Open Source and Free Software: More than Saving Money!

Athina TRAKAS & Arnulf B. CHRISTL

Athina Trakas & Arnulf B. Christl; CCGIS GbR, Siemensstr. 8, 53121 Bonn, Germany, info@ccgis.de

1 INTRODUCTION

Save Money. Currently this catch line is being (mis)used a lot to get legitimate the use of Free Software. This is also what most people first think when they come into contact with the concepts of Open Source and Free Software. Instead we want to show that there are a host of arguments beyond the cost which make Open Source and Free Software a viable and fast growing alternative concept. These arguments in our humble opinion by far outweigh the rather disrespectful reduction of this software to the buzzword „free-of-cost".

The topic of Free Software and Open Source Software (F/OSS[1]) has a long tradition and splits into several categories and branches. If you misunderstand Freeware and Free Software to be the same then the following differentiation should get you going. To understand all the implications of Free Software and Open Source it takes more than this short introduction.

1.1 Open Source

The human readable part of a software is the source code. In traditional software development the source code usually has to be compiled to a binary file which then is readable for machines and can be executed on a computer. The term Open Source describes a software where the source code is readable and freely distributed. Proprietary software developers (and companies) in most cases treat source code like a secret (or even mystery) and the most important and only good they have to compete against competitors.

Nonetheless with fairly simple tools any software can be decompiled (something like reverse compiling) and studied to depth. As in most cases you have to get a copy of the software in order to use it – it is very hard or even impossible to technically really keep it a secret. Therefore proprietary software licenses are very strict about the use and handling of software and contain collections of barriers restricting the use.

The term Open Source explicitly specifies that the sources of a software have to be accessible. The Open Source Initiative (OSI) explicitly extends the openness to other areas, most of which can also be found in the Free Software movement. Open Source is a precondition of Free Software.

1.2 Free Software

The concept of Free Software, initiated by Richard Stallman in 1984, additionally touches rather more philosophical and theoretical questions about copyright, copyleft and users' rights. Initially it was spawned from the simple need for a free UNIX-style operating system, it was named with the acronym GNU (GNU is not UNIX).

Some may think, that "Free" in the context of Free Software is to be understood as in free beer. But you should rather understand "Free" as used in the phrase "free speech" or "freedom". Free Software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. More precisely, it refers to four kinds of freedom, for the users of the software:

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies (freedom 2).
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition (Free Software Foundation / Definition).

For a software to be free in this sense its source code has to be accessible without restrictions.

1.3 Open Source / Free Software

For ease of use we will refer to the term Open Source / Free Software or in short OS/FS when we talk about Free Software and Open Source Software.

For most users the economical aspects are currently the most obvious appealing advantages. Besides supporting very modern and highly accepted software development models, OS/FS can be extended into social and market dimensions.

In the opinion of the oekonux-project[2] the high quality potential of OS/FS is due to the lack of interference of marketing considerations of a company or the influences of a market of commercially available programmers. The focus rather lies on the quality of the software. In the end it is this quality which helps leverage Free Software in the long run – it is definitely not the absence of license fees (Frankfurter Rundschau / Spaß am Programmieren).

The users attitude towards FS in general typically changes the longer it is being used. This is not only true for the viewpoint of customers but also for companies and software producer. In the beginning (and this is what currently is also conveyed to the public)

For the distinction of the words Free Software (FS) and Open Source Software (OSS) please see at the websites of the Free Software Foundation (http://www.fsf.org/philosophy/free-software-for-freedom.html) and the Open Source Initiative http://www.opensource.org/docs/definition.php. In this article both words are used as synonyms or put together in the acronym F/OSS.

In this project economical and political characteristics of Free Software are studied and discussed using different methods (http://www.oekonux.org/ from 2003-12-17).

monetary arguments clearly dominated the decision processes favoring OS/FS. Only when users start using those concepts seriously they start to appreciate the aspects of high quality, security and constant further development.

