitarbeiter schwierig ▪ Problem der Verantwortung/Verlässlichkeit (niemand fühlt sich zuständig) ▪ Sehr viele unabhängige Versionen möglich ▪ Hohe kurzfristige Kosten

Tabelle 2 Vor- und Nachteile von OSSE

Die in dieser Tabelle verwendeten Kategorien basieren auf der Unterteilung Fugettas (2003). Quelle: Khanjani & Sulaiman (2011a, S. 549)

6 Evaluation von Open Source Software

In diesem Kapitel wird OSS aus der ethischen, technischen und ökonomischen Perspektive evaluiert. Der Fokus wird dabei auf die Sicht der Unternehmen gelegt, da diese im nächsten Kapitel zur Erstellung des Migrations-Frameworks im Zentrum steht.

6.1 Ethische Aspekte

Zu den diskutierten ethischen Aspekten zählen unter anderem die Chancengleichheit für Individuen und Unternehmen, eine durch die Offenheit des Quellcodes bedingte faire wirtschaftliche Ausgangslage und die Förderung der Gemeinschaft.

Einer der grössten Vorteile von Open Source ist die Chancengleichheit. OSS bietet eine Plattform, die von allen in gleichem Masse genutzt werden kann. Durch den offenen Quellcode und die durch Open Source Lizenzen bedingten Rechte und Pflichten der Nutzer haben alle dieselben Möglichkeiten. Dabei spielt es keine Rolle, ob es sich um ein Unternehmen oder einen individuellen Entwickler handelt. Weiterhin spielen auch weder Geschlecht, Alter, Bildung, Kultur, Ethnie, sozioökonomischer Status noch geografische Lage eine Rolle. Es gibt keine Grundanforderungen, die erfüllt werden müssen, um einer Open Source Community beizutreten, wie dies beispielsweise bei Unternehmen oder Universitäten der Fall ist. Diffe (2003) behauptet, dass die OSSE nicht zwingend davon profitiert, wenn beliebig viele Programmierer an einem Projekt mitarbeiten. Da nicht alle über genügend hohe Fähigkeiten und Wissen verfügen, kann die Qualität durch minderwertige Inputs leiden. Allerdings argumentiert Diffe (2003) auch, dass diese Nachteile durch Beiträge von Entwicklern, welche die Software selber einsetzen wollen, kompensiert werden. Durch die hohe Eigenmotivation hochwertige Software zu erstellen, werden sich diese Entwickler besonders darum bemühen, allfällige Fehler auszubessern. Dadurch steigt die Qualität der Software.

Auf Unternehmensebene sind die Chancen bei OSS ebenfalls ausgeglichener als in der Entwicklung proprietärer Software. Um OSS einsetzen zu können, sind keine grösseren Investitionen, wie z.B. der Kauf einer Softwarelizenz notwendig. Dies ermöglicht es auch kleineren Unternehmen oder Start-

Ups dieselben Produkte zu verwenden und weiterzuentwickeln, wie sie von Grosskonzernen eingesetzt werden.

Auch auf der Regierungsebene fördert OSS die Chancengleichheit. Laut Bundesrätin Simonetta Sommaruga, Vorsteherin des eidgenössischen Justiz- und Polizeidepartments, hat der Einsatz von Open Source in öffentlichen Verwaltungen grosses Potential. Finanzschwächere Kantone und Gemeinden können OSS einsetzen und so trotz geringem Budget qualitativ hochwertige Dienstleistungen anbieten. Dies wird durch die kostenlose Mehrfachverwendung von Open Source Softwarelösungen erreicht (Stürmer, 2009, S. 10).

Ein weiterer Vorteil von Open Source ist die Förderung der Gemeinschaft, die durch die Zusammenarbeit in einer Community zustande kommt. Die Community kann als zentrale Wissensdatenbank betrachtet werden, an der alle Beteiligten gleichermassen mitarbeiten können. Wird ein Open Source Projekt abgeschlossen, gehen die daraus gewonnen Erkenntnisse nicht verloren oder sind an einzelne Entwickler gebunden, wie dies bei proprietärer Software der Fall ist. Die Erkenntnisse gehen in den Open Source Wissenspool über, sind für jedermann zugänglich und können für weitere Projekte oder Ideen verwendet werden. Durch das gemeinsame Entwickeln wird der Wissensstand der Gemeinschaft erhöht, wovon alle profitieren. Bei der klassischen Softwareentwicklung hingegen verbleibt das gewonnene Wissen in einem einzelnen Unternehmen und nur dieses profitiert davon.

Die oben erwähnten Aspekte sprechen dafür, dass Open Source aus ethischer Sicht vermehrt eingesetzt werden sollte. OSS bietet faire Rahmenbedingungen für alle Beteiligten und begünstigt eine Weiterentwicklung der Gemeinschaft. Durch den gemeinsamen Wissenspool ist eine nachhaltige Softwareentwicklung möglich, bei der sowohl Entwickler, als auch Nutzer direkt von der Erfahrung anderer profitieren können und niemand ausgeschlossen wird. Die OSSE ermöglicht eine schnellere und effizientere Lösungsfindung von Softwareproblemen, da bereits gemachte Erfahrungen der Community verwendet und darauf aufbauend direkt neue Ideen entwickelt werden können.

6.2 Technische Aspekte

Auch technische Aspekte müssen bei der Evaluation von Open Source betrachtet werden. Dazu gehören beispielsweise die Vor- und Nachteile offenen Quellcodes, der Kompatibilität mit anderen Anwendungen, sowie verschiedene Sicherheitsaspekte.

Abbildung 5 zeigt anhand von fünf Beispielen, welchen Einfluss die Offenlegung des Quellcodes auf die entstehende Software haben kann. Durch die Einbindung der Open Source Community kann eine Vielzahl neuer, kompetenter Mitarbeiter gewonnen werden. Dies ist aber nur der Fall, wenn die Mitglieder der Gemeinschaft auch wirklich über entsprechend gute Fähigkeiten verfügen, um beispielsweise Fehler im Quellcode zu entdecken und zu korrigieren. Koru, Zhang und Liu (2007) bemängeln, dass trotz dem Vorhandensein vieler Entwickler eine vollständige Überprüfung des Quellcodes nicht sichergestellt ist. Auch wenn der Quellcode von der Community auf Fehler geprüft wurde, kann eine Anwendung immer noch Fehler enthalten. Dies macht eine abschliessende Prüfung durch die Projektleitung in jedem Fall notwendig.

Die Offenlegung des Quellcodes kann für eine Unternehmung auch ein Sicherheitsrisiko bedeuten. Schwachstellen können schneller entdeckt und angegriffen werden. Allerdings ist offener Quellcode nicht zwangsläufig weniger sicher. Da er von einer Vielzahl von Softwareentwicklern überprüft wird, werden diese Sicherheitslücken deutlich schneller wieder geschlossen. Ob sich ein Unternehmen dieser Gefahr aussetzen will, hängt von dessen Risikobereitschaft ab. Die weiteren Vorteile aus Abbildung 5 sprechen jedoch zusammen mit der meist kürzeren Entwicklungszeit der Software für eine Offenlegung des Quellcodes. Ausserdem ist das Sicherheitsrisiko, aufgrund der Verwendung offenen Codes angegriffen zu werden dem Sicherheitsvorteil, den die Überprüfung durch eine riesige Community bringt, unterlegen.

Laut den Befunden der Open Source Studie 2012 schätzen Unternehmen die Interoperabilität als einer der wichtigsten Vorteile von OSS ein²¹. Gemäss Fuggetta (2003, S. 83) wird diese Kompatibilität unter Anwendungen durch offene Standards erreicht. Befürworter behaupten, OSS sei die einzige

²¹ Siehe Abbildung 12

Möglichkeit offene, nicht beeinflussbare Standards sicherzustellen. Allerdings treten gemäss Stark (2014) bei der Weitergabe von Dokumenten tendenziell weniger Kompatibilitätsprobleme auf, wenn proprietäre Software verwendet wird. Diese beiden scheinbar widersprüchlichen Aussagen lassen sich miteinander vereinbaren. Die Kompatibilität ihrer Produkte mit Drittanwendungen ist für proprietäre Anbieter entscheidend, da kein einzelnes Unternehmen in allen Bereichen die Standards setzt. Deswegen müssen Anwendungen verschiedener Hersteller problemlos kombiniert werden können. Ein gutes Beispiel hierfür ist die Entwicklung des Microsoft Office für das Apple Betriebssystem. Bei OSS hingegen wird unter Kompatibilität verschiedener Anwendungen die mit geringem Aufwand verbundene Anpassung des Quellcodes verstanden. Dies ist auch für Unternehmen interessant. Lässt sich Software problemlos in eine bestehende IT-Landschaft integrieren, fallen dementsprechend geringere Umstellungskosten an.

Die OSSE-Methode versucht folgendermassen Kompatibilität und dadurch auch offene Standards sicherzustellen: Unternehmen geben Teile des Quellcodes bestimmter Produkte frei und beauftragen so die Open Source Community, für sie eine Anbindung an ein bestehendes Betriebssystem zu entwickeln²². So entsteht eine Win-Win-Situation. Das Unternehmen erhält eine günstige und kompatible Softwarelösung für ein bestehendes Problem und stellt der Community im Gegenzug Wissen zur Verfügung, das sonst wahrscheinlich nicht veröffentlicht worden wäre.

Ein weiterer technischer Evaluationspunkt ist die Sicherheit. OSS wird als Weg betrachtet Nutzer zu schützen. Laut Fuggetta (2003, S. 87) sollte es möglich sein, Software auf deren Einhaltung der Sicherheitsstandards zu überprüfen. Die Offenheit des Quellcodes von OSS erlaubt genau dies, da er jederzeit von allen Benutzern eingesehen werden kann. Weiter ist auch eine effizientere Wartung der Software möglich.

Raymond (2001) behauptete, dass alle Fehler in einem Quellcode gefunden werden, wenn nur genügend Augen darüber schauen. Da OSS in der Community entwickelt wird, ist die Anzahl Personen, die den Code bearbeiten

²² Ein Beispiel für dieses Vorgehen ist Intel (Intel Corporation, 2015).

und kontrollieren, meist grösser als in der klassischen Softwareentwicklung. Daher wird OSS meistens in Serverumgebungen verwendet. Auch beim BIT wird OSS hauptsächlich dort eingesetzt²³.

Gemäss Diffe (2003) kann ein Quellcode auch von zu vielen Augen untersucht werden, weshalb er nicht zwangsläufig sicherer sein muss als proprietärer Quellcode. Zudem sei es sinnvoller, wenn eine kleine Gruppe von Experten den Quellcode überprüft, anstelle einer grossen Gruppe von Laien. In jedem Fall gilt es die Entwicklung von OSS gut zu überwachen, die Arbeit zu kontrollieren und nicht blind darauf zu vertrauen, dass die Software keine Fehler enthält.

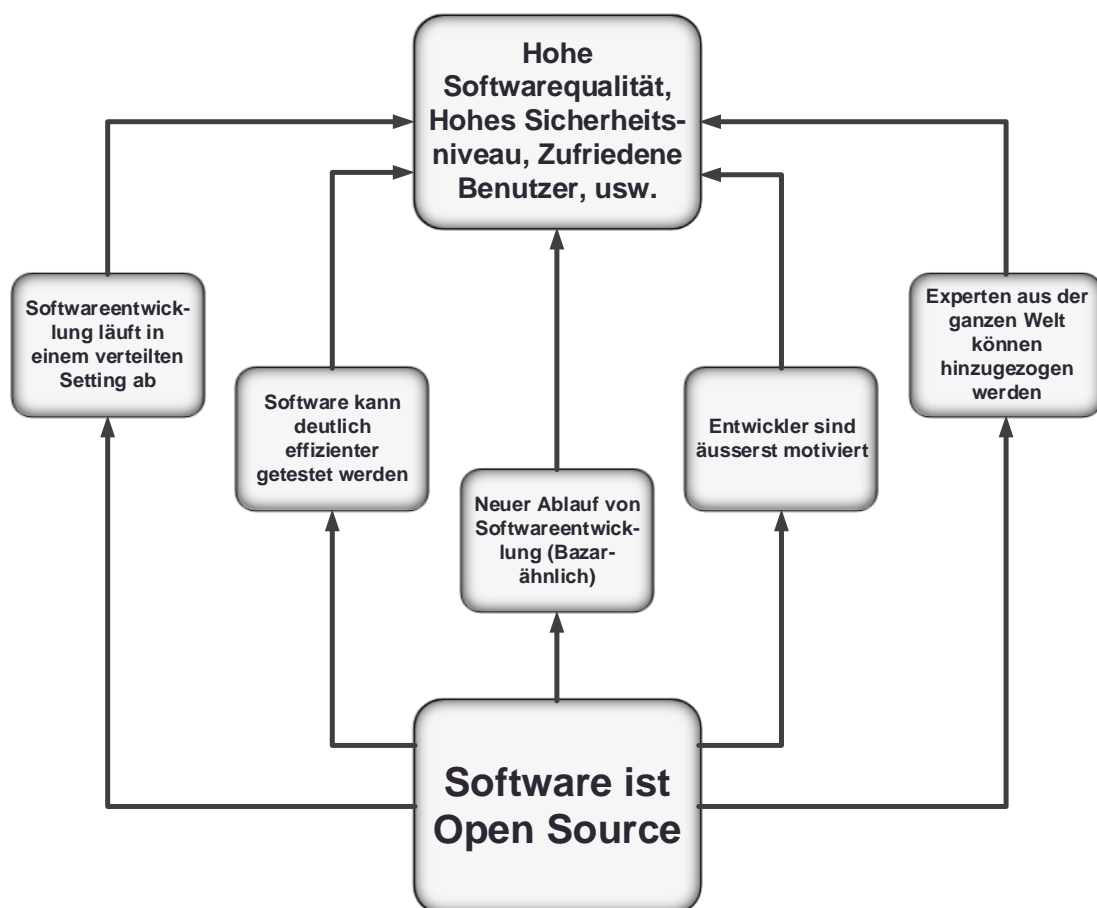


Abbildung 5 Auswirkungen der Offenheit von Quellcode

In diesem Diagramm werden ausgewählte (positive) Effekte dargestellt, die meist eintreten, wenn eine Software Open Source ist oder in eine solche umgewandelt wird.

Quelle: Fuggetta (2003, S. 84)

²³ Siehe 8.4

6.3 Ökonomische Aspekte

Ökonomische Faktoren sind meist die erstgenannten Argumente, die für eine Umstellung auf OSS sprechen. Dazu gehören unter anderem Kosteneinsparungen, eine erhöhte Unabhängigkeit, eine bessere Reputation oder ein verändertes Personalmanagement.

Ein Umstieg auf OSS bewirkt eine unmittelbare Kosteneinsparung, da keine Lizenzkosten mehr anfallen (Hauge, Ayala, & Conradi, 2010, S. 1144). Auch in der Open Source Studie (2012) wird dieses Sparpotential als ein wichtiger Vorteil von OSS genannt²⁴. Bei einer unkoordinierten Umstellung auf OSS können die Einsparungen jedoch rasch durch Umstellungskosten wie Umschulungen der Mitarbeiter oder Datenmigrationen zunichte gemacht werden und sogar zu einem Mehraufwand führen (Ernst & Young AG, 2011, S. 7).

Obwohl bei grösseren Unternehmen die Umstellungskosten allgemein höher sind, lohnt sich eine Umstellung auf OSS trotzdem. Langfristig wird der anfängliche Mehraufwand durch die geringeren oder ganz wegfallenden Ausgaben für Lizenzen, Support, etc. wieder wettgemacht.

Durch eine Migration auf OSS gewinnt das Unternehmen die Möglichkeit, frei auswählen zu können, bei wem eine bestimmte Dienstleistung bezogen wird. Bei einer genügend grossen Anzahl an Anbietern wächst die Wahlfreiheit und damit die Verhandlungsstärke, da eine Verhandlung bei nicht zufriedenstellenden Angeboten leicht abgebrochen und mit anderen Anbietern verhandelt werden kann. Abbildung 12 kann entnommen werden, dass diese Unabhängigkeit von den Teilnehmern der Open Source Studie 2012 hoch bewertet wird. Die erlangte Unabhängigkeit führt aber auch dazu, dass Haftung und Support nicht automatisch geklärt sind. Dies sind die beiden meistgenannten Kritikpunkte in Abbildung 13.

Durch die grössere Unabhängigkeit sind Unternehmen zwar freier in der Wahl des Supportpartners, jedoch dauert die Suche nach einem geeigneten Partner tendenziell länger. Im Gegensatz zu proprietärer Software können Supportdienstleistungen bei OSS nicht immer vom Softwarelieferanten bezo-

²⁴ Siehe Abbildung 12

gen werden. Dies ist nicht zwingend ein Nachteil: Bei OSS kann jeder Supportanbieter auf den Quellcode zugreifen und verfügt so über die gleichen Voraussetzungen wie die Entwickler. Bei proprietärer Software hingegen würde dies wahrscheinlich zu einer schlechteren Servicequalität führen, da Drittanbieter keinen Zugriff auf den Quellcode haben und somit nicht über dasselbe Knowhow verfügen wie der Softwareentwickler.

Die vorbildliche Verwendung und Förderung von OSS wie dies beispielsweise von IBM gemacht wird, kann die Reputation eines Unternehmens steigern. Insbesondere Technologieunternehmen und öffentliche Institutionen werden in den Medien gut dargestellt, wenn sie eigene Software unter einer Open Source Lizenz veröffentlichen. Von öffentlichen Institutionen wird überdies erwartet, dass vermehrt OSS eingesetzt wird, weil dadurch lokale Innovations- und Wirtschaftsförderung betrieben und die Chancengleichheit verbessert werden kann (Ernst & Young AG, 2011, S. 8). Andererseits kann das Image leiden, wenn beispielsweise schlechter Quellcode entwickelt, oder zu wenig an die Community zurückgegeben wird.

Ein Unternehmen, das den Quellcode einer intern entwickelten Software veröffentlicht, muss stets kontrollieren, welche Beiträge in den Quellcode des Endproduktes aufgenommen werden. Das Berücksichtigen und Aufnehmen von Beiträgen aus der Community wird oftmals positiv bewertet und kann den Ruf eines Unternehmens steigern. Dies kann auch das Geschäftsergebnis verbessern, da beispielsweise das Vertrauen der Kunden in das Unternehmen und seine Produkte gestärkt wird.

Eine Umstellung auf OSS betrifft nicht nur das Unternehmen, sondern auch die Mitarbeiter. Eine Migration auf OSS kann sogar im Fall der Abwerbung einiger Mitarbeiter durch andere Firmen noch sinnvoll sein. Durch den Zugang zu einem deutlich grösseren Arbeitsmarkt können mit relativ geringem Aufwand neue Mitarbeiter gefunden werden. Wenn Entwickler das Unternehmen als attraktiv einstufen, ist es einfacher, bereits angestellte, wertvolle Mitarbeiter zu halten. Auf diese Weise kann das durchschnittliche Fähigkeitsniveau der Mitarbeiter sogar steigen. Aufgrund der Qualitätsüberwachung durch

die Community und der Signaling-Absichten²⁵ werden die Mitarbeiter eines Unternehmens motiviert, gute Arbeit zu leisten.

6.4 Fazit

Trotz teilweise berechtigter Kritiken an und den Nachteilen von OSS steigt das Interesse stetig an. Fuggetta (2003, S. 77) führt dies auf drei Gründe zurück. Erstens dienen Produkte wie Linux und Apache als Paradebeispiele für den Erfolg und die Qualität von Open Source. Die Erfolge der dieser beiden Anwendungen erhöhen sowohl die Motivation, an einem Projekt mitzuarbeiten, als auch die Bereitschaft, in ein solches zu investieren. Zweitens steigt die Unzufriedenheit mit dem Scheinmonopol Microsofts und anderer ähnlich grosser Anbieter proprietärer Software, weshalb aktiv nach Alternativen gesucht wird. Drittens gibt es auch immer mehr Vertreter, die der Ansicht sind, *klassische* Methoden der Softwareentwicklung scheiterten immer mehr an der Vorgabe, zufriedenstellende Lösungen zu präsentieren.

Das angesprochene, gesteigerte Interesse zeigt sich auf verschiedenen Ebenen (Fuggetta, 2003, S. 77ff). Auf individueller Ebene stehen grosse Gemeinschaften, die sich oft als Reaktion auf die stärker werdende Abneigung gegenüber proprietärer Software bilden. Die Tatsache, dass solche Communities grösstenteils aus unbezahlten Laien bestehen, ist keineswegs ein Nachteil. Im Gegenteil, die Benutzer sind eher dazu motiviert, neuere und bessere OSS zu kreieren (Lerner & Tirole, 2002, S. 204). Zusätzlich bietet sich so für Unternehmen die Gelegenheit, günstiges Expertenwissen zu bestehenden oder neuen Projekten hinzuzuziehen. Damit dies gelingt, muss ein Unternehmen den eigenen Projekten einen gewissen „Coolness“-Faktor verleihen. Wenn Entwickler es als erstrebenswert betrachten, an einem bestimmten Projekt mitarbeiten zu können, werden sie sich mehr dafür engagieren. Für die Unternehmen gilt es deshalb kompetente Führungspersonen einzusetzen, welche diese Motivation hoch halten und ihre Entwickler so zu Bestleistungen anspornen können.

