

Open Source SW im industriellen Einsatz

Norbert Kraft

Research Technology and Platforms
Nokia Siemens Networks

81541 München
norbert.kraft@nsn.com

Abstract: Die Verwendung von Open Source SW hat ein solch ungeheures Ausmaß angenommen, dass deren Einsatz in vielen Bereichen der Industrie heute nicht mehr wegzudenken ist. In der allgemeinen Wahrnehmung wird Open Source SW oftmals noch als Randerscheinung des universitären und nichtkommerziellen Bereiches angesehen. In der Realität hingegen hat Open Source SW längst seinen Siegeszug in die kommerziellen Produkte der Industrie angetreten. Dies passierte in vielen Fällen auch für das verantwortliche Management nahezu unbemerkt. Dieser Artikel beschreibt sowohl Herausforderungen als auch Chancen für die Industrieunternehmen, die durch den Einsatz von Open Source SW in der Produktentwicklung entstehen, deren Auswirkungen auf Geschäftsprozesse, die erforderlichen Maßnahmen in der Unternehmensorganisation, sowie Einflüsse auf zukünftiges Entwicklungsvorgehen und Projektmanagement.

1. Open Source SW in der Industrie – Chancen, Risiken, Vor- und Nachteile

Zu Beginn unserer Arbeit stand die Analyse des Open Source Gedankens aus der Sicht der Industrie. Hierbei kam die SWOT-Analyse (*Strengths, Weaknesses, Opportunitites & Threats*) zum Einsatz mit dem Ziel alle Vor- und Nachteile, die für oder gegen die Verwendung von Open Source SW in der industriellen Produktentwicklung sprechen, vorurteilsfrei zu ermitteln, und die entsprechenden Maßnahmen für die NSN Organisation abzuleiten.

Die Stärken der Open Source SW sind vergleichsweise einfach zu beschreiben. Unzählige erfolgreiche Beispiele (Linux, OpenOffice, Gnu Compiler ...) belegen, dass selbst hochkomplexe Systeme erfolgreich in einer *Community* entwickelt werden können und zu tragfähigen und stabilen Lösungen führen.

Darüber hinaus etabliert sich der Open Source Gedanke auch immer mehr als Innovationsträger und Kristallisationspunkt für neue Entwicklungen und wird damit unverzichtbar für Unternehmen, um an Innovationen teilzuhaben.

Oftmals vernachlässigt werden auch die Erfolge des Open Source Gedankens zum Thema *Reuse von Software*. Open Source Produkte werden mit hoher Akzeptanz seitens der Entwickler wiederverwendet, wo hingegen die Wiederverwendung von Software im eigenen Unternehmen, über Projektgrenzen hinweg, immer wieder auf unumstößliche Schwierigkeiten stößt.

Neben all diesen positiven Aspekten darf aber nicht verschwiegen werden, dass die Verwendung von Open Source SW im industriellen und kommerziellen Bereich durchaus mit Schwierigkeiten und Risiken behaftet ist, die einer sorgfältigen Prüfung bedürfen.

Das komplexe Lizenzmodell, die hohe Zahl unterschiedlicher Open Source Lizenzen und deren vermeintliche Missinterpretation führen zu vielen Unsicherheiten in der Open Source Verwendung. Mangelndes Wissen bezüglich der auftretenden Rechtsfragen, fehlende Interpretationen, wie diese zu handhaben sind, und anstehende Gerichtsentscheidungen verunsichern darüber hinaus viele Beteiligte.

Open Source hat aber auch neue Wege der globalen Zusammenarbeit geschaffen, die neben einer beeindruckenden Effizienz aber auch Unwägbarkeiten aufzeigen. Das Verhalten von *Communities* ist für den industriellen Nutzer nicht immer beeinflussbar, Meinungsdivergenzen über den *richtigen Weg*, wechselnde Beteiligte, machen in bestimmten Fällen den weiteren Fortschritt und Erfolg eines Open Source Projektes nicht immer vorhersehbar.

Bedingt durch dieses dynamische Verhalten, neigt Open Source SW im Allgemeinen zu einer starken Fragmentierung. Dies ist einerseits von Vorteil, da es zu vielen unterschiedlichen konkurrierenden Lösungen kommt. Es führt aber andererseits zu einer Zersplitterung von wertvollen *Resources* und auch zu einer mangelnden Übersichtlichkeit. Die Zahl von über 150.000 *Sourceforge* Projekten spricht hier Bände. Allein diese große Anzahl lässt schon vermuten, dass Reife und Wert, der darin enthaltenen Projekte, sehr stark unterschiedlich sind.