1.4 Adoption of OS/FS concepts by software user

The adoption of Free / Open Source Software has been initiated by software specialists who had the need to access source codes in order to adopt it for their special purposes. A colorful community communicating via Internet developed. This community did not rely on centralized institutions but lived the concept of multiplying knowledge by sharing it.

Surprisingly FS and OSS have also become a dominant issue in government administrations recently – even although it can be seen as the incarnation of decentralized and deregulated development. Germany for example is deploying Linux and the Open Source model at the federal, state and community levels. Secretary Siegmar Mosdorf argues that OS improves software security (Heise / Wirtschaftsminister). The german Minister of the Interior Otto Schilly explains that Germany is lowering dependence on a single supplier, while creating more diversity in the computer field..

In December 2003 the European Commission (Directorate General Information Society) launched a new sections on their websites providing information on activities conducted within European programs and initiatives related to Free Software and Open Source . The Open Source Observatory (OSO), launched by the Enterprise DG and funded under the IDA Program, aims ultimately to provide a comprehensive overview of Open Source Software activities in current and future EU Member States[3].

1.5 The LINUX grassroot movement

High tribute has to be payed to the Linux movement for propagating OS/FS concepts. The initial slow development and adoption of Free Software has been accelerated considerably because of the ubiquitous presence of Linux and their developers all over the world. It has also brought about some changes which unltimately led to a separation into the movements „Free Software“ and „Open Source“ (see above) which together foster the further development of F/OSS in their own ways (Raymond / Cathedral).

1.6 Not everything boasting the term „open“ is OS/FS

As happens with all successful terms now the prefix „open“ can be found indistinctly combined in whichever combination the marketing whim currently toys around with. So a whole lot of new „openness“ plasters glossy advertisements. Therefore it is important to clearly distinguish between Open Source, Free Software and whichever else good things might be meant.

2 SOFT(WARE) SKILLS – ARGUMENTS BEYOND SAVING MONEY

In the following section we want to light up some of the more important arguments for Free Software and Open Source.

2.1 Collaboration as a basis of all OS/FS concepts improves software development

The opportunity to access the source code not only opens the possibility of greater creativity but also helps to reduce cost and time of development. The synergetic effects of both aspects can be accounted for immediately. These ideas offend against the traditional calculation of software development where all factors are predisposed.

This concept also only sustains if there are enough programmers and interest in the OSS project to solve all technical problems. Without the participation of enough competent programmers a project may not be further developed and will fail, and many do. That is also part of the OS/FS concept – a somewhat more evolutionary approach, only the strong survive, but not in the sense of strong market presence or financial power but simply in the sense of stronger, better software.

Therefore an enormous potential for FS concepts lies in public administrations due to the possibilities of combining efforts to deal with questions and problems in a large knowledge community. This is especially true to GIS environments, because their aim is to make geographic data accessible.

2.2 Lack of complicated, intransparent and constricting licensing models

Licenses which follow OS/FS models (GPL-, Artistic-, BSD-like etc.), have the user/costumer pay only for the work which is really done. Proprietary licensing “protects” (in a short-sighted economic sense) the development from the real world and encapsulates the programmer as a segregate group within the software producers management which is driven by economical directives. Some licensing models from proprietary software producers are difficult to understand, complicated, subject to frequent unannounced changes, what can result in an incomprehensible and unreproducible mess.

Because of the four freedoms which are inherent in FS the license management for the users is much easier. Users can install any number of copies - there is no risk of illegal copies or license audits, and there are no anti-piracy measures (e.g. CD keys, product activation). Current users praise this freedom from licensing burden, but OS/FS doesn't eliminate software management, customers still have to track versions. The quality of any program, especially for security, depends on patches and other updates. Again the responsibility is shifting – away from the software producer and into the domain of the user where it belongs (NETC / management).

