Open Source Automation Development Lab (OSADL) eG

Whitepaper "Free and Open Source Software (FOSS)"

www.osadl.org

OSADL

Open-Source-Software und Virtualisierung in der Automatisierungsindustrie - Modeerscheinungen oder Paradigmenwechsel?

von Carsten Emde

Open Source Automation Development Lab (OSADL) eG

Zusammenfassung

Wie in vielen anderen Bereichen hat Open-Source-Software auch in der Automatisierungsindustrie Einzug gehalten und eröffnet auf den ersten Blick viele neue Möglichkeiten, speziell im Hinblick auf die Anforderungen an eine lange Produktlebensdauer von Maschinen und anderen Embedded-Systemen. Doch was ist eigentlich Open-Source-Software genau? Was muss man bedenken, wenn man sie im industriellen Umfeld einsetzt, weitergibt und weiterentwickelt?

Ein durchaus verwandtes Thema stellt die Hardware-Virtualisierung dar. Auch hierbei handelt es sich um eine faszinierende Technik, die das Potenzial hat, viele Probleme im Langzeit-Management industrieller Steuerungen zu lösen. Allerdings stellt sich auch in diesem Zusammenhang die Frage, ob und unter welchen Bedingungen die neuen Möglichkeiten sinnvoll eingesetzt werden können.

Im Folgenden werden die Mechanismen von Open-Source-Software und Virtualisierung erläutert und Richtlinien für deren gezielten Einsatz gegeben.

Was ist Open-Source-Software?

Die Begriffe "Open-Source-Software" und "Free Software" repräsentieren zwar verschiedene ideologische Richtungen, rechtlich und praktisch sind beide Begriffe aber gleichbedeutende Bezeichnungen für eine bestimmte Art einer Software-Lizenzierung. Daher wird in diesem Artikel der gemeinsame Begriff "Freie und Open-Source-Software" (FOSS) für diese Art von Lizenz verwendet. Die meisten aktiven FOSS-Projekte verwenden die sogenannte GNU General Public License (GNU GPL). Diese gestattet dem Anwender, die jeweilige Software

1. uneingeschränkt einzusetzen,
2. zu analysieren und für eigene Bedürfnisse anzupassen,
3. an andere weiterzugeben,
4. zu verändern und die Veränderungen zu veröffentlichen.

Der zweite und der vierte Punkt erfordern freien Zugang zum Quellcode. Dieser Zugang zum Quellcode ist damit eine notwendige aber keine hinreichende Bedingung für die Eigenschaft als Open-Source-Software. Bei Weitergabe der Software an andere muss der Erwerb dieser Rechte, je nach Art der FOSS-Lizenz, auch dem Empfänger ermöglicht werden. Wird die Software nur selbst verwendet und nicht weitergegeben, entfällt jegliche Verpflichtung aus den Open-Source-Lizenzen. Entgegen häufig geäußerter Vorstellung darf die Weitergabe von FOSS durchaus entgeltlich erfolgen, wobei aber klar geregelt ist, dass dieses Entgelt nicht für die Lizenzierung selbst sondern nur für begleitende Leistungen wie zum Beipiel für den mit der Weitergabe verbundenen Aufwand erhoben werden darf. Unabhängig davon darf natürlich ein Dienstleister die von ihm für Weiterentwicklung und Pflege von FOSS erbrachten Tätigkeiten seinem Kunden in Rechnung stellen.

Was ist Copyleft?

Das Wort "Copyleft" ist eine künstliche Wortschöpfung für das "Gegenteil von Copyright". Enthalten ist auch das Wortspiel des englischen Wortes "left", das gleichzeitig das Gegenteil von "right" (rechts/links) und "überlassen" (von to leave = jemandem etwas überlassen) bedeutet. Während das klassische Copyright die Weitergabe eines urheberrechtlich geschützten Werkes verbietet, erlaubt das Copyleft die Weitergabe ("Überlassung") - aber sie erlaubt diese nicht nur, sondern fordert sie sogar bzw. verlangt mindestens die Bereitschaft dazu. Dabei gilt diese Forderung nicht nur für die ursprüngliche Version des Werkes sondern auch für sämtliche daran vorgenommenen Änderungen und Erweiterungen. Letzteres stellt insofern eine besondere Regelung dar, als dass übliche Urheberrechtsregelungen dem Lizenznehmer normalerweise die Entscheidung überlassen, unter welche Lizenz er die vorgenommenen Änderungen und Erweiterungen stellt. Aber mit der Annahme einer Copyleftbeinhaltenden Lizenz akzeptiert der Lizenznehmer diese besondere Auflage, die bei Weitergabe des Werkes in Kraft tritt. Die verschiedenen zur Zeit bekannten Open-Source-Lizenzen unterscheiden sich unter anderem darin, wie stark ausgeprägt dieser Copyleft-Effekt ist. Die BSD-Lizenz (Berkeley Software Distribution) schreibt dem Lizenznehmer in keiner Weise vor, unter welche Lizenz er die überlassene Software