²⁵ Unter Signaling wird der Abbau von Informationsasymmetrien durch Bereitstellung von Informationen verstanden. Beim Signaling versucht ein Arbeitnehmer, sich möglichst gut zu präsentieren, indem die eigenen Stärken hervorgehoben werden.

Unternehmen bilden die zweite bei Fuggetta (2003) erwähnte Ebene. Viele Unternehmen fokussieren ihre Bemühungen auf OSS, da sie dadurch z.B. die Möglichkeit sehen, bestehende Monopole zu untergraben oder eine konkurrenzfähige Open Source Anwendung zu schaffen. Dieser neu gesetzte Fokus bedingt oftmals eine Umstellung der Geschäftsstrategie weg von Lizenz-, und hin zu Dienstleistungserträgen. Interne Veränderungen wie Personalfluktuationen werden in Kauf genommen, um so von den Kosteneinsparungen oder der gestiegenen Unabhängigkeit gegenüber Anbietern proprietärer Software zu profitieren.

Auf der dritten Ebene stehen nach Fuggetta (2003) Regierungsbehörden. Für Regierungen ist Open Source aus zwei Gründen interessant. Erstens wird durch die Förderung von OSS die Softwareindustrie des eigenen Landes gestärkt und gleichzeitig die Dominanz ausländischer Firmen reduziert. Zweitens verbessert OSS die ökonomische Situation der Regierungsbehörde als Unternehmen, da OSS im Vergleich zu proprietären Anwendungen als sicherer und verlässlicher gilt. Genauer wird der Einfluss von OSS auf einen Regierungsbehörde in Kapitel 0 am Beispiel des Bundesamtes für Informatik und Telekommunikation (BIT) beschrieben.

Trotz der steigenden Bekanntheit und vermehrten Verwendung von OSS auf allen Ebenen, ist immer noch eine grosse allgemeine Verunsicherung vorhanden. Der Mangel an erfolgreichen Open Source Referenzen verstärkt diese Verunsicherung zusätzlich. Als Ursachen werden fehlende Vorkenntnisse, das Misstrauen gegenüber OSS und ein Mangel an Schulungsangeboten genannt. Dies widerspiegelt sich unter anderem in der mangelnden Benutzerakzeptanz oder geringem firmeninternem Open Source Knowhow. Sollen diese Defizite abgebaut werden, reichte es nicht aus, sich lediglich auf die Eigenmotivation von Individuen oder Unternehmen zu verlassen. Open Source Organisationen und Förderer müssen aktiv werden und z.B. attraktive Schulungsangebote und Seminare entwickeln und durchführen. Sowohl Unternehmen, als auch Privatpersonen sollten über die Vorteile von OSS aufgeklärt werden. Bundesrätin Simonetta Sommaruga schlägt vor, dass der Einsatz von Open Source insbesondere an Schulen gefördert wird. Dadurch erfahren Kinder bereits früh, was Open Source ist und wie es verwendet werden kann. Auf

diese Weise kann der digitale Graben geschlossen werden (Stürmer, 2009, S. 10).

Eine Migration auf und Förderung von OSS kann für jedes Unternehmen oder jede Regierung attraktiv sein. Aufgrund bisher genannter Nachteile wie der ungeklärten Supportfrage ist es aber nicht in jedem Fall sinnvoll, OSS einzuführen. Ebenso kann eine Umstellung der Softwareentwicklungsmethode eventuell nicht mit der Strategie eines Unternehmens vereinbart werden. So käme Open Source beispielsweise nicht in Frage, wenn das Management niemand externem Zugriff zum Quellcode der eigenen Produkte gewähren will. Diese Beispiele zeigen, dass durch die Verwendung von OSS entstehende Kosteneinsparungen nicht der einzige Beweggrund sein sollten, eine Umstellung in Erwägung zu ziehen. Im nächsten Kapitel wird ein Framework entwickelt, das über relevante positive und negative Einflussfaktoren einer Entscheidung für Open Source informiert und Unternehmen bei der anschließenden Entscheidungsfindung unterstützt.

7 Framework

Das in diesem Kapitel vorgestellte Framework unterstützt Unternehmen bei der Entscheidung, ob ihre Organisation für eine Umstellung auf OSS bereit ist. Gleichzeitig verdeutlicht das Framework die Vor- und Nachteile, die Open Source für das entsprechende Unternehmen hat. Wie Marwan (2010) behauptet, wäre es möglich ein Unternehmen zu führen, das zu 100 Prozent Open Source Produkte verwendet. Demnach sollte es für jedes Unternehmen möglich sein, zumindest teilweise auf OSS umzustellen.

Es gibt zahlreiche Einflussfaktoren, die sowohl positiv, als auch negativ auf den Entscheidungsfindungsprozess einwirken. Abbildung 6 zeigt in Anlehnung an das das St. Galler Management Modell²⁶ das Umfeld, in dem sich Unternehmen bewegen. Auf der Ebene der Organisation wird zwischen *Softwareutzern* und *Softwareentwicklern* unterschieden, da die damit verbundenen Implikationen einer Umstellung auf OSS unterschiedlich sind. Während bei Softwareutzern lediglich entschieden werden muss, ob proprietäre Software oder eine OSS-Alternative eingesetzt wird, ist diese Entscheidung für Softwareentwickler wesentlich komplexer. Eine Entscheidung für Open Source beeinflusst nicht nur die Produkte, sondern auch den gesamten Entwicklungsprozess.

Die für das Unternehmen wichtigste Ebene ist die Organisationsebene, gefolgt von der Personal- und der Umweltebene²⁷. Die Ebene der Organisation (grüner Kreis in Abbildung 6) betrifft das Fundament des Unternehmens und kann durch dieses direkt beeinflusst werden. Dazu zählen Faktoren wie das Geschäftsmodell, die finanzielle Lage und die Unternehmensgrösse, also Dinge, die nicht an spezifische Personen gebunden sind. Die Personalebene (oranger Kreis in Abbildung 6) ergänzt die Organisationsebene um individuelle Komponenten. Auf der Personalebene ist es wichtig, welche Personen im Unternehmen arbeiten, da sich beispielsweise die Qualität und der Einfluss des

²⁶ Eine Beschreibung des neuen St. Galler Management-Modells kann dem Buch Rüegg-Stürms und Grands entnommen werden (Rüegg-Stürm & Grand, 2014).

²⁷ In der Umweltebene werden keine produktspezifischen Faktoren berücksichtigt. Der entsprechende Softwaremarkt ist beispielsweise den softwareentwicklerspezifischen Faktoren zugeordnet.

Managements stark verändert, wenn Führungspositionen anders besetzt werden. Ein Unternehmen kann die Personalebene beeinflussen, indem z.B. Mitarbeiter ersetzt werden. Das Personal bildet ein Bindeglied zwischen dem Kern der Organisation und dessen Umwelt, da es auf beide Ebenen einen Einfluss hat. Mitarbeiter können sowohl das Geschäftsergebnis ihres Unternehmens, als auch die Situation auf dem Arbeitsmarkt beeinflussen.

Die dritte Ebene ist die Umwelt (roter Kreis in Abbildung 6). Sie kann von den Unternehmen nur indirekt beeinflusst werden. So kann ein Unternehmen beispielsweise eine Gesetzesänderung anstreben, allerdings dauert es sehr lange, bis diese durchgesetzt wird. Da es an der Regierung ist, Gesetze zu ändern, können Unternehmen lediglich den Wunsch einer Änderung äußern und so indirekt Einfluss auf das Geschehen nehmen.

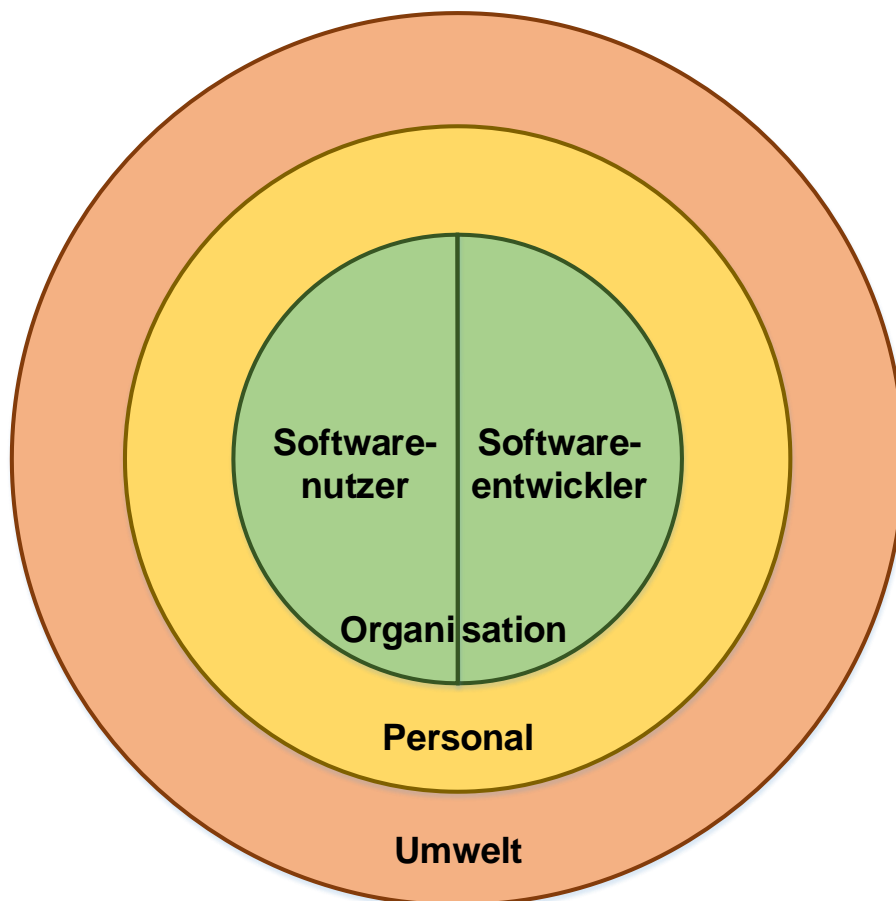


Abbildung 6 Umweltebenen des Unternehmens

In dieser Grafik werden die verschiedenen Ebenen, in denen sich ein Unternehmen bewegt, aufgezeigt und bezüglich der Beeinflussbarkeit durch das Unternehmen farblich unterschieden (grün steht für gut, orange für mittel und rot für schlecht beeinflussbar).

7.1 Grafische Darstellung

In Anlehnung an Abbildung 6 werden in Abbildung 7 die verschiedenen Betrachtungsebenen entsprechend eingefärbt. Obwohl argumentiert werden kann, dass die Nutzer- und Entwicklerperspektive getrennt dargestellt und analysiert werden sollten, werden hier sämtliche Entscheidungs- und Einfluss-ebenen zusammen dargestellt. Diese Zusammenfassung macht Sinn, da die auf den Nutzer wirkenden Einflüsse auch den Entwickler betreffen. Ausserdem soll dieses Framework eine umfassende, holistische Analyse ermöglichen, weshalb alle Ebenen gleichzeitig betrachtet werden müssen. Die einzelnen Kriterien werden in Kapitel 0 im Detail betrachtet. Zur einfacheren Bewertung wurden Multiplikatoren in das Framework integriert. Das Bewertungsverfahren wird in Kapitel 7.3 beschrieben.

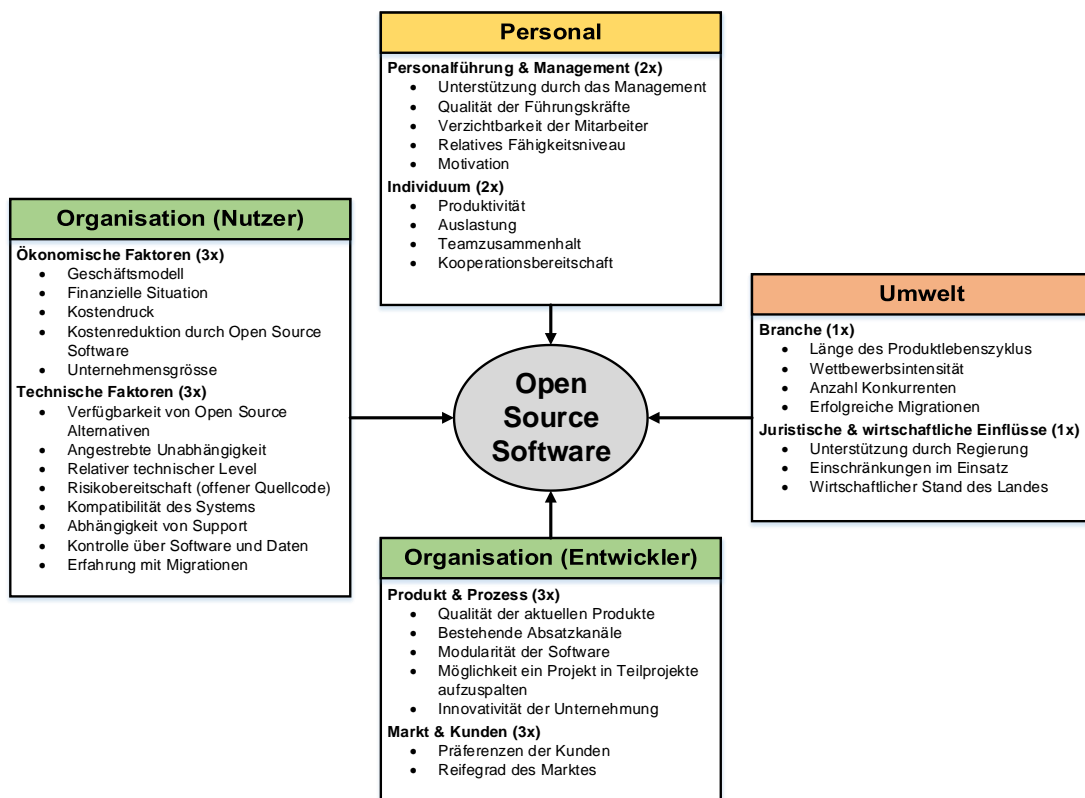


Abbildung 7 Analyseframework für Open Source Migration

Hier wird aufgezeigt, wie die drei Ebenen der Organisation, des Personals und der Umwelt die Entscheidungsfindung, ob auf OSS umgestiegen werden soll, einwirken.

7.2 Entscheidungsebenen

Eine Umstellung auf Open Source betrifft alle Ebenen eines Unternehmens und wird durch Umweltfaktoren wie der rechtlichen und wirtschaftlichen Situation des Landes, sowie branchenspezifischen Faktoren beeinflusst. Die verschiedenen Ebenen werden im Folgenden getrennt beschrieben und analysiert.

7.2.1 Organisation

Die Organisationsebene stellt den Kern eines Unternehmens dar, der auch unabhängig von der Qualität der Mitarbeiter oder von Einflüssen der Umwelt funktionieren muss. Deshalb muss betrachtet werden, ob die unternehmensspezifischen Voraussetzungen für einen Umstieg auf OSS gegeben sind.

Zuerst wird die Perspektive der Softwarenutzer analysiert, anschließend diejenige der Softwareentwickler. Die auf den Softwarenutzer einwirkenden Faktoren gelten auch für den Softwareentwickler. Umgekehrt muss dies nicht zwangsläufig der Fall sein.

7.2.1.1 Softwarenutzer

Bei den für Softwarenutzer relevanten Faktoren wird zwischen *ökonomischen* und *technischen* Faktoren unterschieden. Der betreffende Abschnitt des Frameworks wird in Abbildung 8 vergrößert dargestellt.

Ökonomische Faktoren

Geschäftsmodell. Die Entscheidung bezüglich einer Umstellung auf OSS wird stark durch das Geschäftsmodell eines Unternehmens beeinflusst. Wie in Kapitel 4.2 beschrieben wurde, ist der Verkauf von Softwarelizenzen bei Open Source in der Regel nicht möglich (BITKOM, 2015). Wenn ein Unternehmen seine Gewinne über das Anbieten von Dienstleistungen erzielt, macht eine Umstellung dementsprechend mehr Sinn, als wenn dies hauptsächlich über den Verkauf von Lizenzen geschieht. Eine Anpassung des Geschäftsmodells benötigt viel Aufwand und bringt Unsicherheiten mit sich. Deswegen sollte das Geschäftsmodell nicht von der Art der eingesetzten Software abhängen, sondern unabhängig ausgewählt werden. Die Wahl des Geschäftsmodells hat aus strategischer Perspektive einen deutlich grösseren Stellenwert als die Ent-

scheidung für oder gegen OSS. Unternehmen sollten deshalb zuerst das Geschäftsmodell definieren oder analysieren und danach entscheiden, welche Art von Software(-entwicklung) dazu passt.

Finanzelle Situation. Eine Softwaremigration ist wesentlich einfacher durchzuführen, wenn ein Unternehmen dazu bereit ist, die nötigen Investitionen für neue Hardwarekomponenten, Aus- und Weiterbildungen, etc. zu tätigen. Laut Mijinyawa und Abdulwahab (2014, S. 249) bietet eine stabile finanzielle Situation mehr Spielraum für strategische Entscheide und vereinfacht eine Softwaremigration. Allerdings ist im Fall von Open Source eine Umstellung allgemein einfacher, da die Software kostenlos bezogen werden kann. Aus diesem Grund müssen lediglich Umstellungskosten wie Umschulungen der Mitarbeiter in das Budget aufgenommen werden.

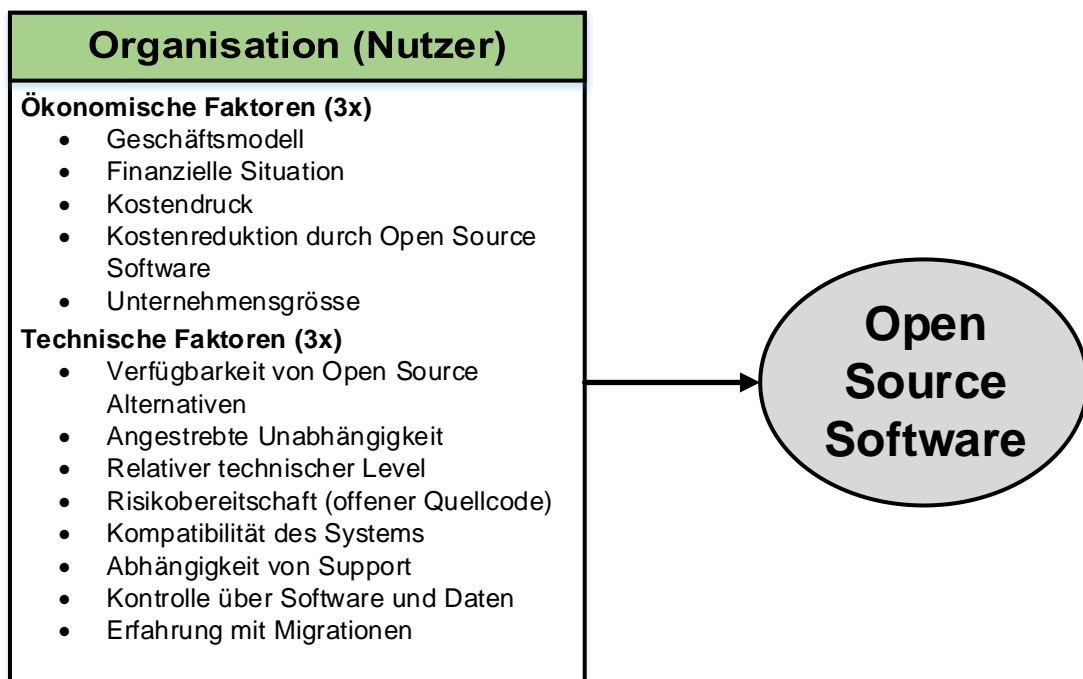


Abbildung 8 Organisationsebene (Nutzer)

Dies ist ein Ausschnitt aus dem erstellten Framework, der die Softwarenutzerperspektive im Detail zeigt.

Kostendruck. Je grösser der Kostendruck auf das Unternehmen ist, desto sinnvoller ist eine Umstellung. Unter Kostendruck wird das Ausmass des Sparzwangs verstanden, dem ein Unternehmen folgen muss. Der Druck

kommt z.B. durch den Wettbewerb mit anderen Unternehmen, Schuldenrückforderungen oder geringeren Budgets zu Stande. Durch die Verwendung von OSS fallen keine Lizenzkosten an. Ebenfalls sind Unterhalts- und Entwicklungskosten tendenziell tiefer als bei proprietärer Software, da z.B. bereits bestehende Softwarekomponenten wiederverwendet werden können (Riehle, 2007, S. 26). Bei proprietärer Software müssen diese Komponenten in der Regel erneut erworben werden. Weiter können aufgrund der grösseren Verhandlungsmacht²⁸ günstigere Dienstleistungsverträge abgeschlossen und so zusätzlich Kosten gespart werden. Eine Umstellung auf OSS entlastet somit das Unternehmen und reduziert dadurch den Kostendruck.

Kostenreduktion durch OSS. Grundsätzlich macht es Sinn auf OSS umzusteigen, sofern dadurch Kosten eingespart werden können. Wichtig ist, dass die kurzfristigen Mehrkosten (z.B. Umstellungskosten) nicht zu hoch sind, da es sonst zu lange dauern würde, diese durch die langfristig tieferen Kosten von OSS zu amortisieren. Eine Migration ist sinnvoller, je grösser die langfristige Kosteneinsparung der entsprechenden OSS ausfällt (Mijinyawa & Abdulwahab, 2014, S. 250). Gemäss den Ergebnissen der Open Source Studie Schweiz (2012, S. 15) beträgt das Kosteneinsparungspotenzial von OSS rund 10-30 Prozent.