2.3 OS/FS is more empowering

Open Source empowers users instead of restricting their rights as proprietary software does. Any user can fix bugs or add the features that matter most to them (or hire someone, or collaborate with others). Software companies may decide a bug or feature isn't critically important to most of their users (or their sales) and ignore the users request. Open Source at least offers the users the

The pages can be found at the following URLs:
Informations Society: http://europa.eu.int/information_society/activities/opensource/text_en.htm

opportunity to decide for themselves and either solve the problem on their own, hire a specialist or align with others who face the same problem.

This argument – becoming (more) independent from the pricing and licensing policies of software companies – is one of the mayor aspects pointed out in the FLOSS study[4]. This is also a major point emphasised by the proponents of OSS in the public administration as the recent discussion in Germany has shown. With budgets tight, previous software versions becoming unsupported and new licensing schemes coming up, which are considered to lead to higher software expenditures, establishments wish to become less dependent from pricing and licensing policies of big software companies (FLOSS Study / p. 20).

Even if the software producer goes out of business (for whatever reason), the community still disposes of the source code and may decide to further develop it. This independence also means "end of life" decisions or undesirable new features can't be forced on the users. Customers aren't caged into a single vendors development model or support department. The original software producer or company may offer the best support, but since the software is open anyone can try to improve or support it. Everybody can choose the best solution now with the freedom to change in the future. Nobody has to rely on a single software company for all aspects of the solution.

2.4 Attitudes and relationships between the "user" and "producer" changes

Due to the disclosure of the source code, the user earns the possibility to get an insight view of the software. It helps to emancipate the user from the dictate of the software producer – not speaking in monetary terms but in a technical sense. The possibility to get to know the software better also improves the chances to influence further development of the software. Additionally the user knows that he receives imperfect software, it is no secret that software never is perfect. This is one of the greater problems for traditional proprietary producers, because in advertising campaigns they have produced the expectance that their software is the ultimately and only true best solution. One of the great advantages of FS is that all users automatically also comprise the test team. No single software producer will ever be capable of recruiting as many beta-tester as users, if following the traditional concept of treating software testing as a separate, closed task within the closed development process.

In proprietary concepts there is a fairly strong antagonism between the user claiming a clearly defined value for his money and the producer who never seems to be able to deliver exactly what is needed. This happens for several reasons:

1. One problem roots in the marketing concepts common within proprietary software development which are in most cases driven by financial considerations and market analysis, but are not user centered. Most producers explicitly proclame thinking and acting in a user centered way – but who really believes them if they are a multinational behemoth?
1. Software development is always based on existing software, which itself also underlies changes. Best example are operating systems, which form the most basic platform for any software. In traditional proprietary development a new operating system is kept secret as long as possible to have a head start on the competitors. The result is that the operating system will not work correctly because it has never before seen the diversity of problems in the real world.
2. New releases invariably contain errors, which provokes the saying: „The banana [software] ripes at the customer's". If that is the case anyway – why should we ignore this fact and deceit users into believing that they receive perfect software?

The result of these contradictions is that users become more and more demanding because they know that they will receive less than promised anyway. Being part of the development **and** user community reduces this antagonism to an open challenge where all protagonists have the same rights. Users can directly influence the software development, either by programming themselves, contributing to discussion and user lists or directly investing money into a special features.

Large software enterprises like Microsoft have silently over the past years adopted many of these concepts (frequent intermediate releases (bug fixes), public user lists, open forums, etc.) but not changed the politics of closed source. This is now turning out to be a larger problem than suspected.

3 TECHNICAL ARGUMENTS FOR OS/FS CONCEPTS

In the following section some background on the technical advantages Free Software projects can provide are pointed out.

3.1 Life cycles and development cycles

Open Source software tends to have very short development cycles before new releases are published. Beware! This does not mean that the life cycle is short. OS/FS fosters shorter development cycles and a higher transparency of the development, having a much bigger and immediate impact on software development concepts than economic factors. The concept to "publish early and release often" is not defined as a flaw but a conscious decision to speed up development and improve quality.

As Torvald says: „If 10 people work one hour a day for a project and share their results, than everybody gets 9 hours out. That is what Linux makes that good: the work which is put in by a person is multiplied." And: "...there is a kind of automatically quality control, because everything is all the time observed by other programmers." (die Zeit / Open Source).