und eventuell daran vorgenommene Änderungen und Erweiterungen stellt; es ist also eine Open-Source-Lizenz ohne Copyleft. Ein Beispiel für eine Open-Source-Lizenz mit maximalem Copyleft ist die bereits genannte GNU GPL; diese zwingt den Lizenznehmer, für die überlassene Software und eventuell daran vorgenommene Änderungen und Erweiterungen ausschließlich die ursprüngliche Lizenz in identischer Form weiterzuverwenden.

Pragmatismus vs. Ideologie

Das Konzept "Freie Software" wurde 1983 von Richard Stallman erstmals vorgestellt und mündete in der Formulierung der genannten GNU GPL. Leitgedanke ist die Vorstellung, dass ein Programm in Maschinensprache, dessen Quellcode dem Benutzer nicht zur Verfügung steht, diesen in eine Form von Unfreiheit versetzt, die aus grundsätzlichen ethischen Überlegungen heraus abzulehnen ist. Nur wenn der Quellcode eines Programms auch verfügbar gemacht wird, ist nach Stallmans Vorstellung die zu fordernde Freiheit gewährleistet.

Der gegen Ende der 1990-er Jahre von Bruce Perens, Eric Raymond und anderen eingeführte alternative Begriff "Open-Source-Software" zielt mehr auf die pragmatische Seite ab. Denn durch die Offenlegung des Quellcodes bildet sich im Vergleich zu proprietärer Software eine ungleich größere Entwicklergemeinde, die zu Softwareprodukten in bis dahin unerreichter Qualität und Stabilität führte.

Etwa seit dieser Zeit wird FOSS in zunehmendem Maße in industriellem Bereich und entsprechend auch in der Automatisierungsindustrie, d.h. in Embedded-Systemen bzw. im Maschinen- und Anlagenbau eingesetzt. Wenn man die Hersteller von Industrieprodukten nach den Gründen für die Bevorzugung von FOSS befragt, hört man nicht selten Argumente wie "vollständige Kontrolle des Maschinenbauers über seine Maschine", "Unabhängigkeit von Abkündigungen" oder "Vermeidung von Vendor-Lock". Daraus lässt sich ableiten, dass die ursprünglichen Beweggründe, die auf die Freiheit des Software-Benutzers abzielten, eine gewisse Berechtigung haben. In der Tat liegt eine der Besonderheiten von FOSS darin begründet, dass nicht wie bei proprietärer Software eine sehr eingeschränkte zeitlich begrenzte Nutzungserlaubnis erteilt wird, sondern dass der Nutzer Rechte erhält, die eine umfassende Nutzungsbefugnis vermitteln - wobei es sich natürlich um eine nicht-exklusive Rechtsposition handelt. Darüber hinaus sind es aber auch hier die genannten pragmatischen Gründe der "Open-Source"-Bewegung, welche diese Software für den In-

dustrieeinsatz attraktiv machen; denn Qualität und Stabilität waren immer schon herausragende Anforderungen an industriell eingesetzte Software.