Unternehmensgrösse. Mit zunehmender Unternehmensgrösse steigen die Kosten für die Aus- und Weiterbildung der Mitarbeiter. Ebenso sind die Aufwände für den Kauf neuer, oder die Neukonfiguration bestehender Hardware tendenziell grösser, da mehr Hardwarekomponenten angepasst werden müssen. Gemäss Lerner und Tirole (2002, S. 226) ist eine Umstellung unter anderem deshalb weniger sinnvoll, je grösser das Unternehmen ist. Dem gegenüber steht die Aussage Qus, Yangs und Wangs (2011, S. 1002), die behaupten, grössere Unternehmen seien aufgrund der meist besseren finanziellen Lage eher dazu bereit, die nötigen Investitionen für einen Umstieg zu tätigen. Ob eine Umstellung auf OSS sinnvoll ist, hängt auch von der ausgewählten Strategie bezüglich Unabhängigkeit oder Sicherheit ab. Je grösser ein Un-

²⁸ Siehe 6.3

ternehmen ist, desto wichtiger sind verfügbare Ressourcen, sowie ein durchdachter Finanzierungsplan. Die Wahrscheinlichkeit von Synergien profitieren zu können, ist bei Grossunternehmen tendenziell grösser. Während bei kleinen Unternehmen allgemein weniger Kosten anfallen, können grosse Unternehmen die höheren Umstellungskosten durch die Nutzung von Synergien kompensieren. Für kleinere Unternehmen besteht die Herausforderung darin, die Umstellungskosten stemmen zu können. Grosse Unternehmen müssen klären, ob die Umstellungskosten auf mehrere Stellen verteilt, Synergien vorhanden sind und diese genutzt werden können.

Technische Faktoren

Verfügbarkeit von Open Source Alternativen. Je mehr Open Source Alternativen verfügbar sind, desto grösser ist die Wahrscheinlichkeit eine Lösung zu finden, die den Anforderungen eines Unternehmens entspricht. Bevor eine Entscheidung getroffen wird, muss deshalb eine gründliche Analyse der möglichen Softwarelösungen durchgeführt werden. Der Nutzen der ausgewählten Open Source Alternative muss dabei mindestens so hoch sein, wie der Nutzen der bestehenden proprietären Software. Ansonsten kann das Unternehmen keinen Mehrwert generieren und eine Umstellung ist nicht nötig.

Angestrebte Unabhängigkeit. Je unabhängiger ein Unternehmen von Softwareanbietern wie SAP, Oracle oder Microsoft sein will, desto sinnvoller ist eine Umstellung auf OSS. Durch die Verwendung von OSS kann der Anbieter von Softwarelösungen oder Supportdienstleistungen jederzeit gewechselt werden, was es dem Unternehmen erlaubt, sehr flexibel auf Umweltveränderungen zu reagieren. Positiver Nebeneffekt des Wechsels ist das Verhindern eines Vendor Lock-Ins²⁹.

Relativer technischer Level. Der relative technische Level beschreibt das Niveau der verwendeten IT-Komponenten (Software und Hardware) eines Unternehmens verglichen mit Branchenstandards oder direkten Konkurrenten. Verwendet ein Unternehmen minderwertige Software und kann sich eine teurere, bessere Version nicht leisten, macht eine Umstellung auf OSS Sinn. Je grösser der Rückstand auf die Konkurrenz, desto sinnvoller ist ein Umstieg

²⁹ Was ein Vendor Lock-In ist, kann Kapitel 3.2 entnommen werden.

(Lerner & Tirole, 2002, S. 226). Aufgrund ihres Entstehungsprozesses und ihrer Charakteristiken wird OSS häufiger aktualisiert. Dadurch können fehlende Funktionen schneller hinzugefügt werden als bei vergleichbarer proprietärer Software. Aus diesem Grund kann ein Unternehmen die Lücke zur proprietären Konkurrenz relativ schnell schliessen (Lerner & Tirole, 2005b, S. 31) und dadurch Wettbewerbsnachteile bedingt durch einen geringeren technischen Level aufholen.

Risikobereitschaft (offener Quellcode). Die Offenlegung des Quellcodes bringt Chancen und Risiken mit sich³⁰. Die Risikobereitschaft des Managements spielt bei der Entscheidung bezüglich OSS eine wichtige Rolle. Je geringer diese Bereitschaft ist, desto eher wird das bestehende, proprietäre System beibehalten (Qu, Yang, & Wang, 2011, S. 1000). Ob dieser Punkt für oder gegen eine Umstellung auf OSS spricht, hängt von den Präferenzen des Unternehmens ab. Wird der Schutz gegen externe Angriffe wichtiger eingestuft, als beispielsweise die Schaffung des Zugangs zum Wissen der Open Source Community, ist von einer Umstellung auf OSS abzuraten. Für Softwareentwickler bezieht sich die Frage der Risikobereitschaft auf die Offenlegung des Quellcodes selbstentwickelter Software. Durch eine Umstellung auf die OSSE gibt ein Unternehmen sein Copyright auf und verliert damit die Basis seiner bisherigen Einkünfte aus dem Verkauf von Lizenzen. Deshalb stellt eine Migration auf OSSE für Softwareentwickler ein grosses Risiko dar, zu scheitern und somit das gesamte Unternehmen zu gefährden.

Für Softwarenutzer liegt das Risiko in der freien Zugänglichkeit des Quellcodes der nach der Umstellung verwendeten Software. Angreifer haben bei OSS die Möglichkeit, den entsprechenden Quellcode genau zu analysieren und Schwachstellen zu finden, weshalb spätere Angriffe tendenziell gefährlicher sind.

Kompatibilität des Systems. Je grösser die Anzahl eingesetzter Anwendungen verschiedener Anbieter ist, desto wichtiger ist deren reibungsloses Zusammenspiel. Werden viele unterschiedliche Produkte verwendet, ist eine

³⁰ Siehe 0

Umstellung auf OSS sinnvoll, da Open Source Anwendungen oftmals kompatibler zueinander sind, als bei proprietären Produkten (Khanjani & Sulaiman, 2011b, S. 646). Funktioniert die aktuelle, proprietäre Systemlandschaft problemlos, ist eine Umstellung auf OSS bezüglich Kompatibilität nicht zwingend nötig.

Abhängigkeit von Support. Mit dem Wegfallen von Lizenzgebühren verschwinden auch die Ansprüche betreffend Support, die an den Softwareentwickler gestellt werden können. Es gilt also bereits vor der Entscheidung abzuklären, wie die Supportmöglichkeiten für die ausgewählten OSS-Alternativen aussehen (Larsen, Holck, & Pedersen, 2004, S. 2). Ist ein Unternehmen stark von gutem Support abhängig, ist eine Umstellung auf OSS nicht sinnvoll (Khanjani & Sulaiman, 2011b, S. 646). Das Risiko, am Ende eine Software auszuwählen, deren Wartung und Support niemand durchführen kann, ist zu gross. Gemäss Qu, Yang und Wang (2011, S. 999) ist dies einer der Hauptgründe, wieso Unternehmen zögern, auf OSS umzusteigen.

Kontrolle über Software und Daten. Geschäftsrelevante Daten gilt es in jedem Unternehmen zu schützen. Eine Umstellung auf OSS ist sinnvoller, je schlechter Software und Daten kontrolliert und geschützt werden können (Ernst & Young AG, 2011, S. 6). OSS erlaubt eine eigenständige und sichere Verwaltung sensibler Daten (Fuggetta, 2003). Aufgrund des Mehraugenprinzips in der Softwareentwicklung werden mehr Fehler und Sicherheitslücken gefunden und behoben als bei proprietärer Software. Deshalb sind Open Source Anwendungen eher sicher und bieten zudem viel Freiheit betreffend Individualisierbarkeit der einzelnen Funktionen.

Erfahrung mit Migrationen. Je mehr Erfahrungen die Mitarbeiter eines Unternehmens mit Softwaremigrationen haben, desto einfacher kann ein Umstieg auf OSS durchgeführt werden (Ernst & Young AG, 2011, S. 6). Die Verwendung von OSS vergrössert den Pool potentieller Arbeitnehmer signifikant, da nicht nur der Arbeitsmarkt an sich, sondern auch die Open Source Community miteinbezogen werden kann. Expertenwissen kann somit zusätzlich auch extern bezogen werden. Unternehmen müssen vor einer Umstellung auf OSS eruieren, wie gross die Erfahrung ihrer Mitarbeiter ist und ob sie gegebenenfalls von der Erfahrung der Community profitieren können.

7.2.1.2 Softwareentwickler

Die für Softwareentwickler relevanten Faktoren werden, wie in Abbildung 9 ersichtlich ist, in die beiden Kategorien *Produkt & Prozess*, sowie *Markt & Kunden* eingeteilt.

Produkt & Prozess

Qualität der aktuellen Produkte. Je besser die momentan angebotene Software ist, desto geringer ist die Notwendigkeit einer Umstellung auf OSS. Sind die momentan entwickelten Produkte bereits erfolgreich und generieren Gewinne, wäre es kontraproduktiv etwas daran zu ändern. Ist ein Unternehmen mit der Qualität seiner Software unzufrieden, bietet die OSSE laut Raymond (2001; zitiert in Hauge, Ayala & Conradi, 2010, S. 1135) nicht nur günstigere Produkte, sondern auch schnellere Fehlerbehebungen und Weiterentwicklungen.

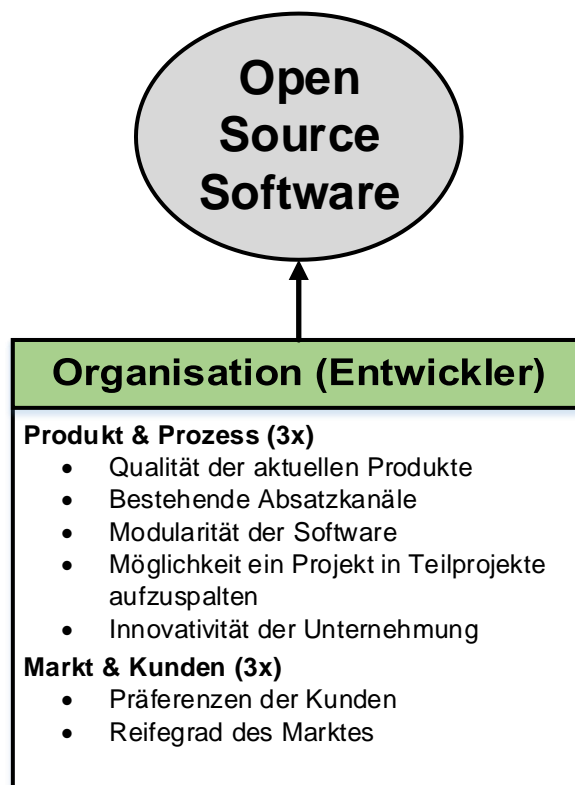


Abbildung 9 Organisationsebene (Entwickler)

Dies ist ein Ausschnitt aus dem erstellten Framework, der die Softwareentwicklerperspektive im Detail zeigt.

Bestehende Absatzkanäle. Will ein Unternehmen durch die Umstellung auf OSS neue Märkte und Kundengruppen erschliessen, ist ein bereits bestehendes, gut ausgebautes Absatznetz hilfreich. Durch das Absatznetz kann der Bekanntheitsgrad neuer Produkte schnell gesteigert werden. Allerdings kann eine Migration auf OSS auch zur Destabilisierung eines erfolgreichen Systems führen und das System hat eventuell nicht mehr die gleiche Wirkung, wie vor der Migration. Durch die Verwendung der OSSE erfolgt die Verteilung nicht nur über firmeneigene Absatzkanäle, sondern auch über diejenigen der Open Source Community. Die OSSE bietet Unternehmen so kostenlos zusätzliche Absatzkanäle. Ein Unternehmen muss evaluieren, ob diese gewonnenen Absatzkanäle für seine Geschäftstätigkeit relevant sind und ob eine Umstellung auf OSS das bestehende System destabilisieren würde.

Modularität der Software. Software ist umso besser für eine Umstellung auf OSSE geeignet, je modularer sie aufgebaut ist (Lerner & Tirole, 2002, S. 220). Durch die Modularität ist es möglich, den Quellcode einzelner Softwareteile zu veröffentlichen und so in Zusammenarbeit mit der Open Source Community weiterzuentwickeln. Laut Bitzer und Schröder (2005, S. 6) müssen die einzelnen Softwareteile zudem möglichst unabhängig voneinander sein. So können negative Einflüsse auf andere Softwareteile verhindert werden.

Möglichkeit ein Projekt in Teilprojekte aufzuspalten. Je leichter sich ein Projekt in Teilprojekte aufzuspalten lässt, desto besser eignet sich die OSSE für dessen Durchführung (Lerner & Tirole, 2002, S. 220). Durch die Aufspaltung können einzelne, möglichst unabhängige Teile in Zusammenarbeit mit der Open Source Community entwickelt und so von deren Wissen und Fähigkeiten profitiert werden. Unternehmen, die ihre Projekte nicht aufteilen können oder wollen, profitieren weniger von einer Umstellung.

Innovativität der Unternehmung. Spielt die Innovativität in einem Unternehmen eine wichtige Rolle und möchte es sich im Markt als innovationsfördernd positionieren, ist eine Umstellung auf Open Source sinnvoll. In der OSSE kann diese Position schneller und einfacher etabliert werden, da dort

der Austausch zwischen dem Unternehmen und der Entwicklercommunity wesentlich ausgeprägter stattfindet³¹. Proprietäre Software hingegen wird im Extremfall erstmals veröffentlicht, wenn die Entwicklung bereits abgeschlossen ist und somit keine Anpassungen oder Verbesserungen mehr vorgenommen werden können. Dadurch liegt in der klassischen Softwareentwicklung oft eine gewisse zeitliche Verzögerung vor, bis neue Ideen in bestehende Software eingebaut werden, was bei OSS normalerweise nicht der Fall ist.

Markt & Kunden

Präferenzen der Kunden. Ob sich eine Umstellung auf die OSSE für ein Unternehmen lohnt, hängt auch von den Präferenzen und Bedürfnissen seiner Kunden ab. Je höher die Sicherheitsanforderungen der Kunden an die verwendete Software sind, desto mehr lohnt es sich, OSS zu entwickeln. Legt die Zielgruppe Wert auf eine ansprechende Benutzeroberfläche, ist es sinnvoller, weiterhin proprietäre Software zu entwickeln. Bei proprietären Projekten sind oftmals mehr finanzielle Mittel vorhanden, weshalb der Fokus z.B. vermehrt auf das Design der Benutzeroberfläche gelegt werden kann. Bei OSS steht meist die Funktionalität im Vordergrund und beim Design werden deshalb eher Abstriche gemacht.

Reifegrad des Marktes. Wird die Einführung eines neuen Produktes in einem bestehenden Markt angestrebt, stellt OSS tendenziell die einfachere Lösung dar. Mit der kostenlosen OSS ist es möglich, sogar in Märkte einzusteigen, die bereits von einem grossen Unternehmen dominiert werden (Riehle, 2007, S. 27). Da bei OSS zu Beginn nur minimale finanzielle Investitionen getätigt werden müssen, liegt die Hemmschwelle eines Unternehmens, in einen Markt einzusteigen, tiefer. Dadurch können Unternehmen einfacher neue Produkte in bestehende Märkte einführen. Vor der Umstellung auf OSS ist es dennoch zwingend nötig, eine Marktanalyse durchzuführen. Obwohl die finan-

³¹ In der klassischen Softwareentwicklung werden unfertige Softwareversionen von einer ausgewählten Gruppe getestet. Im Gegensatz zur OSSE haben die Testpersonen bei proprietären Projekten keinen Zugang zum Quellcode.

ziellen Einbußen bei einem Scheitern geringer sind als bei proprietärer Software, verliert ein Unternehmen trotzdem Zeit, Aufwand und dadurch auch Geld.

7.2.2 Personal

Anhand der in Kapitel 7.2.1 genannten Faktoren wird überprüft, ob die ökonomischen Rahmenbedingungen für eine Migration auf OSS gegeben sind. Zusätzlich zu diesen Faktoren muss geklärt werden, ob auch die Mitarbeiter und das Management eines Unternehmens bereit sind, auf OSS umzusteigen. Das folgende Kapitel behandelt die in Abbildung 10 genannten Faktoren der Personalebene des Modells aus Abbildung 6. Dabei wird zwischen den Kategorien *Personalführung & Management* und *Individuum* unterschieden.

Personalführung & Management

Unterstützung durch das Management. Strategische Entscheidungen werden normalerweise im Management getroffen. Je mehr Open Source dort unterstützt wird, desto einfacher ist eine entsprechende Migration. Der Support durch die Geschäftsführung stellt demnach ein Schlüsselkriterium dar, dessen Fehlen in zu einem sofortigen Abbruch der Analyse führt.

Qualität der Führungskräfte. Damit die Zusammenarbeit mit einer Open Source Community funktioniert, müssen die involvierten Führungskräfte dazu in der Lage sein, die involvierten Entwickler zur Mitarbeit im Projekt zu motivieren. Ansonsten droht ein Forking³². Wichtig ist auch, dass eine Zielkonvergenz glaubhaft vermittelt werden kann. Nur wenn alle Beteiligten das gleiche Ziel anstreben, ist eine erfolgreiche Zusammenarbeit möglich (Lerner & Tirole, 2002, S. 222). Zielkonflikte führen zu Verzögerungen oder im schlimmsten Fall zum Auseinanderbrechen des Teams. Da Open Source seinen Wert aus der intensiven Zusammenarbeit vieler verschiedener Entwickler schöpft, ist eine kompetente Projektleitung, welche die Arbeit in der Gruppe leitet und fördert, essentiell.

³² Siehe 3.2

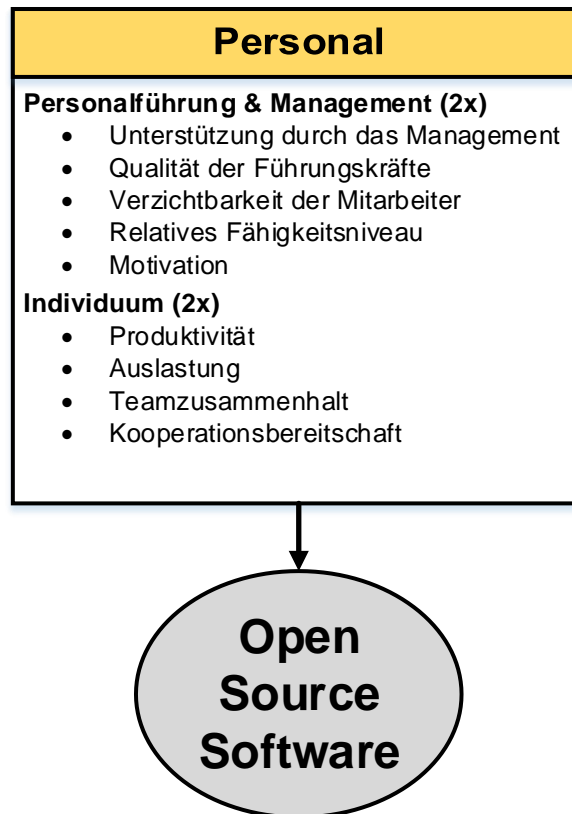


Abbildung 10 Personalebene

Dies ist ein Ausschnitt aus dem erstellten Framework, der die Personalebene im Detail zeigt.

Verzichtbarkeit der Mitarbeiter. Eine Umstellung auf die OSSE erhöht die wahrgenommene Transparenz eines Unternehmens. Gleichzeitig steigt aber das Risiko Mitarbeiter zu verlieren. Im Gegenzug erhöht die Erweiterung des Arbeitnehmerpools die Chance neue, bessere Entwickler zu finden. Besitzt das Unternehmen viele unverzichtbare Mitarbeiter, ist eine Softwaremigration nicht sinnvoll, da bei einer Abwerbung durch andere Unternehmen zu viel (produkt-)spezifisches Wissen verloren ginge. Verfügen die angestellten Entwickler jedoch über kein oder wenig produktspezifisches Wissen, erhöht sich die Attraktivität einer Umstellung auf OSS.

Relatives Fähigkeitsniveau. Das relative Fähigkeitsniveau der Mitarbeiter beschreibt deren Bildungs-, Fähigkeits- und Erfahrungslevel verglichen mit dem Branchendurchschnitt oder der Konkurrenz. Je grösser dabei der Unterschied zum Level in der Open Source Community ist, desto extremer fällt die Bewertung dieses Kriteriums aus. Haben die Mitarbeiter einen höheren Level

als die Community, macht eine Migration nicht zwingend Sinn. Ist das Fähigkeitsniveau der eigenen Mitarbeiter aber tiefer als dasjenige der Community, wäre eine Umstellung sinnvoll, da das Unternehmen und seine Mitarbeiter von den Kenntnissen der Community profitieren können (von Krogh et al., 2012, S. 670). Hierbei spielen weder Alter, Geschlecht, noch andere soziodemografische Variablen eine Rolle, da ausschliesslich die Leistungsfähigkeit der Mitarbeiter betrachtet wird.