The life cycle of a typical OS/FS project end when nobody supports it anymore and that will happen either if there is a better alternative or the problem no longer exists. Never will economic reckoning lead to the death of a software (end of the life cycle) as regularly happens in the proprietary world.

IDA / OSO: http://europa.eu.int/ISPO/ida/jsps/index.jsp?fuseAction=showDocument&parent=news&documentID=1809
FLOSS Study: http://floss.infonomics.nl/report/index.htm or FLOSS-US http://www.stanford.edu/group/floss-us/report/FLOSS-US-Report.pdf

3.2 Reliability

OS/FS may be more reliable than proprietary. We explicitly use the term “may” because for sure there are hundreds of vastly unreliable software fragments bobbing in the Open Source surge currently flooding the software markets. It may not make as many errors or crash as often (e.g. Linux is famous for not crashing). Due to the access to the source code programmers can find and fix bugs, software may be repaired and improved more quickly. Not every initial program may be more reliable than a proprietary alternative, but it may mature faster as a lot of programmers correct mistakes and add features. Eric Raymond calls this Linus' Law: "Given enough eyeballs, all bugs are shallow." (Raymond / Cathedral). Some may think of this as permanent beta testing which is not wrong at all, the Open Source community can endlessly troubleshoot and improve software as needed or desired.

As mentioned in the beginning, the advantage of improving a software depends on the participation of enough competent programmers. Just like proprietary software, the reliability of an Open Source program depends on clear feedback after rigorous use in a variety of environments. Without enduring, sufficient, talented interest, an OS/FS project fails, and many do. In contrast, proprietary software companies may create and support necessary programs that no one would enjoy working on. Some companies are starting to blend the best of both models, by employing a core group of programmers while attracting volunteers from the open source community (NETC / reliable).

3.3 Better security and quality of Free Software due to the free access to the source code.

There are different studies stating that OS/FS products often reach a higher software quality compared to proprietary (closed source) products.[5] Mainly the high transparency causes the high quality of OS/FS projects. Linus's Law describes parallel debugging: multiple programmers independently finding and fixing the same program to discover the best solution. Security threats like viruses and worms exploit software bugs to damage computers. One of the advantage of parallel debugging is the pace of finding and fixing bugs much faster than the traditional way.

For both Free and proprietary software, effective security depends on thoughtful deployment, regular monitoring, and timely upgrades or other modifications. Some proprietary solutions offer potentially robust security, but Linux leads the industry in defensive design. This is one critical reason why business and government are interested in OS/FS. For example, Linux and most FS isolates users from the code viruses and worms need. These threats need unchecked access to execute, replicate, and deliver a payload (e.g. delete all files). In a properly configured Linux environment, an email attachment in a user's inbox can't spread destruction to the whole system (NETC / secure).

The most obvious argument proposing the security of software can be improved due to the availability of the sources is that OS projects constantly underlie the scrutiny of a vast number of users.

3.4 Transparency – everybody can see everything

Sometimes transparency is misinterpreted as a critical security threat. However, the OS model brings a community of programmers to maintain and improve security. This collective benefit seems to outweigh the danger of transparency. Also, the security of a system depends much more on careful deployment and maintenance. For example, the source code for Linux is publicly available. But any good system administrator will deploy a Linux server only with secure passwords, firewalls, additional software, and other defences. Thus, transparency in the "virgin" source code is usually a trivial concern.

If software is transparent, any programmer can see what happens and why. Proprietary software is not transparent and most computer users are familiar with the strange error messages that appear when a program fails. Often, these messages use special codes to express the problem. These codes point to secrets without actually revealing what went wrong. Only someone who knows the secrets can understand the codes. It's the suspicion of such errors that causes people to distrust proprietary programs. Some countries are suspicious of proprietary software because it could contain spyware or other security threats.