Diese starke Fragmentierung lässt sich auch am Beispiel Linux sehr gut nachvollziehen. Die Vielzahl der existierenden Linux Distributionen, deren unterschiedliche Desktops, sowie bestimmte Unterschiede bei der SW Installation durch Pakete, führen dazu, dass es genau *das Linux* nicht mehr gibt. Aus diesem Grund dürfte es auch heute einem SW Hersteller schwerfallen, genau mit einer einzigen Variante seines SW Produktes all diese Derivate zu unterstützen.

2. Das Open Source SW Maturity Model im Unternehmen

Neben einer Bewertung von Chancen und Risiken ist die eigene Positionsbestimmung in Bezug auf die Verwendung von Open Source SW unumgänglich für jedes Unternehmen. Hierzu bedarf es einer Analyse, inwieweit Open Source SW bereits Verwendung findet und in welchen Bereichen.

Generell kann man unterscheiden: Die Verwendung der Open Source SW in der IT Infrastruktur des Unternehmens (z.B. Linux Desktops, Entwicklungswerkzeuge ...), oder aber als Bestandteil von selbst entwickelten, kommerziellen Produkten.

Insbesondere Letzteres erfordert ein erhebliches Maß an Aufmerksamkeit im Unternehmen und ein sorgfältiges Risikomanagement. Anhand eines einfachen, von uns entwickelten Stufenmodells kann man diese Positionsbestimmung durchführen und daraus Maßnahmen ableiten.

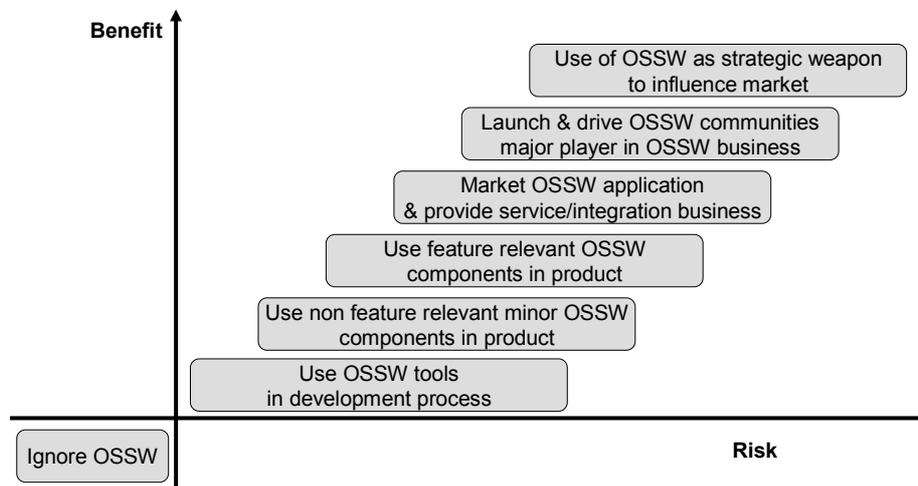


Abbildung 1: Open Source SW Maturity Model

Dieses Stufenmodell beschreibt die Entwicklung eines potentiellen Unternehmens in der Nutzung von Open Source SW Komponenten. Nach der Phase des *Ignorierens* (Stufe -1) sind viele Unternehmen dazu übergegangen, Open Source SW Komponenten in der Entwicklung bzw. in der eigenen IT Infrastruktur (Stufe 1) zu verwenden. Dieser Schritt ist hierbei als relativ unkritisch zu sehen, weil das eigene Produkt noch nicht direkt mit Open Source SW ‚belastet‘ wird.

Sobald Open Source SW Komponenten Einzug halten in die Produkte des Unternehmens, müssen alle Lizenzverpflichtungen, die damit einhergehen, sorgfältig beachtet werden. Sind bereits große Architektur bestimmende Teile (Stufe 2) des eigenen Produktes bereits Open Source, entsteht eine starke, externe Abhängigkeit zu Open Source SW *Communities*, die gezielte Maßnahmen zur Risikominimierung erfordert.

Open Source Applikationen können auch selbst zum Geschäftsinhalt (Stufe 4) werden. Die eigentliche Geschäftstätigkeit konzentriert sich hier meist auf Wartung, Pflege und Adaption. Hierbei tritt der Verkauf von SW gegen Lizenzgebühren ins Hintertreffen und das vorrangige Geschäftsziel besteht im Erbringen von Integrationsleistungen.