Motivation. Bedingt durch die mit einem Umstieg auf OSS einhergehende Öffnung des Unternehmens gegen aussen werden auch die angestellten Entwickler sichtbarer. Eine bestmögliche Positionierung auf dem Arbeitsmarkt liegt demnach auch in ihrem Interesse. Leisten sie bessere Arbeit, steigt ihr Wert und dadurch auch der des Unternehmens. Die Öffnung gibt dem Unternehmen dadurch eine zusätzliche Macht über die angestellten Entwickler. Durch die Zusammenarbeit mit der Community sehen Unternehmen anhand der Qualität des geschriebenen Quellcodes direkt, wie fähig ein Entwickler ist. Sollten grosse Unterschiede festgestellt werden, können Mitarbeiter ausgetauscht werden. Dieser Leistungsdruck spornt Entwickler zusätzlich an, bessere Arbeit zu leisten. Von diesem Standpunkt aus betrachtet hat eine Umstellung auf OSS positive Auswirkungen für das Unternehmen. Insbesondere die Leistung eher unmotivierter Mitarbeiter kann durch einen Umstieg verbessert werden. Allerdings muss die Möglichkeit der Einsicht in die Arbeit einzelner Mitarbeiter verantwortungsvoll und ethisch korrekt behandelt werden.

Individuum

Produktivität. In einigen Studien konnte gezeigt werden, dass sich die Arbeitsleistung unternehmensinterner Softwareentwickler durch die Öffnung eines Unternehmens eher erhöht, als gesenkt hat (Lerner & Tirole, 2005a, S. 103). Dieses Resultat lässt sich dadurch erklären, dass den befragten Programmierern die Mitarbeit an „coolen“ Open Source Projekten mehr Spass gemacht hat, als routinemässige, „normale“ Tätigkeiten. Je grösser der Unterschied zwischen momentaner und potentiell möglicher Produktivität ist, desto eher lohnt es sich für ein Unternehmen, auf OSS umzustellen.

Auslastung. Sind Mitarbeiter nicht genügend ausgelastet, verliert das Unternehmen durch ihre Untätigkeit Geld. Durch eine Umstellung auf OSS würden sie zwar nicht automatisch ausgelastet, könnten aber in der Community ihre Fähigkeiten erweitern und sich neues Wissen aneignen, wovon auch das Unternehmen profitiert (Lerner & Tirole, 2005a). Je geringer die Auslastung der Mitarbeiter, desto sinnvoller ist eine Migration auf OSS.

Teamzusammenhalt. Unterstützen und helfen sich Teammitglieder gegenseitig, können Probleme wesentlich schneller und kostengünstiger gelöst werden. Dabei ist entscheidend, wie gut der Wissenstransfer von im Umgang mit Open Source erfahrenen zu weniger erfahrenen Mitarbeitern funktioniert. Je besser der Zusammenhalt im Team ist, desto einfacher fällt die Umstellung auf OSS und desto mehr Probleme können gelöst werden, ohne dass das Management oder die Migrationsverantwortlichen beigezogen werden müssen.

Kooperationsbereitschaft. Zur Kooperationsbereitschaft gehört beispielsweise die Haltung der Angestellten gegenüber OSS. Würde eine mögliche Umstellung auf Open Source von den Mitarbeitern boykottiert, da diese Angst um ihren Job haben oder nicht gewillt sind, etwas neues zu lernen, hätte dies einen negativen Einfluss auf die Entscheidung als Ganzes. Mitarbeiter könnten zwar falls nötig ersetzt werden, es würden aber Zusatzkosten anfallen, die nur bei einer geringen Anzahl Mitarbeiter finanzierbar wären. Bevor eine Migration auf OSS vollzogen wird, sollten Unternehmen deshalb die Kooperationsbereitschaft und allgemeine Haltung ihrer Mitarbeiter bezüglich OSS erfassen.

7.2.3 Umwelt

Mittels der in den beiden vorherigen Kapiteln definierten Faktoren kann untersucht werden, ob ein Unternehmen als Ganzes eine Migration auf OSS erlauben würde. Selbst bei einem positiven Fazit gibt es allerdings eine dritte Analyseebene. Ein Unternehmen befindet sich in einem ständigen Austausch mit seiner Umwelt. Diese Umweltebene wird, wie Abbildung 11 entnommen werden kann, in *branchenspezifische* und *juristische & wirtschaftliche* Faktoren

eingeteilt. Sowohl der Softwaremarkt, als auch die für das Unternehmen relevanten Kunden wurden bereits im Kapitel über Softwareentwickler behandelt³³.

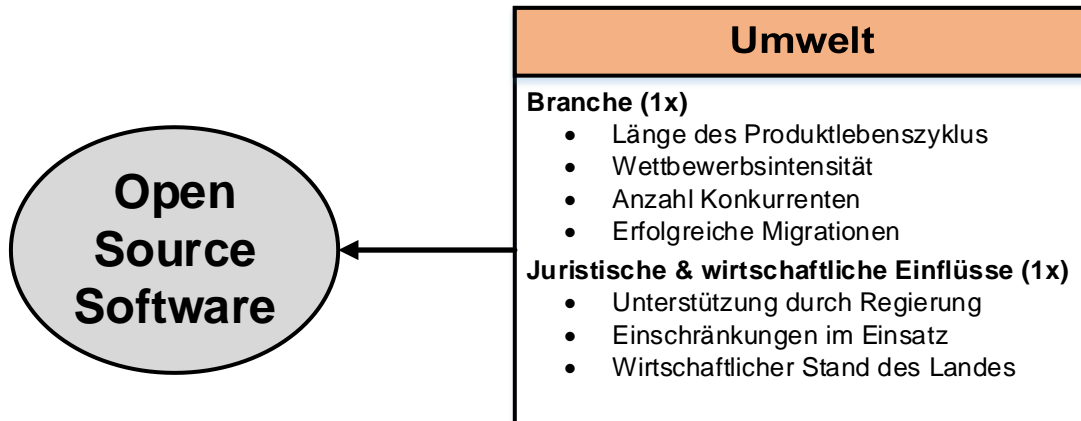


Abbildung 11 Umweltebene

Dies ist ein Ausschnitt aus dem erstellten Framework, der die Umweltebene im Detail zeigt.

Branche

Länge des Produktlebenszyklus. Produktlebenszyklen werden immer kürzer und neue Produkte müssen immer schneller entwickelt werden. Je kürzer die Produktlebenszyklen in der entsprechenden Branche sind, desto eher ist eine Umstellung auf OSS sinnvoll. Die grössere Flexibilität und kürzere Entwicklungsdauer von OSS sind insbesondere von Vorteil, wenn schnelle Anpassungen verlangt werden. Durch diese Vorteile von Open Source können gemäss Raymond (2001) Wettbewerbsvorteile gegenüber proprietären Konkurrenten erzeugt werden. Die OSSE bietet Unternehmen eine Möglichkeit, auf die kürzer werdenden Produktlebenszyklen zu reagieren und sich dadurch von der proprietären Konkurrenz abzuheben.

Wettbewerbsintensität. In einem Markt, indem viele Unternehmen um Marktanteile kämpfen, steigt das Risiko durch die Offenlegung von Quellcode wichtige Wettbewerbsvorteile zu verlieren. Fehler werden eher ausgenutzt, wenn eine hohe Wettbewerbsintensität vorliegt, da es viele direkte Konkurrenten gibt, die den Vorteil anderer Unternehmen kopieren oder verringern wollen.

³³ Siehe 7.2.1.2

Baut ein Unternehmen auf einem bestimmten Wettbewerbsvorteil durch bestehende Produkte auf, ist die Umstellung auf OSS mit einem zu grossen Risiko verbunden und lohnt sich deshalb nicht.

Anzahl Konkurrenten. Die Auswirkungen einer Entscheidung für oder gegen Open Source hängen von der Anzahl direkter Konkurrenten ab, die im Zielmarkt vorhanden ist. Gibt es im entsprechenden Markt bereits viele Unternehmen, die OSS anwenden oder anbieten, macht eine Umstellung wenig Sinn. Entscheidend ist, wie gut sich ein Unternehmen mit den angebotenen Produkten und Dienstleistungen von der direkten Konkurrenz abheben kann. Eine mögliche Verbesserung der Position im Markt spricht für eine Umstellung auf OSS. Bleibt die Situation gleich oder kann eine Veränderung nicht eingeschätzt werden, müssen andere Kriterien wie das Geschäftsmodell oder die Unterstützung durch das Management betrachtet werden. Sprechen strategische Faktoren für einen Umstieg, kann die Konkurrenzsituation analog bewertet werden.

Erfolgreiche Migrationen. Je mehr Beispiele erfolgreicher Umstellungen vorhanden sind, desto leichter fällt es, sich für Open Source zu entscheiden. Um von den Erfahrungen anderer profitieren zu können, ist eine Analyse der Erfolgs- und Misserfolgskriterien der entsprechenden Projekte nötig. Die daraus gewonnenen Erkenntnisse sollen anschliessend in die eigenen Entscheidungen eingebaut werden. Sind in der betrachteten Branche bereits viele erfolgreiche Migrationen durchgeführt worden, kann das Vorgehen aus ähnlichen Projekten kopiert und an das eigene Unternehmen angepasst werden. Dies verringert die Fehlerwahrscheinlichkeit und erhöht die Erfolgsquote des eigenen Projektes. Dadurch erhöht sich die Motivation, eine Umstellung in Erwägung zu ziehen.

Juristische & wirtschaftliche Einflüsse

Unterstützung durch Regierung. Je grösser die Unterstützung durch die Regierung ist, desto sinnvoller ist eine Umstellung auf OSS (Mijinyawa & Abdulwahab, 2014, S. 250). Diese Unterstützung beinhaltet beispielsweise Gesetze, welche die Verwendung von OSS verlangen, oder die Zahlung eines bestimmten Geldbetrages, wenn OSS verwendet wird (ähnlich wie Subventionen). Auf die Rolle der Regierung wird in Kapitel 8 genauer eingegangen.

Einschränkungen im Einsatz. Liegen für den Einsatz von OSS viele Einschränkungen vor, macht eine Migration auf OSS wenig Sinn. Mögliche Einschränkungen sind ungeklärte Haftungsfragen bezüglich Lizenzen oder ein Mangel an Supportdienstleistern (Open Source Studie Schweiz, 2012, S. 5). Solche Einschränkungen steigern das Misstrauen gegenüber OSS. Gibt es zu viele Restriktionen, kann ein Unternehmen möglicherweise nicht von den Vorteilen von OSS profitieren und stellt deshalb nicht um.

Wirtschaftlicher Stand des Landes. Für Unternehmen in Entwicklungsländern ist eine Umstellung auf OSS attraktiv, da sie so trotz oft beschränkter Mittel den Rückstand gegenüber Industrienationen verringern können³⁴ (Qu, Yang, & Wang, 2011, S. 999). In reicheren Ländern sollte eine Umstellung auf OSS einfacher sein, da mehr finanzielle Mittel und Knowhow vorhanden sind. Zusätzlich ist ein Geschäftsmodell, das auf dem Anbieten von Dienstleistungen beruht, weniger anfällig gegen Raubkopien, was bei OSS der Fall ist.

In diesem Kapitel wurden die Entscheidungskriterien des erstellten Frameworks genauer betrachtet. In einem nächsten Schritt werden diese Faktoren bewertbar gemacht. Dazu werden in den folgenden Kapiteln verschiedene Erhebungsmethoden beschrieben, sowie das Bewertungsverfahren erläutert, um dann abschliessend Handlungsempfehlungen abgeben zu können, wie ein Unternehmen seine Position bezüglich Open Source verbessern kann.

7.3 Erhebung der Kriterien

Die Ausprägung der im Framework enthaltenen Faktoren kann auf verschiedene Arten erfasst werden. So können Interviews oder Telefongespräche mit ausgewählten oder allen Mitarbeitern durchgeführt werden. Auch ist eine Evaluation in Form eines selbstverfassten, offenen Berichtes durch die Mitarbeiter denkbar. Dabei können diese z.B. ihre Haltung gegenüber Open Source und einer Migration auf OSS beschreiben. Eine ökonomische, effiziente und objektive Erfassungsmethode stellt ein Fragebogen dar.

³⁴ Eine Begründung kann im Punkt *relativer technischer Level* gefunden werden.

Um die im zuvor beschriebenen Framework genannten Faktoren zu erheben, wurden jeweils zwei Fragen pro Faktor formuliert. Anschliessend wurden die erstellten Fragen zu einem Fragebogen zusammengestellt und an Mitarbeiter des BIT versendet. Aufgrund der Rückmeldungen wurden kritische, unklare oder ambivalente Fragen umformuliert oder ersetzt, was die Validität des Fragebogens erhöht. Dennoch können im Rahmen dieser Arbeit weder die Validität, noch die Reliabilität der Fragen genügend geprüft und deshalb nicht garantiert werden. Aus diesem Grund werden die Fragen in Form eines Fragekataloges in Anhang A.3 zur Verfügung gestellt. Die aufgelisteten Fragen erheben keinen Anspruch auf Vollständigkeit und sind jeweils an die Besonderheiten des zu untersuchenden Unternehmens anzupassen. Bevor die im Katalog enthaltenen Fragen als valides Erfassungsmittel gelten können, müssen sie zuvor mehrfach überprüft, normiert und die Resultate statistisch analysiert werden (Moosbrugger & Kelava, 2012). In dieser Analyse muss überprüft werden, ob die erstellten Fragen die vier definierten Entscheidungsebenen erfassen, ob möglicherweise weitere oder andere Ebenen vorhanden sind und ob die Fragen genügend Varianz aufdecken (Eid, Gollwitzer, & Schmitt, 2011).

7.4 Bewertung

Um die Ergebnisse aus der Anwendung des Frameworks vergleichbar zu machen, ist eine Bewertung der aufgelisteten Faktoren notwendig. Zu diesem Zweck werden die zuvor beschriebenen Faktoren mit einer Punkteskala verknüpft. Sofern nichts anderes angegeben ist, wird in diesem Kapitel vom Entscheidungsprozess eines Softwareentwicklers ausgegangen, um alle Ebenen des Frameworks demonstrieren zu können.

7.4.1 Punktevergabe

Den Faktoren aus Abbildung 7 können +/-1 oder 0 Punkte zugeordnet werden. Die Vergabe von +1 Punkt bedeutet, dass die Ausprägung des betrachteten Faktors für einen Umstieg auf OSS spricht. Ein negativer Wert von -1 hingegen spricht gegen eine Migration. Kann nicht genau bestimmt werden, ob der Einfluss positiv oder negativ ist, oder hat ein Faktor keinen Einfluss auf die Ent-

scheidung, können 0 Punkte vergeben werden. Existieren in einem Unternehmen z.B. keinerlei Erfahrungen mit Softwaremigrationen, kann dieser Faktor nicht bewertet werden und wird mit dem Wert 0 versehen.

Um die drei aus Abbildung 6 bekannten Ebenen voneinander unterscheiden zu können, wurde jeder Kategorie ein Multiplikator zugeordnet. Softwarenutzer und -entwickler erhalten dabei denselben Wert, da beide Bestandteil der Organisationsebene sind. Entsprechend der Wichtigkeit für das Unternehmen erhält die Organisationsebene den Multiplikator drei, die Personalebene wird zweifach verrechnet und die Umweltebene zählt einfach. Diese Einteilung wurde gewählt, da aus wirtschaftlicher Sicht der erfolgreiche Fortbestand des Unternehmens oberste Priorität hat. Da die Mitarbeiter eines Unternehmens austauschbar sind, wurde die Personalebene verglichen mit der Organisationsebene tiefer bewertet. Gleichzeitig ist sie aber wichtiger als die Umweltebene, da sie vom Unternehmen direkt beeinflusst werden kann. Die Umweltfaktoren erhalten am wenigsten Punkte, da diese in Anlehnung an die Zwei-Faktor-Theorie Herzbergs, Mausners und Snydermans (1959) als Hygienefaktoren betrachtet werden können. Sie sind zwar notwendig, aber nicht ausreichend, um darauf basierend eine Entscheidung zu treffen und tragen lediglich zur Schaffung der benötigten Rahmenbedingungen bei. Bei einer Entscheidung für oder gegen Open Source spielen sie jedoch nur eine untergeordnete Rolle.

7.4.2 Gruppen

Die oben beschriebene Methode der Punktevergabe erlaubt eine Maximalpunktzahl von +/- 85 Punkten. Die Ergebnisse werden anhand der erreichten Punktezahl in drei Gruppen eingeteilt. Die Gruppen stehen dabei für den Grad der Umsetzbarkeit der Migration. Eine bedenkenlose Umstellung auf OSS ist dann gegeben, wenn folgende zwei Bedingungen erfüllt sind:

1. Die erreichte Punktezahl ist ≥ 40 .
2. Mindestens die Hälfte der Faktoren aus der Organisationsebene wurde positiv bewertet.

Tiefes Ergebnis (-85 bis 0 Punkte)

Erreicht ein Unternehmen Null oder weniger Punkte, ist eine Umstellung auf Open Source nicht empfehlenswert, da zu viele Kernfaktoren (mit 3-fachem Multiplikator) negativ bewertet wurden. Ein tiefes Ergebnis bedeutet nicht zwingend, dass ein Unternehmen in Schwierigkeiten steckt. Es kann auch auf ein gut funktionierendes, aktuelles IT-System hindeuten, weshalb eine Migration auf OSS nicht nötig ist. Unter dem Motto *never change a winning team* könnte eine Umstellung sogar kontraproduktiv sein, da das Risiko einer Verschlechterung der Geschäftsergebnisse besteht. Eine Migration auf OSS würde zumindest zum Zeitpunkt der Evaluation keinen Sinn machen. Falls eine Umstellung dennoch gewünscht wird, kann die Untersuchung zu einem späteren Zeitpunkt wiederholt werden, nachdem die aufgedeckten Schwachstellen korrigiert wurden.

Mittleres Ergebnis (1 bis 39 Punkte)

Der mittlere Punktbereich zwischen 1 und 39 Punkten wird in die zwei Unterbereiche *Eher Nein* und *Eher Ja* aufgeteilt. Fällt das Ergebnis in den Bereich zwischen 1 und 19 Punkten, würde eine Migration auf OSS zwar grundsätzlich Sinn machen, allerdings müssen zuerst einige organisatorische Anpassungen gemacht werden. Ist zusätzlich zum Nicht-Erreichen der 40-Punkte-Marke die zweite oben genannte Bedingung für einen bedenkenlosen Umstieg nicht erfüllt, wird von einer Umstellung abgeraten.

Für den Bereich zwischen 20 und 39 Punkten sind die vor der Umstellung durchzuführenden Anpassungen vergleichsweise geringer, weshalb eine Umstellung eher Sinn macht. Für den Fall, dass trotz des Nicht-Erreichens der 40-Punkte-Marke mehr als die Hälfte der organisatorischen Faktoren positiv bewertet wurden, ist demnach eine Umstellung mit parallelem Korrigieren der negativ bewerteten Faktoren zu empfehlen.

Hohes Ergebnis (40 bis 85 Punkte)

Unternehmen, die ein Ergebnis von 40 oder mehr Punkten erreichen, sind für eine Umstellung auf OSS sehr gut geeignet. Allerdings muss beachtet werden, dass bei einer Punktezahl von 40 theoretisch auch lediglich fünf der 20 aufgelisteten Kernfaktoren positiv bewertet wurden, während auf der Personal- und der Umweltebene jeweils die maximale Punktezahl erreicht wurde. Somit ist

es auch bei einem hohen Punkteergebnis möglich, die erste Bedingung für eine bedenkenlose Umstellung auf OSS nicht zu erfüllen. Zusätzlich zum Erreichen der Punktezahl von 40 müssen also trotz hoher Totalpunktzahl mindestens die Hälfte der Faktoren aus der Organisationsebene positiv bewertet werden, damit eine Migration empfohlen werden kann.

7.4.3 Softwarenutzer

Wird das Framework auf ein Unternehmen angewendet, das OSS lediglich nutzen, jedoch nicht (weiter) entwickeln will, reduziert sich die Maximalpunktzahl auf 64. Dies wirkt sich wie folgt auf die drei Wertebereiche aus:

- Die tiefste Kategorie bleibt unverändert. Bei einem Ergebnis von ≤ 0 ist eine Migration in keiner Situation empfehlenswert.
- Der Bereich der mittleren Kategorie verkleinert sich auf die Punktwerte von 1 bis 24 mit den beiden Unterkategorien *Eher Nein* von 1 bis 12 und *Eher Ja* von 13 bis 24 Punkten.
- Die Grenze zur höchsten Bewertung wird ab einer Punktezahl von 25 erreicht.

7.5 Handlungsempfehlungen für kritische Faktoren

Nachdem eine Bewertung anhand der im Framework erwähnten Faktoren durchgeführt wurde, gilt es die Schwachstellen des Unternehmens zu erkennen und zu korrigieren. In diesem Kapitel werden Handlungsempfehlungen abgegeben, wie negativ bewertete Punkte verbessert werden können. Die zu Beginn von Kapitel 7 beschriebene Hierarchie kommt auch hier zum Tragen, da nicht alle drei Ebenen gleich gut beeinflusst werden können.

Die Kategorie der ökonomischen Faktoren lässt sich kurzfristig nur bedingt verändern, da es sich grösstenteils um strategische Entscheide handelt. Ist die finanzielle Lage des Unternehmens angespannt oder sind die Kosten zu hoch, kommen eher allgemeine Schritte in Frage. Es können Investoren gesucht werden, die eine OSS-Migration unterstützen und fördern wollen. Wei-

ter müssen Kostentreiber aufgespürt und reduziert, oder zusätzliche Einkommensströme generiert werden. Eine Analyse der *Total Cost of Ownership*³⁵ bietet sich an, da so das langfristige Kosteneinsparungspotenzial von OSS bestimmt werden kann.