Offene Wissensökonomie

Vermutlich noch relevanter als die genannte Diskussion um Freiheit und Offenheit sind Überlegungen, das FOSS-Entwicklungsmodell als ökonomisches Konzept zu verstehen. In diesem Zusammenhang hat FOSS inzwischen auch Eingang in den wissenschaftlichen Diskurs volks- und betriebswirtschaftlicher Disziplinen gefunden. Der Wissens- und Erfahrungsschatz eines Unternehmens wird dabei als Kapital unter dem Aspekt der Alleinstellungsfähigkeit betrachtet. Nur ein Teil dieses Kapitals ist nämlich geeignet, Produkte eines Unternehmens von denen eines Mitbewerbers unterscheidbar zu machen ("differenzierendes Know-how"), während ein anderer Teil diese Fähigkeit nicht besitzt ("nicht-differenzierendes Know-how"). Letzterer ist in der Regel der größere Teil. Mit dem Ziel ökonomischer Erfolgsmaximierung ist ein Unternehmen gehalten, möglichst sämtliche Ressourcen in die Entwicklung und Erhaltung des differenzierenden Know-hows zu investieren. In das nicht-differenzierende Know-how sollte möglichst nicht individuell investiert werden. Hierfür bietet sich nun an, im Rahmen von FOSS-Projekten gemeinsam mit anderen Unternehmen und sogar gemeinsam mit Mitbewerbern tätig zu werden. Dies ist möglich, da es sich gerade um nicht-differenzierendes Know-how handelt. Die Kosteneinsparung liegt dabei auf der Hand, da unnötige Parallelentwicklungen vermieden werden.

Es werden immer wieder Bedenken geäußert, dass die Verwendung eines FOSS-Betriebssystems mit Copyleft-Lizenz Unternehmen zwinge, Firmengeheimnisse im Bereich von Verfahrenstechnologie und auf Applikationsebene preiszugeben. Dies ist aber sicher nicht der Fall; denn die Applikationsschnittstelle zwischen dem Betriebssystem (dem "Kernel") und der Applikation folgt in der Regel dem sogenannten POSIX-Standard. Diese vom IEEE und der Open Group standardisierte Schnittstelle wird von sehr vielen verschiedenen Betriebssystemen unterstützt, so dass eine auf diese Weise mit einem bestimmten Kernel verbundene Applikation nicht als dessen Änderung oder Erweiterung angesehen werden kann. Die Applikation gilt also nicht als "vom Betriebssystem abgeleitet" und muss daher auch nicht unter der GNU GPL freigegeben werden. Insofern können Maschinenbauer zum Beispiel das unter der GNU GPL lizenzierte Linux als Betriebssystem verwenden, ohne ihre Applikationen oder sonstige Verfahrensdetails offen legen zu müssen.

Gemeinsame Entwicklung von Basistechnologien und offene Standards

Die gemeinsame Entwicklung von Basistechnologien durch viele Marktteilnehmer hat neben der genannten Kosteneinsparung auch noch einen weiteren positiven Effekt: Denn man erreicht - gleichsam nebenbei – auch eine Standardisierung, weil die gemeinsame Entwicklung von Basistechnologien und deren Nutzung zwangsläufig auch zu gemeinsam definierten Schnittstellen und Protokollen führt. Aber dieses Argument gilt auch in der anderen Richtung: Viele der zur Zeit existierenden offenen Standards wurden bereits früher im Rahmen von Open-Source-Projekten entwickelt; in diesem Zusammenhang sind Ethernet, TCP/IP, Netzwerkprotokolle, Seitenbeschreibungssprachen usw. zu nennen. Dadurch stehen hervorragende Implementierungen dieser Standards gerade für FOSS-basierte Systeme zur Verfügung. Diese sind häufig sogar deutlich besser für den rauen Maschineneinsatz geeignet als Nach-Implementierungen im Rahmen von proprietärer Betriebssystem-Entwicklung.

Der Globalisierung mit den Mitteln der Globalisierung begegnen

Die gemeinsame Entwicklung von Basistechnologien wäre ohne die Globalisierung und insbesondere ohne das Internet völlig ausgeschlossen. Die gleiche Globalisierung ist aber auch dafür verantwortlich, dass europäische Unternehmen durch Konkurrenten aus Ländern mit geringeren Arbeitskosten unter einen erhöhten Kostendruck geraten sind. Wenn diesem Kostendruck nun durch die gemeinsame Entwicklung von Basistechnologien begegnet wird, kann es also durchaus sein, dass eine Software-Komponente, die ein deutscher Maschinenbauer kostenlos vom Internet bezogen hat und erfolgreich einsetzt, in einem Land entwickelt wurde, das bisher nur unter dem Aspekt der Software-Piraterie gesehen wurde. Mit der Verwendung von FOSS bekämpft man also in gewisser Weise die Nachteile der Globalisierung mit deren Vorteilen.

"Gemeinsame Entwicklung von Basistechnologien" - Wie funktioniert das praktisch?