In den beiden oberen Stufen des Maturity Modells agieren Firmen selbst aktiv in Open Source *Communities*. Dies dient zum Einen, um sich als Technologieführer in bestimmten Segmenten zu etablieren oder auch zur Erreichung von strategischen Zielen. Letzteres kann beispielsweise der Etablierung von De-facto-Standards, oder aber der Entwertung von Konkurrenzprodukten dienen.

Auch die Motive der Firmen ändern sich auf den unterschiedlichen Stufen des Maturity Modells. Gerade Marktführer in renditestarken Industrien können sich die Missachtung der Open Source Bewegung durchaus noch leisten. Sobald sich aber der Kostendruck erhöht, gelangt immer mehr Open Source SW in die Unternehmen und deren Produkte. Hier spielen insbesondere kostengetriebene Argumentationen eine Rolle. Diese Sichtweise verändert sich dann aber zusehends auf den oberen Stufen des Modells in Richtung einer stark strategischen Sichtweise mit Auswirkungen auf die praktizierten und geplanten Geschäftsmodelle.

3. Auswirkungen der Open Source SW Nutzung auf industrielle Geschäftsprozesse

Die Anpassung von Geschäftsprozessen als Reaktion auf die firmeninterne Nutzung und die Verbreitung von Open Source SW Komponenten als Bestandteil der eigenen Produkte dient dem Zweck, den offensichtlichen Nutzen zu optimieren und vorhandene Risiken zu vermeiden. Die nachfolgend beschriebenen Schritte dienen als Empfehlung für eine Vorgehensweise, um die existierenden Geschäftsprozesse an diese neuen Anforderungen anzupassen.

3.1 Formulierung einer unternehmensspezifischen *Open Source Governance Policy*

Jedes Unternehmen sollte eigene Regeln und Vorgehensweisen in Bezug auf die Verwendung von Open Source SW Komponenten in den eigenen Produkten festlegen. Diese Regeln sind bindend und sollten den gleichen Stellenwert im Unternehmen haben, wie z.B. die Überwachung und Einhaltung der üblichen gesetzlichen Bestimmungen.

Es ist sowohl festzulegen, in welchem Rahmen Open Source SW verwendet werden kann, als auch unter welchen Umständen Source Code des Unternehmens an externe *Communities* weitergegeben werden darf. Diese Regeln sind verbindlich sowohl für die eigenen Mitarbeiter als auch für im Auftrag handelnde Unternehmen oder Consultants. Ein zu installierendes Gremium überwacht diese Regeln und trifft die erforderlichen Entscheidungen.

Da Source Code eines der ureigenen Vermögenswerte eines Unternehmens darstellt, ist die Schaffung eines leistungsfähigen *Open Source SW Asset Managements* erforderlich. Hierin wird Buch geführt, welche Open Source SW Komponenten im Unternehmen eingesetzt werden bzw. welche Teile der Unternehmenssoftware bereits an Open Source Communities weitergegeben wurden.

Die Akquisition von Open Source SW ist prinzipiell mit jeder anderen externen, kommerziellen Software gleichzusetzen, was Auswahl und Test angeht. Spielt bei kommerziellen Angeboten meist die Aushandlung finanzieller Konditionen eine große Rolle, tritt bei Open Source Akquisitionen oftmals die Frage nach einem kommerziellen Support mit einem definierten *Service Level Agreement* in den Vordergrund.

Für diese Support Aufgaben kommen insbesondere Unternehmen in Betracht, die eine gute Beziehung zu relevanten *Communities* haben und auch dort über eine entsprechende Reputation verfügen.

3.2 Definition eines Open Source SW Prozesses in der Entwicklung

Der Open Source SW Prozess legt die einzelnen Schritte und Maßnahmen fest, die vor der Verwendung von Open Source SW Komponenten in den Produkten des Unternehmens durchzuführen sind. Dazu zählen: Die Prüfung der verwendeten Komponenten hinsichtlich der urheberrechtlichen Fragen, die Ermittlung der angewendeten Lizenzen und deren Auswirkung auf das Produkt, und die Festlegung der daraus resultierenden Lizenzverpflichtungen (z.B. optionale Bereitstellung von Source Codes für Kunden, Publikationspflichten, Garantiausschluss).