Auf der technischen Seite ist eine Einflussnahme bedeutend einfacher. Sind zu wenig oder die falschen Open Source Alternativen vorhanden, gibt es zwei Verbesserungsmöglichkeiten. Zum einen kann abgewartet werden, bis mehr oder bessere Alternativen verfügbar sind. Zum anderen kann die Entwicklung der entsprechenden Anwendung finanziell und/oder durch Knowhow unterstützt werden, um deren Fertigstellung zu beschleunigen.

Die Abhängigkeit von gutem Support lässt sich reduzieren, indem entsprechende Fachkräfte eingestellt werden. Dies senkt längerfristig ebenfalls den Kostendruck, da interne Lösungen in den meisten Fällen günstiger sind als extern eingekaufte. Auf dem gleichen Weg kann auch die interne Erfahrung im Umgang mit Migrationen vergrößert werden.

Ob eine Anpassung der eigenen Produkte sinnvoll ist, hängt sehr mit der grundsätzlichen Einstellung gegenüber OSS zusammen. Ist ein Umstieg auf Open Source bereits geplant, macht eine Verbesserung bestehender Produkte nach der alten Entwicklungsmethode keinen Sinn mehr. Unabhängig davon kann auch die Struktur der Produkte angepasst werden. Eine modulärere Struktur würde es selbst bei einem Verbleib bei proprietären Produkten ermöglichen, gewisse Softwareteile zu veröffentlichen und so von der Open Source Community und deren Wissen zu profitieren.

Befindet sich ein Unternehmen in einer guten finanziellen Lage, ist das Investieren in Absatzkanäle ebenfalls hilfreich. Davon würde das gesamte Unternehmen profitieren, da so längerfristig Kosten gespart werden können. Eine Umstellung auf OSS würde so wieder an Attraktivität gewinnen.

Die markt- und kundenbezogenen Faktoren lassen sich kaum beeinflussen, da sie unabhängig vom Unternehmen existieren. Als Verbesserungs-

³⁵ Unter den Total Cost of Ownership sammeln sich alle Kosten an, die ein Produkt über die Gesamtnutzungsdauer, einschließlich Kauf und Entsorgung, verursacht (Vogel, 2015).

möglichkeiten bieten sich das Steigern der Marketingaktivitäten im entsprechenden Markt, oder eine Neuorientierung bezüglich Zielmarkt und Zielkundengruppe an.

Werden auf der Ebene des Managements Defizite festgestellt, kann versucht werden, diese mittels des Besuchs bestimmter Seminare oder Kurse zu korrigieren. Eine etwas drastischere Lösung wäre die Umstrukturierung des Managements, respektive die Ergänzung des Teams durch neue Mitarbeiter, sowie der Austausch nicht zufriedenstellender Mitglieder.

Um die Motivation der Mitarbeiter allgemein zu erhöhen und diese in Richtung Open Source zu lenken, können Ausflüge oder Workshops organisiert werden, bei denen die Belegschaft an das Thema *Open Source* herangeführt wird. Dadurch kann auch eine Verbesserung des Teamzusammenhalts erreicht werden. Eine bessere Stimmung zwischen Mitarbeitern und Management, sowie den Mitarbeitern untereinander wirkt sich ebenfalls positiv auf die Kooperationsbereitschaft aus, da durch eine umfassende Information viele Unsicherheiten und daraus entstehende Abneigungen eliminiert werden könnten.

Auf der Individualebene ist es wichtig, die Auslastung zu verteilen. Dadurch könnte das Fähigkeitsniveau der Mitarbeiter ausgeglichen werden, was sich positiv auf die Produktivität und Motivation auswirkt.

Wie Abbildung 6 zeigt, ist die Umweltebene am schwierigsten zu beeinflussen. Sowohl die Branche, als auch die rechtlichen Rahmenbedingungen können praktisch nur verändert werden, wenn ein Unternehmen z.B. an Tagungen aktiv für eine Gesetzesänderung oder ein neues Gesetz wirbt und versucht Befürworter der Idee zu finden. Selbst wenn dies gelingt, könnte es unter Umständen Jahre dauern, bis ein Gesetz neu formuliert oder angepasst wird. Um zu verhindern, dass diese Starre die Entscheidung betreffend einer Open Source Migration zu sehr beeinflusst, wurden die Umweltfaktoren entsprechend tief gewichtet (Multiplikator 1).

8 Staatliche Förderung von Open Source

Nachdem in Kapitel 7 das erstellte Open Source Framework beschrieben wurde, soll nun die Rolle der Regierung genauer betrachtet werden. Dazu wird auf die Frage eingegangen, ob eine Landesregierung in den Softwaremarkt eingreifen soll und bis zu welchem Grad diese Unterstützung sinnvoll ist. Danach wird am Beispiel des *Bundesamtes für Informatik und Telekommunikation* aufgezeigt, wie Open Source in der Schweizer Bundesverwaltung verwendet wird.

8.1 Staatliche Interventionen in der Softwareindustrie

Eine Regierung kann OSS auf zwei verschiedene Arten unterstützen oder deren Entwicklung beeinflussen. Sie kann selbst OSS beschaffen und so die Nachfrage im Markt ankurbeln. Eine andere Möglichkeit, mit wesentlich mehr Optionen, wäre es, mittels Gesetzen, Bestimmungen, etc. die Rahmenbedingungen im Markt zu verändern. Im folgenden Kapitel wird deshalb die zweite Vorgehensweise genauer beschrieben.

8.1.1 Argumente für eine Intervention

Eine staatliche Intervention in der Softwareindustrie kann sowohl auf die im Markt tätigen Unternehmen, als auch auf die intervenierende staatliche Verwaltung einen entscheidenden Einfluss haben.

Comino, Manenti und Rossi (2011, S. 91) listen mehrere Argumente auf, wieso eine Intervention sinnvoll ist. So erwähnen sie, dass eine Intervention in der Softwareindustrie dazu führt, dass sich die Innovationstätigkeit erhöht. Gemäss Bessen und Maskin (2009; zitiert in Comino, Manenti & Rossi, 2011) sollte die Regierung die Entwicklung neuer Software fördern, indem sie deren Einschränkung durch bestehende Patente verhindert. Offener und frei zugänglicher Quellcode, wie dies bei OSS der Fall ist, erleichtert die Softwareentwicklung erheblich.

Weiter behaupten Comino, Manenti und Rossi (2011), dass eine Intervention den Wettbewerb in der Softwareindustrie erhöhen würde, was auch von Evans und Reddy (2003, S. 387) bestätigt wird. Da die Verwendung von OSS nicht an hohe Kosten gebunden ist, wie dies bei proprietärer Software

der Fall ist, sind die Einstiegsbarrieren in einen Markt wesentlich tiefer. Unternehmen können ein beliebiges, kostenloses Open Source Produkt wählen und dazu beispielsweise Supportdienstleistungen anbieten. Dies erhöht die Wahrscheinlichkeit, dass der Konkurrenzkampf in einem bestehenden Markt grösser wird.

Als letzten positiven Effekt einer staatlichen Intervention in der Softwareindustrie nennen Comino, Manenti und Rossi (2011) die fördernde Wirkung für eGovernment Bestrebungen. Durch die Reduktion der Installations- und Unterhaltskosten, die im OSSE-Prozess erfolgt, kann sich eine eGovernment-Software wesentlich schneller verbreiten, weshalb es ihr leichter fällt, sich durchzusetzen.

Durch eine gezielte Förderung von OSS ist es zudem möglich, die regionale Softwareindustrie zu stärken, wodurch auch die Abhängigkeit von US-amerikanischen und allgemein ausländischen Unternehmen verringert wird. Die in OSS investierten Mittel fließen zudem an andere, meist lokale Unternehmen (Evans & Reddy, 2003; Stürmer, 2009; Dobusch, 2014).

Ein weiteres Argument, das für eine Investition in OSS spricht, ist die Schaffung von Chancengleichheit. Durch den Einsatz von OSS wird ein Beitrag zur gesellschaftlichen Chancengleichheit geleistet. Dies wirkt sich sowohl auf den Bildungsbereich, als auch auf die öffentliche Verwaltung positiv aus (Stürmer, 2009, S. 10).

8.1.2 Argumente gegen eine Intervention

Neben den im vorherigen Kapitel genannten Vorteilen staatlicher Interventionen in der Softwareindustrie gibt es auch Nachteile, die gegen ein Eingreifen in den Markt sprechen. Auffälligstes Gegenargument ist die erfolgreiche Entwicklung von OSS, die auch ohne staatliche Unterstützung funktioniert hat (Comino, Manenti, & Rossi, 2011).

Weiter argumentieren Comino, Manenti und Rossi auch, dass es keine klaren Beweise für ein Marktversagen gäbe, weshalb Eingriffe ebenfalls nicht zwingend nötig seien. Evans und Reddy (2003, S367) vergleichen die Situation der Softwareindustrie mit dem Kampf zwischen Plan- und freier Marktwirtschaft des 20. Jahrhunderts. Dabei kommen sie zum Schluss, dass sich bessere Ergebnisse bezüglich Wohlstand ergeben, wenn das System nicht zu

stark beeinflusst wird. Übertragen auf die Softwareindustrie spricht dies gegen eine staatliche Intervention.

Schmidt und Schnitzer (2003; zitiert in Comino, Manenti & Rossi, 2011) fügen an, dass die Softwareindustrie äusserst gut funktioniere. Eine finanzielle Unterstützung durch die Regierung könne sehr schnell zu einem falschen Gewinner führen während sich gleichzeitig für die restlichen Unternehmen der Anreiz, innovativ zu sein oder bestehende Software zu verbessern, verringert.

8.2 Open Source in der Schweizer Bundesverwaltung

Im folgenden Abschnitt soll aufgezeigt werden, wie innerhalb der Bundesverwaltung und gegenüber der Öffentlichkeit mit OSS umgegangen wird. Ein konkretes Beispiel für eine Verwaltungsbehörde wird anhand des BIT gegeben.

8.2.1 Interne Sicht

Im Jahr 2005 veröffentlichte das *Informatikstrategieorgan Bund* (ISB) die letzte Version der OSS-Strategie für die Bundesverwaltung (Informatikstrategieorgan Bund, 2005). Ziel war es, bis zum Jahr 2007 OSS in der Bundesverwaltung als gleichwertige Alternative zu proprietärer Software zu etablieren.

Aus diesem Grund werden in der Strategie drei Stossrichtungen festgelegt. Erstens werden bei konkretem Bedarf beide Softwarevarianten nach denselben Grundsätzen überprüft. Die Evaluationskriterien sind hierbei Funktionalität, Wirtschaftlichkeit, Sicherheit, Qualität und die Anforderungen der Benutzer. Eine finanzielle Förderung von OSS ist aber nicht vorgesehen. Als zweites soll eigenentwickelte Software nach Möglichkeit wiederverwendet und anderen öffentlichen Verwaltungen in der Schweiz zur Verfügung gestellt werden. Die dritte Stossrichtung entstand aus einer damals durchgeführten Situationsanalyse. Diese kam zum Ergebnis, dass für einen erfolgreichen Einsatz von OSS zuerst die Voraussetzungen in den Bereichen Standardisierung, Organisation, Ausbildung und Information, Wirtschaftlichkeit und Recht geschaffen werden müssen.

Grundsätzlich sollte bei der Softwarebeschaffung weder OSS, noch proprietäre Software bevorzugt werden. Es galt jeweils die beste Lösung zu finden, unabhängig der Lizenzierung der Software. Laut Evans und Reddy (2003, S. 378) sollte sich eine Regierungsbehörde für diejenige Alternative

entscheiden, die den Anforderungen am besten entspricht, was in der OSS-Strategie so aufgenommen wurde.

Trotz des Bewusstseins um Vorteile von OSS wie geringere Abhängigkeiten von Softwarelieferanten oder die Qualität und Wirtschaftlichkeit von Open Source Informatiksystemen vertrat das ISB die Meinung, dass eine Bundesverwaltung kein *Early Adopter* sein sollte. Dies bedeutet, dass OSS erst in Betracht gezogen wird, nachdem sie sich bereits auf dem Markt bewährt hat.

Die Open Source Strategie für die Bundesverwaltung ist eher konservativ, abwartend formuliert. Oberste Priorität hat die Schaffung der Rahmenbedingungen für einen fairen Wettbewerb zwischen OSS und proprietärer Software. Gleichzeitig wird aber auch darauf geachtet, die Abhängigkeit von Anbietern mittels einer Mehrproduktstrategie, zu verringern (Schweizer Bundesversammlung, 2007).

8.2.2 Öffentliche Haltung gegenüber Open Source

Nachdem im obigen Kapitel beschrieben wurde, wie innerhalb der Bundesverwaltung mit OSS umgegangen werden soll, folgt nun die öffentliche Haltung der Schweizer Regierung bezüglich Open Source.

Trotz diversen Forderungen nach mehr Unterstützung von OSS und daraus resultierenden Untersuchungen existiert keine offizielle Open Source Strategie der Schweizer Regierung. Die einzige Strategie, die formuliert wurde ist diejenige des ISB, die jedoch lediglich den verwaltungsinternen Umgang mit OSS regelt.

Aus Interpellation 14.3563 lässt sich ableiten, dass die Schweizer Regierung keinen Bedarf sieht, zu untersuchen, wie gross das wirtschaftliche Potential von OSS ist. Begründet wird diese Haltung mit den hohen Kosten einer entsprechenden Untersuchung. Die beschränkten Mittel sollten nur dort eingesetzt werden, wo dringender Wissens- und Handlungsbedarf besteht. Weiter liege im Softwaremarkt kein Marktversagen vor, weshalb ein Eingriff nicht nötig sei (Schweizer Bundesversammlung, 2014a).

Ein höherer Marktanteil von OSS wird aber grundsätzlich begrüsst, da damit die geforderte Unterstützung offener Standards, Formaten und Schnitt-

stellen per Definition gewährleistet sein sollte. So werden seit 2003 auch regelmässig Informationsveranstaltungen über OSS für Behörden durchgeführt (Schweizer Bundesversammlung, 2007).

Die mittel- und langfristigen finanziellen Auswirkungen eines erhöhten OSS-Einsatzes könnten jedoch nicht beurteilt werden, da in Referenzprojekten wie dem Linux-Projekt der Stadt München noch kein endgültiges Fazit gezogen worden ist. Am Beispiel des Münchner Projekts liesse sich lediglich ableiten, dass den Einsparungen bei den Lizenzkosten teils hohe Migrationskosten gegenüberstehen, weshalb eine Umstellung sorgfältig geplant und nach wirtschaftlichen Kriterien durchgeführt werden müsse (Schweizer Bundesversammlung, 2009).

8.3 Das BIT

Das BIT ist einer der internen IKT-Leistungserbringer³⁶ der Schweizer Regierung (Bundesamt für Informatik und Telekommunikation, 2015). Das Amt unterstützt die Verwaltung mittels Informatik, wirtschaftliche, sichere und benutzerfreundliche Lösungen zu entwickeln. Das BIT beschäftigt etwas mehr als 1'000 interne Mitarbeiter und betreut rund 25'000 Arbeitsplatzsysteme und 4'600 Server. Konkrete Aufgaben des BIT umfassen unter anderem den Betrieb von Rechenzentren und Fachanwendungen, die Bewirtschaftung von Arbeitsplatzsystemen, sowie den Betrieb von Datennetzen und Telekommunikations-Infrastrukturen für die gesamte Bundesverwaltung oder einzelne Departemente. Zusätzlich unterstützt das BIT die Geschäftsprozesse in der Bundesverwaltung und sorgt für eine funktionierende Telekommunikation zwischen allen Bundesstellen im In- und Ausland.

Als Erbringer von Informatik-Standarddiensten ist das BIT bundesweit in den Bereichen Telekommunikation, Internet, betriebswirtschaftliche Lösungen und operative Sicherheit tätig. Dabei werden wo immer sinnvoll partnerschaftliche Zusammenarbeiten mit anderen IKT-Leistungserbringern angestrebt. Die angebotene Produktpalette an Arbeitsplatzsystemen umfasst aber

³⁶ IKT steht für Informations- und Kommunikationstechnologie (Wirtschaftslexikon24.com, 2015)

nicht nur standardisierte Lösungen, sondern auch die Entwicklung und Anwendung individualisierter Fachanwendungen (Bundesamt für Informatik und Telekommunikation, 2015).

Im Jahr 2012 wurde eine Mission mit insgesamt sechs Stossrichtungen definiert, die den Rahmen für die Kernaufgaben³⁷ des BITs bilden. Sie richtet sich dabei nach der Bundesinformatikverordnung 172.010.58 (Bundeseinformatikverordnung BinfV, 2012). Laut Giovanni Conti, Direktor des BIT, haben folgende Missionsziele Priorität. Auf Wunsch der Verantwortlichen des BIT, werden folgende sechs Punkte direkt von der Webseite zur Mission des BIT übernommen (Eidgenössisches Finanzdepartement, 2012b):

1. Durchgängige integrierte IKT-Leistungen

Das BIT erbringt für seine Kunden durchgängige und integrierte IKT Leistungen. Es sichert die dazu notwendige Kompetenz und wirkt zwischen den verschiedenen Akteuren der Leistungserbringung verbindend.

2. Integration und verlässlicher Betrieb

Für das dem BIT anvertraute Anwendungs- und Systemportfolio sorgt das BIT im Auftrag des Kunden für

- einen verlässlichen Servicebetrieb und Servicemanagement;
- die dazu notwendige, lösungsorientierte Gesamtintegration der Services in die IKT Landschaft;
- je nach Fall für die entsprechende Applikationsentwicklung oder die Spezifikation und Beauftragung der Entwicklung an Dritte.

Sinngemäss ist das BIT auch Leistungserbringer für

- Infrastrukturleistungen für andere Leistungserbringer der Bundesverwaltung (im Sinne einer Vorleistung);

³⁷ Das BIT erbringt für seine Kunden durchgängige und integrierte Informations- und Telekommunikations-Leistungen (IKT-Leistungen). Es sichert die dazu notwendige Fachkompetenz und wirkt zwischen den verschiedenen Akteuren der Leistungserbringer verbindend (Eidgenössisches Finanzdepartement, 2012a).

- Komponenten der Service Oriented Architecture (SOA) , wo Verwaltungseinheits-übergreifende Bedürfnisse es rechtfertigen.

3. Standarddienste

Das BIT positioniert sich als Leistungserbringer für Standarddienste, unter Berücksichtigung des Gesamtinteresses der Bundesverwaltung.

4. Kundennähe

Das BIT arbeitet mit der notwendigen Nähe zum Kunden. Das BIT unterstützt den Kunden in der Optimierung des Einsatzes der IKT. Das BIT kennt, versteht und unterstützt den Kunden und hilft ihm, seinen IKT Nutzen zu maximieren.

5. Fokus

Das BIT konzentriert sich auf eine gute Beherrschung und Erfüllung seiner Kernaufgaben.

6. Partnerschaft

Das BIT arbeitet als Partner der Leistungsbezüger und Leistungserbringer in der Bundesverwaltung.

Diese sechs Punkte umfassen die 2012 definierte Mission des BIT, wie sie im entsprechenden Dokument beschrieben ist. Im nächsten Kapitel wird genauer betrachtet, wie OSS beim BIT verwendet und eingesetzt wird.

8.4 Open Source beim BIT

Die Analyse der Verwendung von OSS im BIT basiert auf dem im Rahmen dieser Arbeit erstellten OSS Portfolio und der OSS Strategie der Bundesverwaltung (Informatikstrategieorgan Bund, 2005). Um die Verwendung von OSS beim BIT untersuchen zu können, wurde eine IST-Erhebung der im Einsatz stehenden Software erstellt.

Die Zusammenstellungen in Tabelle 3 zeigen, dass insgesamt 64 verschiedene Arten von OSS, die in den verschiedensten Anwendungsgebieten verwendet werden, wovon 24 in den Clientbereich fallen. Die restlichen 40 Anwendungen werden im Serverbereich verwendet. Die Palette eingesetzter Software erstreckt sich von Internetbrowsern wie dem *Mozilla Firefox* oder der

SeaMonkey-Suite, über Datenbankmanagementsystemen wie *MySQL* oder *MariaDB* bis hin zu Webservern wie *Apache Tomcat* oder *XAMPP*. Von der eingesetzten OSS stehen mehr als die Hälfte unter der GPL. OSS, die gleichzeitig verschiedene Lizenztypen verwendet, wurde nicht dazu gezählt.

Ein Vergleich mit Abbildung 4 zeigt, dass die GPL-Lizenz beim BIT noch dominanter ist, als dies in der IT-Branche heute zu erwarten wäre. Wie in Kapitel 4.3 erwähnt wird, werden restriktive Lizenzen eher im Entwicklungsbereich eingesetzt. Ein derart hoher Anteil restriktiver Lizenzen würde in einem herkömmlichen Unternehmen für eine hohe Innovationstätigkeit sprechen. Aufgrund des vom Bundesrat formulierten Leistungsauftrages ist das BIT jedoch lediglich dafür verantwortlich, seine Kernaufgaben zu erfüllen. Innovativität spielt darin keine Rolle. Aus Interpellation 14.4199 geht zusätzlich hervor, dass es für Bundesämter nicht erlaubt Quellcode freizugeben, was die Zusammenarbeit mit einer Open Source Community unmöglich macht (Schweizer Bundesversammlung, 2014b).