Das übliche FOSS-Entwicklungsmodell führt zunächst einmal dazu, dass ein Unternehmen, das als erstes eine bestimmte Software-Funktionalität benötigt, diese entwickelt und den Bedingungen der GNU GPL folgend im Quellcode veröffentlicht. In

der Konsequenz bedeutet dies, dass ein Pionier auf einem bestimmten Gebiet Kosten für anfallende Entwicklungsarbeiten übernimmt, die dann von den weniger fortschrittlichen Mitbewerbern kostenlos übernommen werden können. Ein Unternehmen wird zwar die prinzipielle Forderung nach Offenlegung des Quellcodes akzeptieren können, die daraus resultierende Ungerechtigkeit bei der Finanzierung ist aber nicht annehmbar. Um eine gerechtere Verteilung der Entwicklungskosten auf möglichst viele Nutznießer zu ermöglichen, muss also eine Organisation geschaffen werden, die von vielen interessierten Unternehmen gemeinsam finanziert wird, und mit deren Hilfe mehrheitlich von den Beteiligten gewünschte Software-Projekte in Auftrag gegeben werden. Genau eine derartige Organisation ist das Open Source Automation Development Lab (OSADL), das Ende 2005 gegründet und im Sommer 2006 als eG in das Genossenschaftsregister eingetragen wurde (http://osadl.org/). In den ersten beiden Jahren seiner Geschäftstätigkeit hat das OSADL bereits eine nennenswerte Anzahl von gemeinsam finanzierten Projekten realisiert, wie z.B. die Weiterentwicklung der Echtzeitfähigkeit des Linux-Kernels, Bereitstellung verschiedener Linux-Treiber, Entwicklung von Board Support Packages, Migrationstools usw.

Wozu braucht man Virtualisierung?

Im Rahmen von FOSS-Projekten sind eine Vielzahl von Systemen und Applikationen entwickelt worden wie zum Beispiel der Linux-Kernel, der Apache-Webserver, die OpenOffice-Bürosuite und der Internet-Browser Firefox. Es gibt aber immer noch viele Anwendungsbereiche, für die Softwarelösungen nur auf proprietären Betriebssystemen verfügbar sind, und nicht wenige Unternehmen haben - bevor sie sich für FOSS entschieden haben - nennenswerte Entwicklungsarbeit auf proprietären Plattformen investiert, die sich bei einem Wechsel auf FOSS nicht einfach abschreiben lassen. Es entstand also der Wunsch, auch diese proprietären Systeme unter eine gewisse Kontrolle zu bringen. Diese Kontrolle bezieht sich in erster Linie darauf, dass Software nicht dadurch wertlos werden soll, dass eine von dieser Software benötigte Hardware nicht mehr hergestellt wird.

Virtualisierung bietet nun die Möglichkeit, die komplette Hardware eines Computers mit Hilfe von Software zu simulieren. Auf diese Weise lässt sich eine "nicht abkündbare" Hardware herstellen. Dies setzt allerdings voraus, dass das zugrundeliegende System (das sogenannte "Host-System") wiederum aus FOSS-Komponenten besteht, damit auch dieses nicht abgekündigt werden kann. Seit Dezember 2006 ist die Ker-

nel Virtual Machine (kvm) Bestandteil des Mainline-Linux-Kernels, so dass in der Tat ein nicht abkündbares Host-System mit einer Virtualisierungs-Plattform verfügbar ist.

Neben dem genannten Vorteil, proprietäre Software unter konstanten Hardware-Bedingungen laufen zu lassen und diesen Teil des Systems sozusagen "einfrieren" zu können, weist Virtualisierung noch eine Reihe weiterer Vorteile auf. Unter anderem kann die Verfügbarkeit von Servern durch Virtualisierung deutlich erhöht werden. Dies ist auch der Grund, warum sich Virtualisierung in Servern besonders schnell durchgesetzt hat. Es wird davon ausgegangen, dass zur Zeit etwa ein Drittel aller existierenden und die Hälfte aller neu eingerichteten Server virtualisiert sind. Im Maschinenbau existieren zur Zeit nur wenige derartige Projekte, aber es ist durchaus möglich, dass in einigen Jahren Maschinensteuerungen in der Häufigkeit von 10 bis 20% diese Technik nutzen werden.