Neben lizenzrechtlichen Fragen sind unter Umständen auch patentrechtliche von Bedeutung. Die grundsätzliche Prüfung auf Patentverletzungen, die im Open Source Code (sog. *IPR Infringement*) enthalten sein könnten, wird dringend empfohlen. Auch muss verhindert werden, dass eigene anhängige Patente durch die Veröffentlichung als Open Source SW möglicherweise entwertet werden.

Die Beurteilung patent- und lizenzrechtlicher Fragen ist meist die Aufgabe speziell geschulter Experten und sollte daher nicht von Software Entwicklern, mangels erforderlicher Spezialkenntnisse, durchgeführt werden.

3.3 Schulung aller betroffenen Mitarbeiter

Keine der beschriebenen Maßnahmen ist von Wert, wenn sie nicht in der eigenen Organisation verstanden und auch gelebt werden. Unsere Erfahrungen zeigen, dass das Wissen bezüglich Open Source SW Lizenzierung, den rechtlich Folgen und Risiken, sowie den daraus entstehenden Verpflichtungen des Nutzers, insbesondere bei Entwicklern schwach ausgeprägt oder zumindest lückenhaft ist.

Diese Schulungen sind auch deshalb schwierig, weil hier technische und juristische Fragen miteinander konfrontiert werden. Als Beispiel möge hier die Frage dienen, ob eine ‚Java Subclass‘ als ein ‚derived work‘ im Sinne des Copyrights anzusehen ist. Im Rahmen einer solchen Diskussion fehlt Juristen meist das technische Verständnis für den Sachverhalt, hingegen Techniker werden sich mit dem tieferen Verständnis des Urheberrechts schwer tun.

Grosse Konzerne verfügen aus diesem Grund über spezialisierte Juristen für die Bewertung dieser Fragen. Kleinere Unternehmen sollten sich auf dem freien Markt in die entsprechende Expertise bemühen.

3.4 Verankerung von Open Source spezifischen Vorgehensweisen in Begleitprozesse

Im Anschluss an die Verankerung von Open Source Aspekten in den firmeneigenen Entwicklungsprozessen sollte die Aufmerksamkeit auch auf andere Begleitprozesse gelenkt werden. Die Verwendung von Open Source SW hat beispielsweise einen Einfluss auf die Produktauslieferung. So müssen Kunden auf Antrag bestimmte Source Codes und Werkzeuge zur Verfügung gestellt werden.

Darüber hinaus unterliegt Open Source SW den Exportkontrollrichtlinien der Länder und muss vor der Ausfuhr sorgfältig klassifiziert werden.

Der Einsatz Open Source SW hat selbst einen Einfluss auf die Gestaltung von Arbeitsverträgen. Fragen des anzuwendenden Copyrights bei Veröffentlichungen in Open Source Communities sind ebenso zu beantworten, wie die mögliche Veröffentlichung von Open Source SW durch Mitarbeiter in der Freizeit, die in einem Zusammenhang steht mit den innerbetrieblichen Arbeiten.

Wie bereits erwähnt, muss die Handhabung, Publikation und Nutzung von Open Source SW auch in Dienstleistungsverträgen mit externen Zulieferern Berücksichtigung finden.

Ein Thema von besonderer Brisanz hat das Thema Open Source SW auch im Rahmen von Firmenzusammenschlüssen oder Akquisitionen. Die Bewertung des vorhandenen Vermögens eines Unternehmens, sowie vorhandene Produktrechte und bestehende Verpflichtungen sind maßgeblich vom Anteil an Open Source SW in den Produkten des Unternehmens beeinflusst.

3.5 Aktive Mitarbeit in einer Open Source Community

Viele Unternehmen nutzen Open Source SW in einer passiven Art und Weise, es gibt aber vielerlei Gründe hier aktiv an der Entwicklung mitzuarbeiten. Falls selbst Fehlerkorrekturen vorgenommen wurden, kann sich z.B. die Wartung eines Produktes erheblich vereinfachen nach einem erfolgreichen Rückfluss in die *Community*.

Durch aktive Mitarbeit besteht auch die Möglichkeit der Beeinflussung und die Chance, die verwendete Open Source SW Komponente für die eigenen Bedürfnisse zu optimieren. Hierbei darf nicht unterschätzt werden, dass die Zusammenarbeit in diesen Open Source Communities eigenen Gesetzen gehorcht und mit dem oftmals hierarchischen Vorgehen in Industrieunternehmen wenig gemein hat.