Lizenztyp	Produkt	
	Server	Client
Affero GPL	Monit	
Apache	Apache Tomcat SpamAssassin Subversion	Activiti Java Spring jBPM Maven .NET
BSD	HyperSQL Database Engine OpenSSH	
CPL		Junit
DFSG		Debian
EPL	Nexus	BIRT Developer Eclipse
GPL	Ghostsript Cricket FileZilla KeePass Mantis MariaDB mRemote NG MySQL Nessus PostGIS (PostgreSQL) RRDtool Satellite Server (Redhat) SCCM Client Center Squid SuSE Linux, RedHat, RPM TortoiseSVN WinSCP Wireshark XAMPP Xming	FreeMind Git ILIAS 3 Java EE Mediawiki Nagios Notepad++ Samba StarUML Webalizer WinMerge
IBM PL	Postfix	
ISCL	BIND (DNS)	
LGPL	Hibernate SOAPUI SonarQube Syslog-ng	JavaMelody
MIT	Jenkins PuTTY	
MPL, GPL, LGPL	SeaMonkey	Mozilla Firefox
OpenLDAP PL		OpenLDAP
Sendmail	Sendmail	
TrueCrypt	TrueCrypt	
zlib		Nunit

Tabelle 3 OSS beim BIT

8.5 Fazit

Wie die Befunde aus der Open Source Studie (2012) in Abbildung 12 zeigen, verlangen die befragten Unternehmen von der Regierung, dass Innovationen und Wettbewerb in der IT-Branche gefördert werden. Ebenfalls wünschen sie sich, dass lokale OSS Anbieter unterstützt werden.

Das Fehlen einer aktuellen Strategie zu einem Thema, das derart aufstrebend, zukunftsweisend und expandierend ist, vermittelt den Eindruck, Open Source spiele in der Schweizer Regierung nur eine untergeordnete Rolle. Das ISB erkannte in der 2005 veröffentlichten, internen OSS-Strategie für die Bundesverwaltung, dass OSS und proprietäre Software gleichbehandelt werden sollten. Die Hauptaufgabe der Regierung ist es, die entsprechenden Rahmenbedingungen zu schaffen. Dies entspricht auch den Empfehlungen für das Verhalten von Regierungen in der Literatur (z.B. Comino, Manenti & Rossi, 2011).

Die Reaktion des Bundesrates auf die Interpellation 14.3563 im Jahr 2014 zeigt jedoch eine andere Haltung. Eine Untersuchung des wirtschaftlichen Potentials von OSS wurde abgelehnt, da sie zu unwichtig und teuer sei (Schweizer Bundesversammlung, 2014a). Diese Begründung ist nicht nachvollziehbar, zumal das Angebot von OSS und damit verbundenen Dienstleistungen in den letzten Jahren deutlich professioneller geworden ist und so ein noch grösserer Zusatznutzen aus der Verwendung von OSS resultieren kann. Im Gegensatz zur Haltung des Bundesrates wird die Open Source Studie Schweiz regelmässig durch das ISB finanziert (z.B. Open Source Studie, 2015, S. 2). Auffällig ist aber, dass die interne OSS-Strategie des ISB seit mehr als zehn Jahren nicht mehr überarbeitet wurde und auch erste Evaluationen erst Jahre nach dem gesetzten Zeitfenster³⁸ von Politikern gefordert wurden (Schweizer Bundesversammlung, 2009). Auf die Anfrage Walter Donzés nach möglichen Gründen, wieso die Strategie „vor sich her schlummere“, antwortete der Bundesrat mit einer Aufzählung der durchgeführten Massnahmen. Mögliche Schwachstellen oder eine Aktualisierung der Strategie wurden nicht angesprochen.

³⁸ Die Strategie beschreibt die Ziele in der Bundesverwaltung bis ins Jahr 2007.

Die fehlende Unterstützung von OSS ist allerdings nicht nur ein schweizerisches Phänomen. Gemäss Dobusch (2014) scheiterten Open Source Projekte in Städten wie Berlin oder Wien nicht aufgrund ökonomischer Faktoren, sondern weil politische Unterstützung und Entschlossenheit fehlten. Das Fehlen einer offiziellen Open Source Strategie setzt politische Zeichen, die eine Bundesverwaltung aufgrund ihrer Vorbildrolle nicht aussenden darf. In den letzten zehn Jahren hat sich der Markt für OSS stark verändert und ist wesentlich komplexer geworden. Viele Mängel und Fehler wurden seit 2005 verbessert und professionalisiert. Für die Bundesverwaltung als Unternehmen müsste Open Source deshalb ebenfalls auch eine wichtigere Rolle einnehmen. In Ländern wie Deutschland, Frankreich oder Italien wird Open Source bereits seit Jahren eingesetzt und aktiv gefördert (Evans & Reddy, 2003, S. 374ff). Stürmer (2010) machte auf das Problem aufmerksam, indem er behauptete, dass ein Umstieg auf OSS aufgrund des einwirkenden Kostendrucks unmittelbar erfolgen müsse. Auch wenn finanzielle Aspekte nicht das Hauptargument für eine Migration auf OSS darstellen, könnten die jährlichen Kosten durch einen Umstieg doch klar verringert werden. Jähnert (2012) argumentiert, dass die Skepsis gegenüber OSS im öffentlichen Sektor nach wie vor sehr gross ist. Diese Skepsis hat ihren Ursprung aber oftmals nicht in einer mangelhaften Qualität von OSS, sondern in fehlendem Knowhow der Nutzer und sollte deshalb schnellstmöglich durch Aufklärung, Aus- und Weiterbildungen abgebaut werden.

Das Problem in der Schweizer Regierung scheint die Haltung der verschiedenen involvierten Parteien gegenüber Open Source zu sein. Das BIT als Verwaltungsbehörde führt lediglich die ihm aufgetragenen Aufgaben aus und erstickt jegliche Innovationen bereits im Keim. Ein Bundesamt für Informatik sollte über genügend Knowhow, Kapazitäten und vor allem Eigenmotivation verfügen, um nicht nur das nötige anzubieten, sondern seine Produkte und Dienstleistungen kontinuierlich zu hinterfragen und zu verbessern. Das ISB machte 2005 mit der Veröffentlichung der internen OSS-Strategie einen

Schritt in die richtige Richtung. Zusätzlich wurde ab 2009 beschlossen, die Open Source Studie Schweiz finanziell zu unterstützen³⁹.

Die Vernachlässigung der internen und das Fehlen einer offiziellen OSS-Strategie passen aber nicht zum damals eingeschlagenen Weg. Möglicherweise liegt ein ähnliches Problem wie beim BIT vor und das ISB führt nur aus, was beispielsweise vom Bundesrat angeordnet wird. Auf Ebene des Bundesrates ist vermutlich auch das Hauptproblem auszumachen. Die Haltung des Bundesrates ist nicht mehr zeitgemäss. Das Thema Open Source sollte nicht nur am Rand behandelt werden. Die gewonnene Bedeutung der letzten Jahre fordert jede Regierung dazu auf, OSS fest in die strategische Planung einzubauen. Wie oben bereits erwähnt setzen Länder wie Deutschland klare Zeichen in Richtung Open Source und zeigen so, dass es möglich ist, erfolgreich ein Land zu führen und gleichzeitig neue Dinge wie Open Source zu berücksichtigen.

Handlungsempfehlung

Die Veröffentlichung einer offiziellen OSS-Strategie sollte oberste Priorität haben, weshalb das aktuelle Potenzial von OSS schnellstmöglich untersucht werden muss. Wie in der internen Strategie richtig erkannt wird, sollte eine Regierung für Rahmenbedingungen sorgen, die einen fairen Wettbewerb zwischen OSS und proprietärer Software erlauben. Eine Verbannung proprietärer Software ist keine Lösung und würde den gesellschaftlichen Wohlstand verringern (Evans & Reddy, 2003, S. 371). Allerdings könnten faire Rahmenbedingungen geschaffen werden, indem der Einfluss bestehender Patente verringert wird. Diese schränken die Innovationsfähigkeit der Branche stark ein und schrecken potentielle Investoren ab (Comino, Manenti, & Rossi, 2011, S. 95). Um dies zu verhindern schlägt Lessig (2002; zitiert in Comini, Manenti & Rossi, 2011) vor, Quellcode, der in staatlich finanzierten Projekten entstanden ist, automatisch wieder zu veröffentlichen. Dies erhöht die Neutralität staatlicher Investitionen in Open Source Projekte und verhindert die Bevorteilung

³⁹ Die Open Source Studie Schweiz wurde 2006 unter dem damaligen Namen *FOSS-Studie Schweiz* zum ersten Mal durchgeführt. Seither fand die Studie im Abstand von jeweils drei Jahren statt.

einzelner Unternehmen. Derartige Investitionen könnten so als eine Art Entwicklungshilfe verstanden werden, von der alle profitieren. Gemäss Dobusch (2014) müssten auch Partnerschaften unter Städten, respektive Regierungsbehörden angestrebt werden, da die Anforderungen an eine Software sich zwischen den Behörden nicht grundlegend unterscheiden. Dadurch könnte entwickelte Software ausgetauscht, Kosten gespart und vom Wissen anderer Behörden profitiert werden.

Comino, Manenti und Rossi (2011) schlagen vier mögliche Wege vor, wie eine Regierung im Softwaremarkt intervenieren kann. Erstens kann eine bestimmte Software selbst verwendet und somit die Nachfrage erhöht werden. Zweitens könnte eine Informationskampagne gestartet werden, die zum Ziel hat, potentielle Anwender über die Existenz und Eigenschaften von Open Source zu informieren. Dies geht mit den Befunden Jähnerts (2012) einher, wonach fehlendes Wissen und Unsicherheit der wichtigste Grund für die nach wie vor bestehende Skepsis gegenüber OSS ist. Laut Comino, Manenti und Rossi (2011) könnten öffentliche Institutionen auch dazu gezwungen werden, OSS zu verwenden. Drittens könnte die Entwicklung neuer Software aktiv gefördert werden, indem gewisse Projekte finanziell oder mit Knowhow unterstützt werden. Die letzte Möglichkeit wäre die Zahlung einer Subventionsprämie für die Verwendung von OSS. Sämtliche Interventionen sollen aber gemäss Evans und Reddy (2003) erst unternommen werden, wenn ein Marktversagen vorliegt und dies durch eine staatliche Lösung behoben werden kann, ohne zu hohe Kosten zu verursachen. Ebenso muss sichergestellt werden, dass durch die getroffenen Massnahmen nicht neue Probleme entstehen.

Um Open Source insbesondere in der Privatwirtschaft zu einer grösseren Bekanntheit zu verhelfen, würde sich eine Zusammenarbeit mit dem *OSS Directory* anbieten. Das *OSS Directory* ist eine Internetplattform, die das einfache Auffinden von Unternehmen, die professionellen Support für bestimmte OSS erbringen, aus der Schweiz und dem nahen Ausland ermöglicht. Es verfügt über eine umfangreiche Datenbank von rund 300 Firmen, die Dienstleistungen für rund 400 Open Source Produkte anbieten. Zusätzlich sind insgesamt etwa 270 Nutzer erfasst, bei denen rund 300 Referenzen umgesetzt worden sind. Das *OSS Directory* wurde 2004 durch den Verein *Swiss Open Systems User Group /ch/open* gestartet und wird heute innerhalb des Open

Source Portals www.opensource.ch betrieben (Swiss Open Systems User Group /ch/open, 2015). Durch einen Ausbau dieser Plattform könnte ein branchenweiter Marktplatz geschaffen werden, auf dem Unternehmen oder Privatpersonen mit vergleichsweise geringem Aufwand nach spezifischer OSS oder nach Geschäftspartnern z.B. für Supportdienstleistungen suchen könnten oder ihre Produkte anbieten. OSS Anbieter könnten dadurch Kosten sparen, da sie ihre Marketingleistungen im besten Fall auf eine einzige Plattform reduzieren könnten. Benutzern könnte eine Plattform geboten werden, auf der sie sich sowohl über Open Source an sich informieren, aber auch Open Source Lösungen für ein spezifisches Problem suchen könnten.

Die in diesem Kapitel behandelten Aspekte zeigen, dass OSS in der Schweizer Regierung grundsätzlich wahrgenommen wird. Die letzte Entschlossenheit einer konsequenten Förderung von Open Source findet jedoch nicht statt, wie sich am Beispiel der fehlenden offiziellen Open Source Strategie zeigt. Das Fehlverhalten zeigt sich aber nicht nur gegen aussen. So wurde die interne OSS-Strategie der Bundesverwaltung seit zehn Jahren nicht mehr erneuert, weshalb die darin formulierten Ziele ihre Aktualität und Relevanz verlieren. Eine Regierung ist nicht primär dafür verantwortlich, neue Wege einzuschlagen, sie hat jedoch auch eine Vorbildfunktion, die sie wahrnehmen muss. Um dieser gerecht zu werden, muss zwingend eine offizielle Open Source Strategie existieren. Zudem sollte die interne Strategie regelmässig und dem technischen Fortschritt entsprechend aktualisiert werden. Empfehlenswert wäre eine klare Zuteilung der Verantwortung an das ISB, das bereits die letzte Strategie formuliert hat. Das ISB verfügt über das nötige Knowhow, das für das Verfassen einer solchen Strategie benötigt wird.

9 Zusammenfassung und Ausblick

OSS hat sich seit seiner Entstehung in den 1960er Jahren zu einer gleichwertigen, konkurrenzfähigen und geschäftstauglichen Alternative zu proprietärer Software entwickelt. Dennoch scheint sich Open Source nicht gegen Konkurrenzprodukte von Microsoft, Oracle, SAP oder anderen proprietären Anbietern durchsetzen zu können. Als Gründe werden z.B. fehlende Supportdienstleistungen, die unsichere rechtliche Lage oder eine schlechte Benutzerakzeptanz angegeben (Open Source Studie Schweiz, 2012). Auch wird behauptet, die Vorteile von OSS seien nicht so gross, wie angenommen (Fuggetta, 2003). Eine Vielzahl der genannten Nachteile könnten jedoch laut Stürmer (2009) positiv beeinflusst werden, wenn Anwender besser informiert wären.

Das in dieser Arbeit erstellte Framework hilft dabei, Informationsdefizite abzubauen. Es macht Unternehmen auf die verschiedenen positiven und negativen Aspekte von OSS aufmerksam und zeigt ihnen gleichzeitig, ob ihre Organisation für eine Umstellung bereit ist. Besonders das Aufklären bezüglich relevanter Vor- und Nachteile für ein Unternehmen ist essentiell, da Open Source oft fälschlicherweise darauf reduziert wird, Kosteneinsparungen zu bewirken. Dabei ist das Wegfallen der Lizenzkosten nur eine von vielen Veränderungen, die eine Migration auf OSS empfehlenswert macht. So können Unternehmen durch eine Umstellung auf OSS beispielsweise auch ihre Unabhängigkeit gegenüber externen Software- und Dienstleistungsanbietern reduzieren oder erhalten die Möglichkeit, Wissen mit der Open Source Community auszutauschen. Allerdings bewirkt eine Migration auf OSS auch, dass Haftung und Support nicht mehr automatisch und eindeutig geklärt sind. Weiter steigen die Anforderungen an Führungskräfte aufgrund der meist grösseren Entwicklerteams, der schlechteren Projektorganisation und fehlender Verpflichtungen der Community-Entwickler gegenüber dem Unternehmen. Diese Beispiele verdeutlichen, dass die Konsequenzen eines Umstiegs auf OSS sowohl im positiven, als auch im negativen Sinn oft unterschätzt werden.

Um die Geschäftsleitung eines Unternehmens zu entlasten, werden zusätzlich zum Framework Handlungsempfehlungen abgegeben, wie negativ bewertete Faktoren verbessert werden können, sollte dennoch eine Software-

migration auf OSS angestrebt werden. Im Framework wird sowohl das Unternehmen an sich, als auch die darauf einwirkende Umgebung bewertet. Diese Verbindung ist wichtig, da die Entscheidung, auf OSS umzusteigen, auch von Faktoren beeinflusst wird, die ein Unternehmen nur schlecht beeinflussen und verändern kann.

Zu diesen externen Faktoren zählen beispielsweise rechtliche Einflüsse. Hierbei spielt eine Regierungsbehörde eine wichtige Rolle, da sie als einzige im Softwaremarkt aktive Partei die Möglichkeit hat, die Rahmenbedingungen für einen fairen Wettbewerb zwischen OSS und proprietärer Software zu schaffen und anzupassen. Die Schweizer Regierung tritt diesbezüglich sehr inkonsequent auf. Zwar wurde 2005 eine interne OSS-Strategie formuliert, die den Umgang mit Open Source regelt, allerdings existiert keine externe, offizielle Open Source Strategie. Zudem wird diese interne Strategie nicht konsequent umgesetzt und wurde seit zehn Jahren nicht mehr aktualisiert. Besonders im Hinblick auf den steigenden Einfluss und die Zukunftsorientiertheit von Open Source ist das Fehlen einer öffentlichen Stellungnahme und einer offiziellen Strategie betreffend Open Source nicht nachvollziehbar. Hinzu kommt, dass sich die Informatikbranche im letzten Jahrzehnt erneut stark verändert hat und sich in Zukunft wahrscheinlich noch schneller verändern und weiterentwickeln wird. Deshalb müsste die Marktsituation erneut analysiert und eine auf diesen Erkenntnissen basierende OSS-Strategie formuliert werden.

Damit sich Open Source in Zukunft vermehrt durchsetzen kann, muss es von der Gesellschaft flächendeckend als gleichwertige Alternative zu proprietärer Software anerkannt werden. Stellvertretend für die gesamte Open Source Community sollten Organisationen wie die Free Software Foundation die offene Softwareentwicklung nicht nur fachspezifisch unterstützen, sondern die Gesellschaft detailliert über Open Source und seine Vorzüge informieren. Dies könnte durch den Aufbau eines „Vorzeigeunternehmens“, das ausschliesslich OSS einsetzt, erreicht werden. Eine erfolgreiche Durchführung dieses Projekts würde beweisen, dass Open Source sämtliche Geschäftsbereiche in einem Unternehmen erfolgreich abdecken kann, wie dies Marwan (2010) behauptet. Gleichzeitig wäre es so besser möglich, Lücken oder notwendige Verbesserungen bestehender Anwendungen zu entdecken und entsprechende Lösungen zu entwickeln.

Die Entwicklung des hier vorgestellten Frameworks ist mit dieser Arbeit nicht abgeschlossen. Das Modell sollte in der Praxis getestet und danach anhand dieser Erfahrungen ergänzt, angepasst und erweitert werden. Bei der Weiterentwicklung wäre besonders die Zusammenarbeit mit einem Unternehmen erstrebenswert, um die definierten Faktoren genauer und praxisnaher erheben und bewerten zu können. Dabei könnten verschiedene Versionen des Frameworks erstellt werden, die sich z.B. anhand der Charakteristiken des untersuchten Unternehmens oder der Branche unterscheiden. Ebenfalls könnte mit dem entsprechenden Partnerunternehmen eine Analyse der Total Cost of Ownership durchgeführt werden. Diese kann als zusätzliches Analysewerkzeug zur Bestimmung der längerfristigen Rentabilität einer Umstellung auf OSS verwendet werden. Weiter wäre es sinnvoll, noch klarer zwischen Softwarenutzern und –Entwicklern zu unterscheiden oder auch hierfür separate Versionen zu erstellen.

Bezüglich der Erhebung der für das Framework relevanten Kriterien kann der Fragenkatalog ausgeweitet werden. Auch könnte er mittels eines professionellen online Tools erstellt und so einfacher verteilt und ausgewertet werden. Diese beiden Massnahmen tragen zur Steigerung der Objektivität und Reliabilität bei und erhöhen somit die allgemeine Güte und Akzeptanz dieses Erhebungswerkzeugs (Moosbrugger & Kelava, 2012). Als Ergänzung zu den gewonnenen Informationen aus dem Fragebogen wären Interviews mit dem Management und ausgewählten Personen der relevanten Abteilungen hilfreich.

Weiter wäre es interessant zu untersuchen, wie die Regierungen anderer Länder mit OSS umgehen. Daraus könnten konkrete Handlungsempfehlungen für die Schweizer Regierung abgeleitet werden.

Die Frage, ob eine Umstellung auf Open Source sinnvoll ist, kann in dieser Arbeit nicht für jedes Unternehmen und jeden Fall beantwortet werden. OSS bietet einige verlockende Vorteile. Ein Unternehmen, das eine Migration in Erwägung zieht, muss sich aber auch im Klaren sein, welche Veränderungen ein solcher Wechsel bewirken würde. Aufgrund der strategischen Implikationen ist eine seriöse Situationsanalyse des Unternehmens selber und der externen Rahmenbedingungen sehr wichtig. Hierbei bietet diese Arbeit mit

dem darin entwickelten Framework ein Werkzeug, das sowohl die Situationsanalyse, als auch die darauffolgende Entscheidung für oder gegen eine Umstellung auf OSS erleichtert.

Anhang A

A.1 Motivatoren in Open Source

Diese Auflistung entstammt der Studie von Kroghs et al. (2012, S. 654), welche die verschiedenen Arten von Motivation in Open Source Projekten auflistet.