Empfehlungen zum Einsatz von FOSS in industriellen Systemen

Wie bereits ausgeführt eignen sich nicht alle Unternehmensbereiche, in denen Software entwickelt oder eingesetzt wird, zur Teilnahme an einem FOSS-Projekt, sondern nur diejenigen Bereiche, in denen allgemein verfügbare Basistechnologien eingesetzt werden. Vor der Entscheidung zur Teilnahme an einem FOSS-Projekt sollte daher analysiert werden, inwieweit das Projekt - wenn auch nur teilweise - Wissens- und Erfahrungsschatz betrifft, der nicht preisgegeben werden kann, ohne die Marktposition des Unternehmens zu gefährden. Hierfür lassen sich zwar branchentypische Richtwerte angeben, aber im Einzelfall können durchaus erhebliche Abweichungen bestehen. Typischerweise gehören Maschinenbauer zu denjenigen Unternehmen, die in der Regel keine Alleinstellungsmerkmale aus den Eigenschaften des Betriebssystems und dessen Programmierschnittstelle beziehen. Man kann sich auf der anderen Seite aber auch vorstellen, dass spezielle Steuerungen für außergewöhnliche Anforderugen es erfordern, Betriebssystemkomponenten zu entwickeln, die dann Alleinstellungsmerkmale für den jeweiligen Maschinenbauer enthalten. In einem solchen Fall käme natürlich die Offenlegung des Quelltextes nicht in Frage, so dass ein FOSS-Betriebssystem mit Copyleft beinhaltender Lizenz nicht eingesetzt werden kann. Auf der anderen Seite ist es aber auch möglich, dass eine zunächst als schützenswert angesehene Software sich bei näherer Analyse doch nicht als als eine solche entpuppt, und die Vorteile einer offenen Weiterentwicklung dann die Freigabe der Software in einem FOSS-Projekt rechtfertigen.

Zur Veranschaulichung der verschiedenen Kompetenzen eines Unternehmens lassen sich diese am besten in Form einer Pyramide darstellen, wobei im unteren Bereich der Pyramide die Basistechnologien ohne Alleinstellungswert und im oberen Bereich die individuellen Kompetenzen angeordnet sind. Irgendwo zwischen dem Fuß und der Spitze der Pyramide liegt dann die jeweils für ein bestimmtes Unternehmen individuell festzulegende "Alleinstellungsgrenze" (Abbildung 1). Befindet sich diese Grenze direkt am Fuße der Pyramide, d.h. besteht kaum oder gar kein allgemeines und mit Mitbewerbern austauschbares Know-how, so kommt FOSS (zumindest mit einer Copyleft-Lizenz) nicht in Frage. Befindet sich die Alleinstellungsgrenze aber weiter oben (gelbe und grüne Linien), so stehen dem Einsatz von FOSS mit Copyleft-Lizenz keine grundsätzlichen Bedenken im Wege.

Ist die Entscheidung gefallen, FOSS einzusetzen oder gar ein eigenes FOSS-Projekt ins Leben zu rufen, sollte man in jedem Fall Rechtsberatung einholen. Diese bezieht sich auf die Art und Weise, wie der Kunde informiert werden muss, dass in einem bestimmten Produkt FOSS eingesetzt wird, wie die Software-Offenlegung am besten zu realisieren ist und welche Rechte und Pflichten mit der jeweils gewählten FOSS-Lizenz verbunden sind. Auch sollte genau geprüft werden, inwieweit auf die Geltendmachung von Patenten - zumindest bei deren Verwendung in diesem und anderen FOSS-Projekten - verzichtet werden muss. In der Regel handelt es sich aber nur um Spielregeln und Handlungsweisen, die bei deren Kenntnis leicht einzuhalten sind. Hilfestellung leisten z.B. Rechtsanwälte, die sich auf FOSS spezialisiert haben, oder auch das private Institut für Rechtsfragen der Freien und Open Source Software (ifrOSS, http://ifross.de/).

Empfehlungen zum Einsatz von Virtualisierung in industriellen Systemen.