Auch erkennen die Unternehmen immer stärker die Bedeutung solcher *Communities* für die eigene Reputation, für die Gewinnung guter Mitarbeiter und zur Durchsetzung eigener strategischer Ziele.

3.6 Lernen vom Entwicklungsvorgehen in Open Source Communities

Unternehmen tun auch gut daran, sich nicht nur auf die Ergebnisse der Open Source Communities zu fokussieren. Diese Communities haben nebenbei auch eine neue Art und Weise des Entwickelns begründet, die durchaus innerhalb von Unternehmen Anwendung finden könnte.

- Communities unterscheiden nicht zwischen Entwicklern und Anwendern, Anwender sind oftmals aktive Entwicklungsteilnehmer und Tester.
- Entscheidungen fallen im Konsensprinzip und Hierarchien *ergeben* sich basierend auf Anerkennung und Kompetenz.
- Communities sind lose gekoppelt, global verteilt und entsprechen daher auch einem in der Industrie gelebten Idealbild.
- Bedingungslose Offenheit in der Verteilung und kurze Release Zyklen erlauben eine hohe Innovationsrate.
- Open Source Communities sind untereinander hochgradig vernetzt und betreiben aktiven *Reuse*.

All die oben genannten Aspekte machen auch im industriellen Zusammenhang Sinn, und machen es interessant, diese Vorgehensweisen und Methoden auf kommerzielle Projekte anzuwenden, auch vor dem Hintergrund, dass die dabei entstehenden Software Produkte nicht als Open Source SW publiziert werden. Für diese Form der Methodenadaption hat sich der Begriff ‚Inner Source‘ etabliert.

4. Zukunftsaussichten

Der Open Source Gedanke wird sich weiter etablieren und zusätzliche Bereiche erobern. So gibt es bereits heute Open Source Hardware, und andere Veröffentlichungen unter der Flagge von ‚Creative Commons‘.

Open Source wird nicht nur Einfluss nehmen auf die Struktur unserer zukünftigen Software Produkte, sondern auch auf die Art und Weise wie wir diese entwickeln. So wird vielleicht bald die Frage für einen SW Hersteller nicht mehr sein: ‚*Make or Buy*‘ sondern: ‚*Make, Buy or go Open Source*‘.

Zukünftig werden Methoden des globalen Entwickelns in lose gekoppelten *Communities* auch verstärkt in der Industrie zu Anwendung kommen und damit das interne Projektmanagement stark verändern.

5. Fazit

Für viele Unternehmen stellt sich heute nicht mehr Frage, ob Open Source SW zum Einsatz kommt, sondern wie dieses Phänomen in die betrieblichen Prozesse zu integrieren ist. Hierbei sind insbesondere die patent- und urheberrechtlichen Fragen relevant.

Open Source wird damit auch zu einem Thema im kommerziellen Umfeld, dem man sich zur Minimierung der damit verbundenen Risiken aber auch zu Nutzung der Chancen aktiv stellen sollte. Eine Strategie zum Thema Open Source SW wird damit zum absoluten Muss für jedes Industrieunternehmen.

6. Literaturverzeichnis

- [FB] Fossbazaar (<https://fossbazaar.org/>) a community for OSSW Governance
- [IFR] IFross (<http://www.ifross.de/>) Institut für Rechtsfragen der freien und Open Source SW
- [GOV] Hewlett Packard: Foss Governance Fundamentals
- [Sca] Walt Scacchi, Joseph Feller, Brian Fitzgerald, Scott Hissan, Karim Lakhani: Understanding Free/Open Source Software Development Processes
- [OSJB] Bernd Lutterbeck, Matthias Bärwolff und Robert A. Gehring : Open Source Jahrbuch 2008 - Zwischen freier Software und Gesellschaftsmodell
- [FHG] Thomas Renner, Michael Vetter, Sascha Rex, Holger Kett: Open Source Software: Einsatzpotenziale und Wirtschaftlichkeit, Eine Studie der Fraunhofer-Gesellschaft
- [Val] Mikko Välimäki: The Rise of Open Source Licensing, a Challenge to the Use of Intellectual Property in the Software Industry
- [BK] BITKOM: Open Source Software, Rechtliche Grundlagen und Hinweise, LEITFADEN
- [EC] Study of European Community: Free/Libre Open Source Software: A Guide for SMEs