Quellen A-L

Quelle	Jahr	Intrinsisch				Extrinsisch (internalisiert)				Extrinsisch	
		Ideologie	Altruismus	Community	Spaß	Reputation	Reziprozität	Lerneffekte	Eigengebrauch	Karriere	Finanziell
Alexy/Leitner	2007										X
Baldwin/Clark	2006										
Benkler	2002				X						X
Berquist/Ljungberg	2001						X				
Bitzer et al.	2003		X						X		
David et al.	2003	X					X	X	X		
David/Shapiro	2008	X		X			X	X	X		
Fershtman/Gandal	2004										
Ghosh	2005	X	X			X		X	X	X	X
Hars/Ou	2002		X	X		X		X	X	X	
Haruvy et al.	2003		X								
Hemetsberger	2004	X	X	X	X	X	X	X	X	X	
Hertel et al.	2003	X		X	X	X			X	X	X
Ke/Zhang	2008		X								
Lakhani/von Hippel	2003	X			X	X	X		X		
Lakhani/Wolf	2005	X		X	X	X	X	X	X	X	X
Lattemann/Stieglitz	2005					X			X		X
Lee/Cole	2003										
Lerner/Tirole	2002					X				X	
Luthiger/Jungwirth	2007				X						X

Quellen M-Z

Quelle	Jahr	Intrinsisch				Extrinsisch (internalisiert)				Extrinsisch	
		Ideologie	Altruismus	Community	Spass	Reputation	Reziprozität	Lerneffekte	Eigengebrauch	Karriere	Finanziell
Markus	2007										
Okoli/Oh	2007					X					
O'Mahony/Ferraro	2007										
Oreg/Nov	2008		X			X		X			
Osterloh/Rota	2007		X						X		
Riehle	2007									X	
Roberts et al.	2006				X	X		X	X	X	X
Rullani	2007										
Schofield/Cooper	2006		X	X					X		
Shah	2006	X			X		X	X	X		
Späth et al.	2008					X		X			
Stewart et al.	2006										
Stewart/Gosain	2006	X	X			X		X			
von Hippel/von Krogh	2003				X			X	X		
von Krogh et al.	2003										
Wu et al.	2007		X					X	X	X	
Xu et al.	2009	X			X	X		X	X		
Ye/Kishida	2003							X			
Yu et al.	2007	X	X			X	X	X		X	
Zeitlyn	2003			X							

A.2 Befunde der Open Source Studie

Die folgenden beiden Abbildungen zeigen die in der 2012 durchgeführten Open Source Studie von Unternehmen genannten Vor und Nachteile von OSS. Dabei konnten die genannten Punkte zusätzlich priorisiert werden.

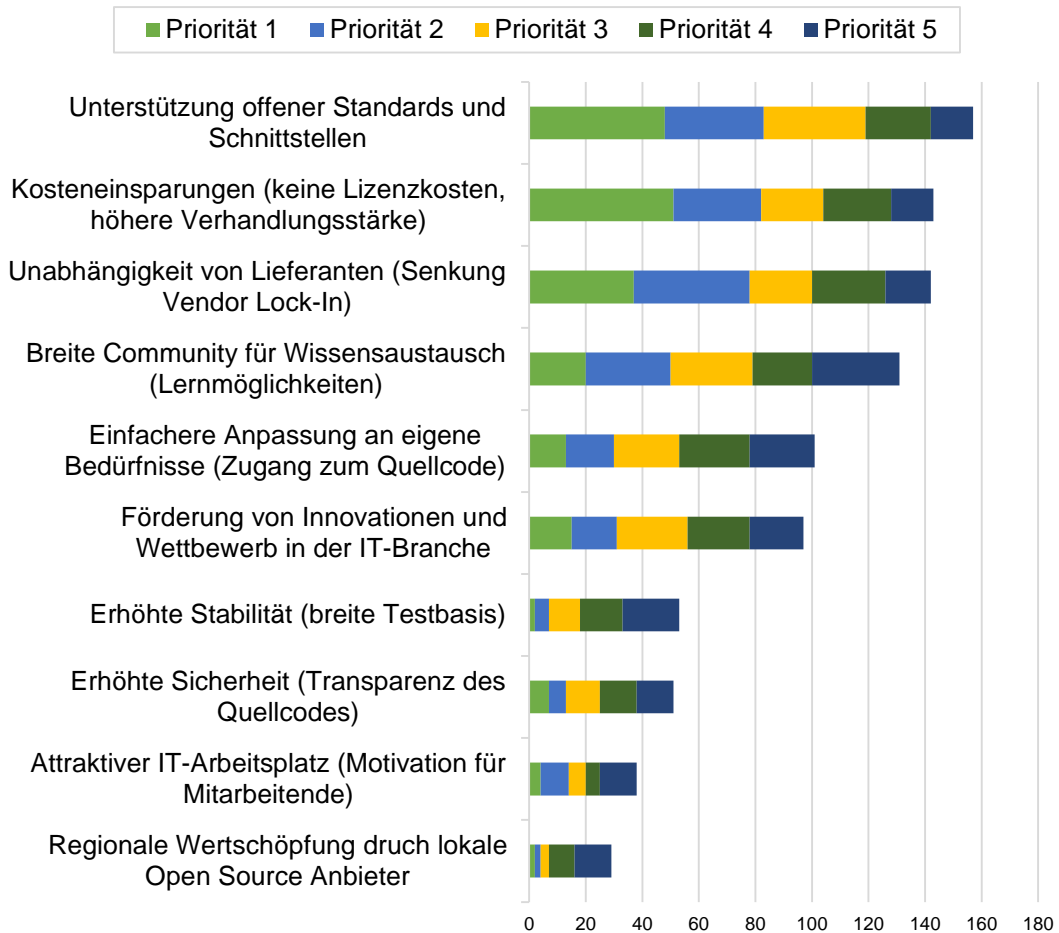


Abbildung 12 Vorteile von Open Source

Quelle: Open Source Studie 2012, S. 7

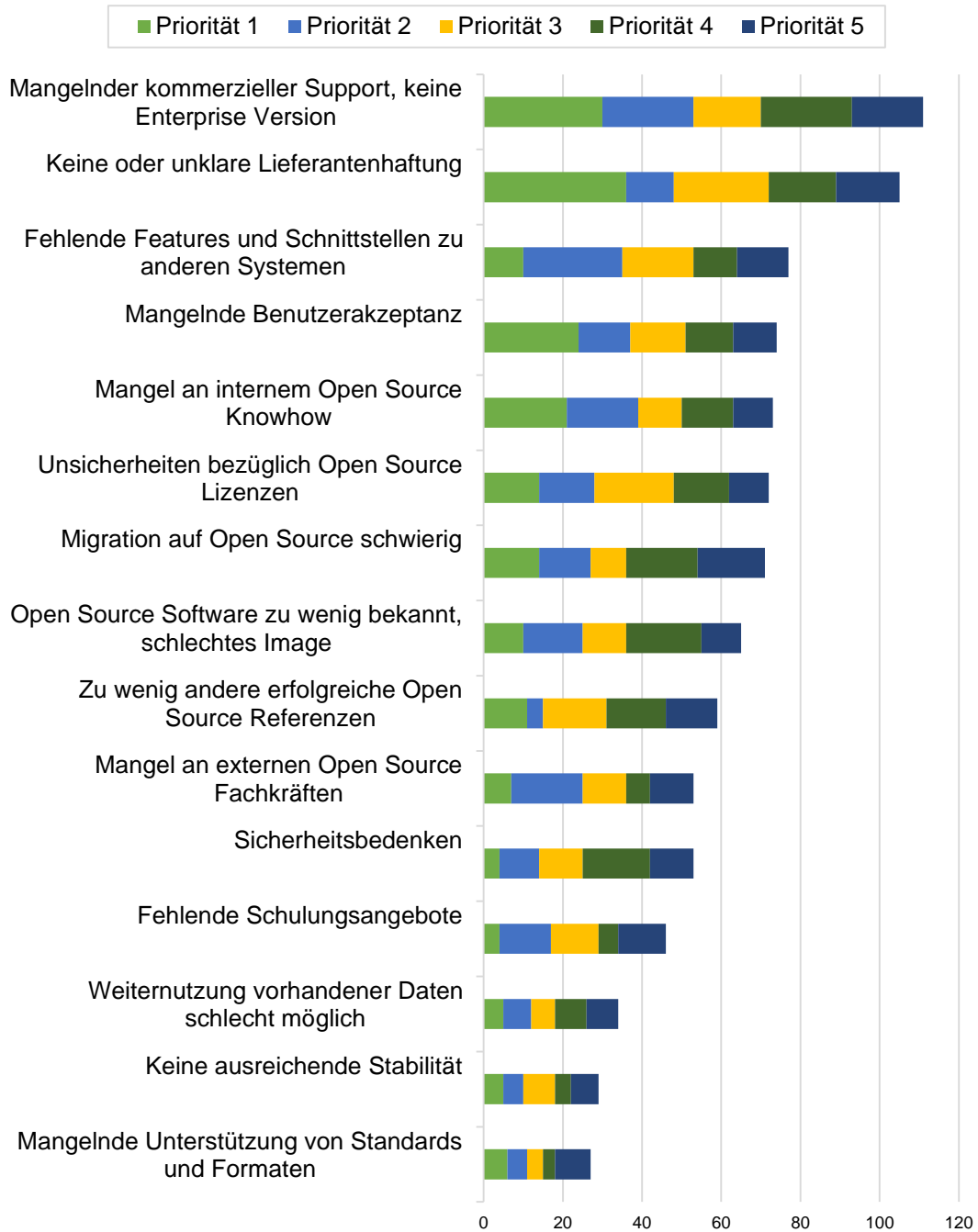


Abbildung 13 Nachteile von Open Source

Quelle: Open Source Studie 2012, S. 10

A.3 Fragenkatalog für Framework

Der folgende Fragenkatalog kann verwendet werden, um die im Rahmen des in Kapitel 7 erstellten Frameworks bestimmten Entscheidungskriterien zu analysieren. Um die aufgelisteten Fragen übersichtlich und vergleichbar zu machen, werden sämtliche Antwortmöglichkeiten entsprechend ihrem Einfluss auf die Entscheidung für oder gegen OSS in der Reihenfolge +1, 0, -1 aufgezählt. Eine detaillierte Beschreibung des Bewertungsverfahrens kann unter 7.4.1 gefunden werden. Es gilt aber zu berücksichtigen, dass nicht in jedem Fall eine negative Bewertung eines Faktors sinnvoll ist, so ist eine Umstellung auf OSS grundsätzlich bei jeder Höhe der erwarteten Kosteneinsparungen möglich.

Ökonomische Faktoren

Geschäftsmodell

Wie sieht das Geschäftsmodell Ihres Unternehmens aus?

Bitte geben Sie an, ob das Unternehmen, in dem Sie arbeiten, seine Erträge eher aus dem Verkauf von Softwarelizenzen oder Dienstleistungen (Support, Wartung, etc.) erzielt oder ob es sich um eine Mischlösung handelt.

- Dienstleistungen
- Beides/unklar
- Lizenzerträge

Mein Unternehmen erzielt seine Einkünfte aus dem Verkauf von...

- Dienstleistungen
- Beidem/Keine Angabe möglich
- Lizenzen

Finanzielle Situation

Mein Unternehmen verfügt über eine gute finanzielle Basis.

- Ja
- Teilweise
- Nein

Wie schätzen Sie die finanzielle Situation Ihres Unternehmens ein?

- Gut
- Mittel
- Schlecht

KostendruckDie Sparmassnahmen in meinem Unternehmen sind...

- Gross
- Mittel
- Gering

Wie schätzen Sie den generell herrschenden Kostendruck auf Ihr Unternehmen ein?

- Hoch
- Mittel
- Tief

Kostenreduktion durch Open Source SoftwareWie gross schätzen Sie das Kosteneinsparungspotential von Open Source für Ihr Unternehmen ein?

- Hoch
- Mittel
- Tief

Wie viel Geld könnte Ihr Unternehmen durch eine Umstellung auf Open Source Software sparen?

- Viel
- Mittel
- Wenig

Unternehmensgrösse

Denken Sie, dass eine Migration auf Open Source Software für ein Unternehmen Ihrer Grösse sinnvoll ist?

- Ja
- Teilweise
- Nein

Aufgrund der Grösse meines Unternehmens ist eine Umstellung auf Open Source Software nicht sinnvoll.

- Trifft nicht zu
- Trifft teilweise zu
- Trifft zu

Technische Faktoren

Verfügbarkeit von Open Source Alternativen

Anstelle der in meinem Unternehmen verwendeten proprietären Software (Microsoft, SAP, Oracle, etc.) gibt es Open Source Alternativen.

- Es gibt Alternativen, die einen Mehrwert bieten.
- Es gibt zwar Alternativen, sie bieten aber keinen Mehrwert.
- Es gibt keine Alternativen.

Denken Sie, dass es viele Open Source Alternativen gibt, die Ihr Unternehmen anstelle von aktuell verwendeter proprietärer Software (Microsoft, SAP, Oracle, etc.) einsetzen könnte?

- Ja
- Teilweise
- Nein

Angestrebte Unabhängigkeit

Wie wichtig ist es für ihr Unternehmen, in der Wahl der verwendeten Software unabhängig von kommerziellen Softwareanbietern zu sein?

- Wichtig
- Mittel
- Unwichtig

Wie hoch schätzen Sie die angestrebte Unabhängigkeit ihres Unternehmens von kommerziellen Softwareanbietern wie Microsoft, Oracle, SAP, etc. ein?

- Hoch
- Mittel
- Tief

Relativer technischer Level

Wie schätzen Sie die Qualität der IT-Infrastruktur Ihres Unternehmens im Vergleich zur Konkurrenz ein?

- Schlechter
- Gleich
- Besser

Denken Sie, dass ein Unternehmen durch eine Umstellung auf Open Source Software den technischen Rückstand auf die Konkurrenz verringern kann?

- Ja
- Teilweise
- Nein

Risikobereitschaft (offener Quellcode)

Wie hoch schätzen Sie die Risikobereitschaft Ihres Unternehmens im Zusammenhang mit offenem Quellcode ein?

Offener Quellcode bietet z.B. die Chance, vom Wissen der Open Source Community zu profitieren. Gleichzeitig ist dadurch aber auch das Risiko von Hackerangriffen höher.

- Gross
- Mittel
- Gering

Mein Unternehmen ist bereit, die Risiken, die offener Quellcode mit sich bringt, einzugehen, um die damit verbundenen Vorteile nutzen zu können.

Durch die Offenlegung des Quellcodes steigt das Risiko von Hackerangriffen. Allerdings kann dadurch auch vom Wissen der Open Source Community profitiert werden.

- Trifft zu
- Trifft teilweise zu
- Trifft nicht zu

Kompatibilität des Systems

Wie gut funktioniert das Zusammenspiel des aktuell verwendeten IT-Systems in Ihrem Unternehmens (Soft- und Hardware)?

- Schlecht
- Mittel
- Gut

Denken Sie, dass die Interoperabilität zwischen den eingesetzten Komponenten im System Ihres Unternehmens durch einen Umstieg auf Open Source Software verbessert werden könnte?

- Ja
- Teilweise
- Nein

Abhängigkeit von Support

Wie gross schätzen Sie die Abhängigkeit von externem Support in Ihrem Unternehmen ein?

Eine grosse Abhängigkeit besteht, wenn auftretende Probleme nicht intern gelöst werden können.

- Gering
- Mittel
- Gross

Wie gross schätzen Sie den Anteil an Softwareproblemen, die intern und ohne fremde Hilfe gelöst werden können, ein?

- Gross
- Mittel
- Gering

Kontrolle über Software und Daten

Wie wichtig ist Ihrem Unternehmen die Kontrolle über verwendete Software und geschäftsrelevante Daten?

- Wichtig
- Mittel
- Unwichtig

Legt Ihr Unternehmen Wert darauf, geschäftsrelevante Software und Daten stets kontrollieren zu können?

- Ja
- Teilweise
- Nein

Erfahrung mit Migrationen

Wie gross ist die Erfahrung mit Softwaremigrationen in Ihrem Unternehmen?

- Gross
- Mittel
- Gering

Verfügt Ihr Unternehmen über Mitarbeiter, die bereits Erfahrung mit Softwaremigration haben?

- Viele
- Mittel
- Wenige

Produkt & Prozess (nur für Softwareentwickler)**Qualität der aktuellen Produkte**

Wie schätzen Sie die Qualität der aktuell in Ihrem Unternehmen entwickelten Produkte ein?

- Schlecht
- Mittel
- Gut

Wie schätzen Sie die Qualitätsunterschiede zwischen Open Source und proprietärer Software ein?

- Open Source Software ist besser
- Beide sind gleich
- Proprietäre Software ist besser

Bestehende Absatzkanäle

Denken Sie, dass eine Erschließung neuer Märkte durch eine Umstellung auf Open Source Software im Interesse Ihres Unternehmens wäre?

- Interessant
- Unsicherheit des Ergebnisses zu gross
- Uninteressant

Die Erschließung neuer Märkte durch eine Umstellung auf Open Source Software, wäre für mein Unternehmen interessant?

- Ja
- Teilweise
- Nein

Modularität der Software

Wie modular sind die Produkte, die Ihr Unternehmen anbietet?

Modularität bezeichnet das Mass, zu welchem die entwickelte Software in unabhängige Komponenten aufgespaltet ist oder werden kann.

- Sehr modular
- Mittel
- Nicht modular

Die von meinem Unternehmen entwickelte Software kann in viele unabhängige Teile aufgeteilt werden.

- Trifft zu
- Trifft teilweise zu
- Trifft nicht zu

Möglichkeit ein Projekt in Teilprojekte aufzuspalten

Wie gut ist es möglich, Projekte (bestehende und neue) in unabhängige Teilprojekte aufzuteilen, die dann in Zusammenarbeit mit der Open Source Community entwickelt werden?

- Gut
- Mittel
- Schlecht

Wie wahrscheinlich ist es, dass (Teil-)Projekte in Ihrem Unternehmen ausgelagert und durch die Open Source Community entwickelt werden?

- Wahrscheinlich
- Teilweise möglich
- Unwahrscheinlich

Innovativität der Unternehmung

Mein Unternehmen ist innovativ.

- Trifft zu
- Trifft teilweise zu
- Trifft nicht zu

In meinem Unternehmen ist es wichtig, Produkte immer weiter zu entwickeln und kreative Lösungen zu erstellen.

- Trifft zu
- Trifft teilweise zu
- Trifft nicht zu

Markt & Kunden (nur für Softwareentwickler)

Präferenzen der Kunden

Wie schätzen Sie die Präferenzen der Kunden Ihres Unternehmens bezüglich der verwendeten Software ein?

- Neu ist immer besser
- Unentschlossen
- „Never change a winning team“

Die Kunden meines Unternehmens legen grossen Wert darauf, immer die neusten Produkte zu erhalten.

- Trifft zu
- Trifft teilweise zu
- Trifft nicht zu

Reifegrad des Marktes

Wie schätzen Sie den Reifegrad des Marktes ein, in dem Ihr Unternehmen tätig ist?

- Wachstumsphase
- Reifephase
- Rückgangsphase

Der Markt, in dem mein Unternehmen tätig ist, befindet sich in einem Stadium wo Neueintritte Sinn machen.

- Trifft zu
- Trifft teilweise zu
- Trifft nicht zu

Personalführung & Management

Unterstützung durch das Management

Wie stark würde eine Umstellung auf Open Source Software in ihrem Unternehmen durch das Management unterstützt?

- Ziemlich
- Mittel
- Wenig

Denken Sie, dass die Geschäftsleitung Ihres Unternehmens einer vollständigen Umstellung auf Open Source Software zustimmen würde?

- Ja
- Teilweise
- Nein

Qualität der Führungskräfte

Wie gross schätzen Sie die Qualität der Führungskräfte in Ihrem Unternehmen ein?

In diesem Fall werden unter Qualität vor allem die sozialen Fähigkeiten verstanden. In einem Open Source Projekt ist es entscheidend, wie gut der Leader Mitglieder der Community motivieren.

- Hoch
- Mittel
- Tief

Denken Sie, dass die Führungskräfte Ihres Unternehmens für eine Umstellung auf Open Source Software gewappnet wären?

- Ja
- Teilweise
- Nein

Verzichtbarkeit der Mitarbeiter

Denken Sie, dass die Qualität aller Mitarbeiter gesteigert werden kann, wenn durch eine Umstellung auf Open Source gewisse Positionen neu besetzt werden?

Durch eine Umstellung auf Open Source erhält das Unternehmen Zugang zu einem wesentlich grösseren Pool an potentiellen Mitarbeitern.

- Ja
- Teilweise
- Nein

Besitzt Ihr Unternehmen viele unverzichtbare Mitarbeiter?

- Viele
- Wenige
- Keine

Relatives Fähigkeitsniveau

Wie hoch schätzen Sie das Fähigkeitsniveau der Mitarbeiter Ihres Unternehmens verglichen mit der Konkurrenz ein?

In diesem Fall wird lediglich die reine Leistungsfähigkeit der Mitarbeiter im Vergleich zur Konkurrenz betrachtet. Soziodemografische Faktoren wie Alter, Geschlecht, etc. spielen keine direkte Rolle.

- Hoch
- Mittel
- Tief

Wie leistungsfähig schätzen Sie die Mitarbeiter Ihres Unternehmens verglichen mit der Konkurrenz ein?

- Besser
- Gleich
- Schlechter

Motivation

Wie motiviert schätzen Sie die Mitarbeiter Ihres Unternehmens ein, für Ihr Unternehmen zu arbeiten?