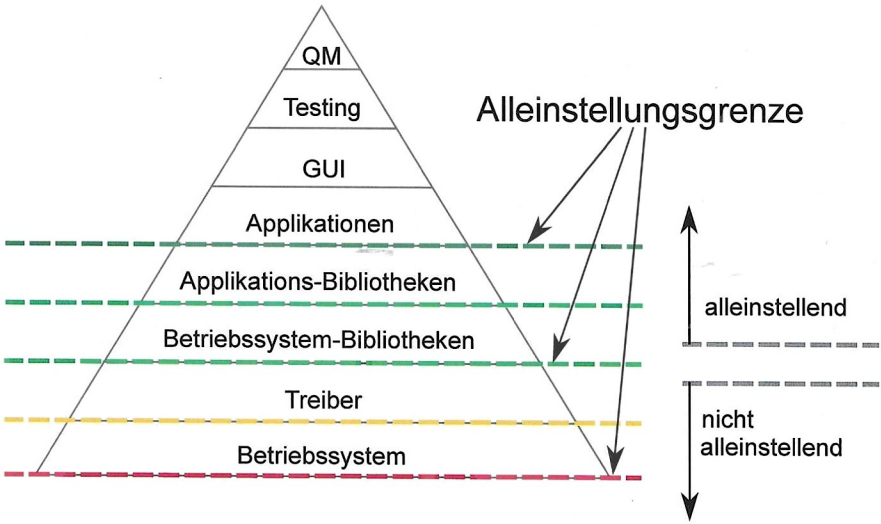
Einer der häufigsten Gründe, FOSS in einer Maschine einzusetzen, besteht darin, eine vollständige Kontrolle über die Maschine zu erlangen und unabhängig von Preiserhöhungen, Abkündigungen, Firmenaufgaben oder sonstigen Liefereinschränkungen die langfristige Funktionsfähigkeit einer Maschine zusichern zu können. Dieses Ziel wird natürlich nur erreicht, wenn sämtliche Software-Komponenten als FOSS vorliegen. Ist dies bei allen Komponenten der Fall und ist die Migration schnell und komplikationslos möglich, so stellt sich die Frage nach Virtualisierung überhaupt nicht. Im Gegensatz zu Servern spielt diese zur Erhöhung der Verfügbarkeit vermutlich im Ma-

schinenbau keine Rolle - am besten ist es also, wenn man Virtualisierung überhaupt nicht benötigt.

Stehen jedoch einige der benötigten Software-Komponenten wie zum Beispiel die graphische Bedienoberfläche nicht oder nicht schnell und einfach genug als FOSS zur Verfügung und ist man gezwungen, ein proprietäres Betriebssystem einzusetzen, so bietet sich durchaus an, dieses in einer virtualisierten Laufzeitumgebung zu betreiben. Diese sorgt dann dafür, dass Abkündigungen von Hardware oder proprietärer Software keinen Einfluss auf die Produktlebensdauer der Maschine haben, da die bei der initialen Produktentwicklung bestehenden Bedingungen jederzeit reproduziert werden können.

Fazit

Der Einsatz von FOSS in der Automatisierungsindustrie ist sicher kein "Allheilmittel", bei gezieltem Einsatz und unter Berücksichtigung der rechtlichen Rahmenbedingungen kann FOSS aber zu messbaren Kostensenkungen und Qualitätsverbesserungen führen. Das gleiche gilt für die seit einigen Jahren verfügbare Hardware-Virtualisierung; auch diese bietet, wenn gezielt eingesetzt, eine neuartige Möglichkeit zum Langzeit-Management und zum Investitionsschutz von maschinennaher Software. Organisationen wie das Open Source Automation Development Lab (OSADL) bieten eine Plattform zur Zusammenarbeit interessierter Unternehmen und gewährleisten eine faire Lastenverteilung.



Abbildung

Abbildung 1: Beispiel einer "Alleinstellungspyramide", in der die verschiedenen Know-how-Bereiche eines Unternehmen aufgeführt sind. Oberhalb der "Alleinstellungsgrenze" liegt dasjenige Know-how, das ein Unternehmen vom Mitbewerber unterscheidbar macht, unterhalb davon liegen Basistechnologien, die am Markt keine Unterscheidbarkeit bewirken. Liegt bei einem Unternehmen diese Grenze sehr weit unten in der Pyramide (rote Linie), d.h. praktisch alles Know-how ist marktrelevant und geheim, so kommt die Verwendung von FOSS eher nicht in Frage. Je höher die Grenze liegt (gelbe und grüne Linien), desto sinnvoller - und unter dem Kostenaspekt desto notwendiger - ist der Einsatz von Freier und Offener Software (FOSS).