- Motiviert
- Durchschnittlich motiviert
- Unmotiviert

Wie viele Mitarbeiter verlassen jährlich Ihr Unternehmen, weil sie unzufrieden sind?

- Wenige
- Durchschnittlich viele
- Viele

Individuum**Produktivität**

Wie schätzen Sie die Produktivität der Mitarbeiter in Ihrem Unternehmen ein?

- Tief
- Mittel
- Hoch

Denken Sie, dass die Mitarbeiter in Ihrem Unternehmen produktiv sind?

- Nein
- Teilweise
- Ja

Auslastung

Wie schätzen Sie die Arbeitsauslastung der Mitarbeiter in Ihrem Unternehmen ein?

- Tief
- Mittel
- Hoch

Die Mitarbeiter meines Unternehmens haben einen stressigen Job.

- Trifft nicht zu
- Trifft teilweise zu
- Trifft zu

TeamzusammenhaltWie schätzen Sie den Zusammenhalt der Mitarbeiter in Ihrem Unternehmen ein?

- Hoch
- Mittel
- Tief

Haben Sie das Gefühl, dass sich die Mitarbeiter in Ihrem Unternehmen bei Problemen gegenseitig aushelfen?

- Ja
- Teilweise
- Nein

KooperationsbereitschaftWie schätzen Sie die Kooperationsbereitschaft der Mitarbeiter in Ihrem Unternehmen im Falle einer Migration ein?

Diese Frage befasst sich mit der Teilnahme bei einer bevorstehenden Umstellung auf Open Source.

- Hoch
- Mittel
- Tief

Denken Sie, dass sich die Mitarbeiter Ihres Unternehmens im Falle einer Migration weigern würden, diese zu unterstützen?

- Nein
- Teilweise
- Ja

Branche

Länge des Produktlebenszyklus

Haben Sie in der Branche, in der Ihr Unternehmen tätig ist, genügend Zeit, um neue Produkte zu entwickeln?

Diese Frage bezieht sich auf das Lifecycle-Management eines einzelnen Produktes.

- Zu wenig
- Die Zeit ist knapp
- Genügend

Gibt es in Ihrem Unternehmen oft Probleme, weil die Entwicklung neuer Produkte zu lange dauert?

- Ja
- Teilweise
- Nein

Wettbewerbsintensität

Wie schätzen Sie den Wettbewerb in der Branche, in der Ihr Unternehmen tätig ist, ein?

- Hoch
- Mittel
- Tief

Mein Unternehmen hat Mühe, sich gegen Konkurrenten zu behaupten.

- Trifft zu
- Trifft teilweise zu
- Trifft nicht zu

Anzahl Konkurrenten

Gibt es in der Branche, in der Ihr Unternehmen tätig ist, viele Konkurrenten, die Open Source Software anbieten?

- Wenige
- Mittel
- Viele

In der Branche, in der mein Unternehmen tätig ist, gibt es nicht viele Konkurrenten, die Open Source Software anbieten.

- Trifft zu
- Trifft teilweise zu
- Trifft nicht zu

Erfolgreiche Migrationen

Wie viele Beispiele erfolgreicher Migrationen auf Open Source Software gibt es in der Branche, in der Ihr Unternehmen tätig ist?

- Viele
- Wenige
- Keine

In der Branche, in der mein Unternehmen tätig ist, gibt es viele Experten, die Erfahrung mit Softwaremigrationen haben.

- Trifft zu
- Trifft teilweise zu
- Trifft nicht zu

Juristische & wirtschaftliche Einflüsse**Unterstützung durch Regierung**

Open Source Software wird von der Landesregierung unterstützt.

- Stark
- Mittel
- Wenig

Wie schätzen sie die Unterstützung ein, die Open Source von der Landesregierung erhält?

- Gross
- Mittel
- Gering

Einschränkungen im Einsatz

Gibt es Einschränkungen, die den Einsatz von Open Source betreffen?

Unter Einschränkungen werden Gesetze, Branchenstandards, etc. verstanden.

- Keine
- Kaum
- Viele

Die Gesetzeslage im Land meines Unternehmens erlaubt einen problemlosen Einsatz von Open Source Software.

- Trifft zu
- Trifft teilweise zu
- Trifft nicht zu

Wirtschaftlicher Stand des Landes

Wie schätzen Sie den wirtschaftlichen Stand des Landes in dem Ihr Unternehmen tätig ist im internationalen Vergleich ein?

- Gut
- Mittel
- Schlecht

Durch eine Umstellung auf Open Source Software könnten viele Unternehmen in meinem Land den Rückstand auf die Industrienationen verringern.

- Trifft zu
- Trifft teilweise zu
- Trifft nicht zu

Abbildungsverzeichnis

Abbildung 1 Intrinsische Motivatoren	11
Abbildung 2 Extrinsisch, internalisierte Motivatoren	12
Abbildung 3 Extrinsische Motivatoren	14
Abbildung 4 Meistverwendete Open Source Lizenzen	28
Abbildung 5 Auswirkungen der Offenheit von Quellcode	41
Abbildung 6 Umweltebenen des Unternehmens	48
Abbildung 7 Analyseframework für Open Source Migration	49
Abbildung 8 Organisationsebene (Nutzer)	51
Abbildung 9 Organisationsebene (Entwickler).....	56
Abbildung 10 Personalebene	60
Abbildung 11 Umweltebene	63
Abbildung 12 Vorteile von Open Source	92
Abbildung 13 Nachteile von Open Source	93

Tabellenverzeichnis

Tabelle 1 Vor- und Nachteile einer OSS-Migration für Unternehmen.....	16
Tabelle 2 Vor- und Nachteile von OSSE	36
Tabelle 3 OSS beim BIT	80

Abkürzungsverzeichnis

BIT	Bundesamt für Informatik und Telekommunikation
BSD	Berkeley Software Distribution (License)
GPL	GNU General Public License
ISB	Informatikstrategieorgan Bund
LGPL	Lesser General Public License
OSS	Open Source Software
OSSE	Open Source Softwareentwicklung (<i>engl. Open Source Software Development, OSSD</i>)

Literaturverzeichnis

- BITKOM. (2015). *Open Source Software - Rechtliche Grundlagen und Hinweise*. Berlin: BITKOM Fachausschuss AGB.
- Bitzer, J., & Schröder, P. J. (2005). *The Impact of Entry and Competition by Open Source Software on Innovation Activity*. Berlin: International Business Section.
- Black Duck Software Inc. (19. Oktober 2013). *Top 20 Open Source Licenses*. Abgerufen am 6. Mai 2015 von <https://www.blackducksoftware.com/resources/data/top-20-open-source-licenses>
- Bundesamt für Informatik und Telekommunikation. (2015). *Das BIT*. Abgerufen am 17. Juni 2015 von www.bit.admin.ch
- Bundesinformatikverordnung (BinfV). (1. Januar 2012). Verordnung über die Informatik und Telekommunikation in der Bundesverwaltung. Bern, Schweiz.
- Comino, S., Manenti, F. M., & Rossi, A. (2011). Public intervention for Free/Open Source Software. *Revue d'économie industrielle*(136), S. 89-108.
- Crocoll, S. (11. März 2011). *Zeit Online*. Abgerufen am 24. Juni 2015 von Maximierer des Sinns: <http://www.zeit.de/2011/11/Arbeit-Motivation>
- Diffe, W. (7. Februar 2003). *Proprietär vs. Open Source: Eine Frage der Sicherheit*. Abgerufen am 24. Juli 2015 von ZDNet: <http://www.zdnet.de/2130055/proprietar-vs-open-source-eine-frage-der-sicherheit/>
- Dobusch, L. (2014). Wert und Wandel von Open Source im öffentlichen Sektor. *Kommune21*(3).
- Eid, M., Gollowitzer, M., & Schmitt, M. (2011). *Statistik und Forschungsmethoden Lehrbuch*. Weinheim: Beltz Verlag.
- Eidgenössisches Finanzdepartement. (2012a). *Leistungsauftrag BIT 2013-2015*. Bern: Schweizerische Eidgenossenschaft.
- Eidgenössisches Finanzdepartement. (2012b). *Mission BIT*. Abgerufen am 30. Juli 2015 von Bundesamt für Informatik und Telekommunikation: <http://www.bit.admin.ch/org/04522/index.html?lang=de>

- Ernst & Young AG. (2011). *Open Source Software im geschäftskritischen Einsatz*. Zürich: Ernst & Young AG.
- Evans, D. S., & Reddy, B. J. (2003). Government Preferences for Promoting Open-Source Software: A Solution in Search of a Problem. *Michigan Telecommunications and Technology Law Review*, 9(2), S. 365-394.
- Free Software Foundation. (2015a). *GNU Operating System*. Abgerufen am 19. Juni 2015 von About the GNU Operating System:
<http://www.gnu.org/gnu/about-gnu>
- Free Software Foundation. (2015b). *GNU Operating System*. Abgerufen am 17. Juli 2015 von What is Copyleft:
<http://www.gnu.org/copyleft/copyleft.en.html>
- Fuggetta, A. (2003). Open source software - an evaluation. *The Journal of Systems and Software*, 66, S. 77-90.
- Gartner. (31. März 2014). *Gartner Says Worldwide Software Market Grew 4.8 Percent in 2013*. Abgerufen am 26. Juni 2015 von
<http://www.gartner.com/newsroom/id/2696317>
- Gates, W. H. (3. Februar 1976). *An Open Letter to Hobbyists*. Abgerufen am 12. Mai 2015 von
<http://www.blinkenlights.com/classiccmp/gateswhine.html>
- Hars, A., & Ou, S. (2002). Working for Free? - Motivations of Participating in Open Source Projects. *International Journal of Electronic Commerce*, 6(3), S. 25-39.
- Hauge, O., Ayala, C., & Conradi, R. (2010). Adoption of open source software in software-intensive organizations - A systematic literature review. *Information and Software Technology* 52, S. 1133-1154.
- Heise Medien GmbH. (14. März 2002). *Ältestenrat stimmt für Linux auf Bundestags-Servern*. Abgerufen am 4. Mai 2015 von
<http://www.heise.de/newsticker/meldung/aeltestenrat-stimmt-fuer-Linux-auf-Bundestags-Servern-61387.html>
- Henkel, J. (2008). Open-Source-Software: Grundlagen. In *Offene Innovationsprozesse* (S. 57-90). München: Gabler.

- Herrmann, W. (19. Juni 2008). *Warum Open Source zum Milliardenmarkt wird*. Abgerufen am 24. Juli 2015 von Computerwoche:
<http://www.computerwoche.de/a/warum-open-source-zum-milliardenmarkt-wird,1866969>
- Herzberg, F., Mausner, B., & Snyderman, B. B. (1959). *The Motivation to Work* (2. Ausg.). New York: Wiley.
- Informatikstrategieorgan Bund. (15. März 2005). OSS-Strategie der Bundesverwaltung. Bern, Schweiz.
- Intel Corporation. (2015). *01.org Intel Open Source*. Abgerufen am 16. Juli 2015 von About: <https://01.org/about>
- Jähnert, J. (10. Februar 2012). Welche Bedeutung wird Business Open Source im öffentlichen Sektor haben? (A. Huber, Interviewer)
- Khanjani, A., & Sulaiman, R. (2011a). The Process of Quality Assurance under Open Source Software Development. *Computer & Informatics (ISCI)* (S. 548-552). Kuala Lumpur: IEEE.
- Khanjani, A., & Sulaiman, R. (2011b). The Aspects of Choosing Open Source versus Closed Source. *Computer & Informatics (ISCI)* (S. 646-649). Kuala Lumpur: IEEE.
- Kollock, P. (1999). The Economics of Online Cooperation: Gifts and Public Goods in Cyberspace. In M. A. Smith, & P. Kollock, *Communities in Cyberspace* (S. 220-242). London: Routledge.
- Koru, A. G., Zhang, D., & Liu, H. (2007). Modeling the Effect of Size on Defect Proneness for Open-Source Software. *International Conference on Software Engineering Companion* (S. 115-124). Minneapolis: IEEE.
- Lakhani, K. R., & von Hippel, E. (2003). How open source software works: "free" user-to-user assistance. *Research Policy*(32), S. 923-943.
- Larsen, M. H., Holck, J., & Pedersen, M. K. (2004). *The Challenges of Open Source Software in IT Adoption: Enterprise Architecture versus Total Cost of Ownership*. Frederiksberg: Copenhagen Business School.
- Lerner, J., & Tirole, J. (2001). The Open Source Movement: Key Research Questions. *European Economic Review*, 45(4-6), S. 819-826.
- Lerner, J., & Tirole, J. (2002). Some Simple Economics of Open Source. *The Journal of Industrial Economics*, 50(2), S. 197-234.

- Lerner, J., & Tirole, J. (2005a). The Economics of Technology Sharing: Open Source and Beyond. *Journal of Economic Perspectives*, 19(2), S. 99-120.
- Lerner, J., & Tirole, J. (2005b). The Scope of Open Source Licensing. *The Journal of Law, Economics & Organization*, 21(1), S. 20-56.
- Lindner, M. (18. September 2013). *Pro-Linux.de*. Abgerufen am 5. Mai 2015 von IBM investiert eine Milliarde in Linux auf Power-System:
<http://www.pro-linux.de/news/1/20262/ibm-investiert-eine-milliarde-in-linux-auf-power-systemen.html>
- Marwan, P. (4. November 2010). *ZDNet*. Abgerufen am 3. Juni 2015 von 100 Prozent Open Source ist in Unternehmen machbar:
<http://www.zdnet.de/41540236/100-prozent-open-source-ist-in-unternehmen-machbar/>
- Mijinyawa, M. K., & Abdulwahab, L. (2014). An Extended Framework for Evaluation of Open Source Software Adoption in Small Business. *Research Journal of Information Technology*, 6(4), S. 248-269.
- Moody, G. (2001). *Rebel Code: Linux And The Open Source Revolution* (1. Ausg.). Cambridge MA: Perseus Publishing.
- Moosbrugger, H., & Kelava, A. (2012). *Testtheorie und Fragebogenkonstruktion* (2. Ausg.). Berlin: Springer-Verlag.
- Murad, A. (31. März 2009). *Microsoft accepts defeat to Wikipedia and kills off Encarta*. Abgerufen am 27. Juli 2015 von The Times:
<http://www.thetimes.co.uk/tto/technology/article1859433.ece>
- National Security Agency. (4. Mai 2015). *National Security Agency Shares Security Enhancements to LINUX*. Von https://www.nsa.gov/public_info/press_room/2001/se-linux.shtml abgerufen
- Open Source Initiative. (2015a). *The Open Source Definition*. Abgerufen am 25. Juni 2015 von <http://opensource.org/osd>
- Open Source Initiative. (2015b). *Licenses by Name*. Abgerufen am 25. Juni 2015 von <http://opensource.org/licenses/alphabetical>
- Open Source Studie Schweiz. (2012). *Open Source Studie Schweiz*. Zürich: SwissICT, Swiss Open Systems User Group /ch/open.

- Open Source Studie Schweiz. (2015). *Open Source Studie Schweiz*. Zürich: SwissICT, Swiss Open Systems User Group /ch/open.
- Osterloh, M., Rota, S., & Kuster, B. (2004). Open-Source-Softwareproduktion: Ein neues Innovationsmodell? In R. A. Gehring, & B. Lutterbeck, *Open-Source-Jahrbuch 2004*. Berlin: Lehmanns Media.
- Phipps, S. (22. August 2014). *InfoWorld*. Abgerufen am 6. Mai 2015 von Walmart's investment in open source isn't cheap: <http://www.infoworld.com/article/2608897/open-source-software/walmart-s-investment-in-open-source-isn-t-cheap.html>
- Pingdom Royal. (28. Februar 2012). *Webseite*. Abgerufen am 20. Mai 2015 von <http://royal.pingdom.com/2012/02/28/linux-is-the-worlds-fastest-growing-desktop-os-up-64-percent-in-9-months/>
- Qu, W. G., Yang, Z., & Wang, Z. (2011). Multi-level framework of open source software adoption. *Journal of Business Research* 64, S. 997-1003.
- Raymond, E. S. (1998). Homesteading the Noosphere. *First Monday*, 3(10). Abgerufen am 27. Juli 2015 von <http://firstmonday.org/ojs/index.php/fm/article/view/621/542>
- Raymond, E. S. (2001). *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol: O'Reilly Media.
- Riehle, D. (2007). The Economic Motivation of Open Source Software: Stakeholder Perspective. *IEEE Computer*, 40(4), S. 25-32.
- Rüegg-Stürm, J., & Grand, S. (2014). *Das St. Galler Management-Modell*. St. Gallen: Haupt Verlag.
- Schweizer Bundesversammlung. (2007). *Curia Vista*. Abgerufen am 17. Juli 2015 von Interpellation 07.3887: http://www.parlament.ch/d/suche/seiten/geschaefte.aspx?gesch_id=20073887
- Schweizer Bundesversammlung. (2009). *Curia Vista*. Abgerufen am 17. Juli 2015 von Interpellation 09.3584: http://www.parlament.ch/d/suche/seiten/geschaefte.aspx?gesch_id=20093584

- Schweizer Bundesversammlung. (2014a). *Curia Vista*. Abgerufen am 17. Juli 2015 von Interpellation 14.3563:
http://www.parlament.ch/d/suche/seiten/geschaefte.aspx?gesch_id=20143563
- Schweizer Bundesversammlung. (2014b). *Curia Vista*. Abgerufen am 17. Juli 2015 von Interpellation 14.4199:
http://www.parlament.ch/d/suche/seiten/geschaefte.aspx?gesch_id=20144199
- Software Package Data Exchange. (6. Mai 2015). *SPDX License List*. Von <http://spdx.org/licenses/> abgerufen
- Stadt München. (26. Januar 2015). *Das Projekt LiMux*. Von <http://www.muenchen.de/rathaus/Stadtverwaltung/Direktorium/LiMux.html> abgerufen
- Stark, J. (26. Februar 2014). *7 Gründe gegen Open Source*. Abgerufen am 2. Juli 2015 von www.computerworld.ch:
<http://www.computerworld.ch/news/software/artikel/7-gruende-gegen-open-source-software-65290/>
- Stürmer, M. (2009). *Open Source Software und die öffentliche Hand*. Zürich: Swiss Open Systems User Group /ch/open.
- Stürmer, M. (20. Mai 2010). *Parlamentarische Gruppe Digitale Nachhaltigkeit (Parldigi)*. Abgerufen am 17. Juni 2015 von Parlamentarische Gruppe Digitale Nachhaltigkeit: «Kostendruck fordert den Umstieg auf Open Source Software»: <http://www.digitale-nachhaltigkeit.ch/2010/05/skr-artikel-2/>
- Swiss Open Systems User Group /ch/open. (2015). *Über das OSS-Directory*. Abgerufen am 7. Juli 2015 von OSS-Directory:
<http://www.opensource.ch/chopen-initiativen/single/init/oss-directory/>
- Thomas Krenn. (Mai 17 2015). *Das Thomas-Krenn-Wiki - mehr als nur ein Lexikon*. Von <https://www.thomas-krenn.com/de/wiki/Hauptseite> abgerufen
- Vogel, M. (6. Mai 2015). *Kleines EDV-Lexikon*. Von <http://lexikon.martinvogel.de/> abgerufen

-
- Volpi, M. (25. März 2014). *Index Ventures*. Abgerufen am 25. Juni 2015 von
A Perfect Storm: Now is the Time for Multibillion Dollar Open Source
Companies: <https://indexventures.com/news-room/blog/a-perfect-storm-now-is-the-time-for-multibillion-dollar-open-source-companies>
- Von Krogh, G., Häfliger, S., Späth, S., & Wallin, M. W. (Juni 2012). Carrots
and Rainbows: Motivation and Social Practice in Open Source
Software Development. *MIS Quarterly*, 36(2), S. 649-676.
- WhatIs.com. (17. Mai 2015). *Computer Glossary, Computer Terms*. Von
<http://whatis.techtarget.com/> abgerufen
- Wirtschaftslexikon24.com. (18. Mai 2015). *Webseite*. Von
<http://www.wirtschaftslexikon24.com/> abgerufen

Selbständigkeitserklärung

„Ich erkläre hiermit, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen wurden, habe ich als solche gekennzeichnet. Mir ist bekannt, dass andernfalls der Senat gemäss Artikel 36 Absatz 1 Buchstabe o des Gesetzes vom 5. September 1996 über die Universität zum Entzug des aufgrund dieser Arbeit verliehenen Titels berechtigt ist.“

Ort, Datum:

Unterschrift:

Bern, 07.08.2015

Roman Schmid

Veröffentlichung der Arbeit

I.d.R. werden schriftliche Arbeiten in der Bibliothek des Instituts für Wirtschaftsinformatik öffentlich zugänglich gemacht.

- Hiermit erlaube ich, meine Arbeit in der Bibliothek des Instituts für Wirtschaftsinformatik zu veröffentlichen.
- Ich möchte auf eine Veröffentlichung meiner Arbeit verzichten.

Falls eine Vertraulichkeitserklärung unterschrieben wurde, ist es Sache des Studierenden, das Einverständnis des Praxispartners einzuholen. Es muss der Arbeit eine schriftliche Bestätigung des Praxispartners beigelegt werden.

Die Benotung der Arbeit erfolgt unabhängig davon, ob die Arbeit veröffentlicht werden darf oder nicht.

Ort, Datum:

Unterschrift:

Bern, 07.08.2015

Roman Schmid