OS/FS programs in contrast aren't trying to protect themselves as secrets. Therefore they can offer more exact information about an error. With this information any programmer can start to identify and fix problems. This does not have negative effects on the security of Open Source and Free Software.

3.5 Better networking and customisation with OS/FS

A lot of the popularity of OS/FS comes from its performance on the Internet and lesser networks. Open Source software is often very networkable. For example, more than half the World Wide Web runs on Apache, an OS solution. Apple built its OS X on BSD, an Open Source operating system. The critical reason is the Internet: Some proprietary companies recognize that they can't privately innovate Internet functionality as well or as fast as the Free Software community. Furthermore most of the FS networking solutions are compatible with proprietary software, due to norms and standards. One of the reasons, that the OS/FS movement is getting so strong can partly be explained as a response to incompatibility in proprietary software. Due to a earnest commitment to open formats OS/FS programs are generally better at working together.

By its very nature, OS software allows users with enough expertise to tailor the software to their needs. The diversity of free based server-system (Linux) distributions reflects this flexibility. Each distribution offers a customized operating system targeting a specific market. On the back end, the open, modular nature of open source solutions allows advanced users extraordinary power to customize any aspect of a network (e.g. firewalls, spam filtering, email filtering) (NETC / customized).

3.6 Open formats and Standards

Open formats make integration of different systems and data much easier. The term “open” in this context does not refer to the OS/FS software development models and should therefore not be mixed up with them (see above). Nonetheless they very much improve the acceptance of any software package, especially in the context of GIS where the OGC (Open GIS Consortium) specifications and ISO standards help to integrate software from both concepts and offer a nearly seamless blending of existing data architectures. Especially in the field of geographical questions it is important to integrate different data into a system. (The data is by far much more valuable than the software package, no matter how expensive it might have been.) For example, suppose different cities in the same district have different database programs, and each program has a proprietary format. To integrate the data of all cities, the district would have to select a program that licenses all those formats, or convert to the data to the new format, or abandon some data. Using open formats forecloses on these problems and may curb opportunity costs.

Integrating any two programs is often difficult. It may also be tricky to integrate Open Source and proprietary solutions. This can also happen because proprietary companies prefer their customers to exclusively use their software for most or all of their needs. In contrast, the TMTOWTDI ("There's More Than One Way To Do It") philosophy[6] of most OS/FS projects promotes integration. In the beginning FS was a minority solution, so it's designed to integrate as smoothly as possible with proprietary solutions. For example, a Linux lab can be nested in a Microsoft network using Samba.

Most newer OS/FS projects in the GIS field implement open formats and standards because the chance that the software is accepted by users is a lot higher. This is especially true for German administrations where the interoperability of systems and data are always a primary goal for larger infrastructures.

3.7 Necessary expertise – Application of OS/FS is good investment

For any software solution whether Free/Open or proprietary expertise is needed to deploy, secure and maintain. In the past, most OS/FS was technically tricky and required considerable know-how. Today, OSS can be just as easy be set up as proprietary alternatives. Solutions like OpenOffice.org are designed for entry- or mid-level user expertise, offering installers with graphical interfaces and suggested configurations.

As everybody knows, using software always includes a learning curve. People want computers to work like appliances. But neither the proprietary nor the OS/FS model have produced many solutions as reliable and user friendly as toasters. For better or worse, most users are more familiar and more comfortable with proprietary software, especially Microsoft Windows. However, many OS solutions are somehow cloning proprietary interfaces and environments. For example, anyone familiar with Microsoft Office will probably find it easy to use OpenOffice.org for simple productivity.

But which are the barriers to using OS/FS? Mindshare and comfort are perhaps most influential among users working with front end solutions. In contrast, most of the back end users (e.g. server technicians) are already familiar with Open Source solutions or may adjust more readily. But even proprietary software may change from version to version, causing mindshare dissonance and user discomfort. Customers using front end solutions may have very legitimate reasons to resist migration to OS/FS. They may have projects working and dependent on specific proprietary software. They may not have the time or energy to learn new software. Notably, some advanced users prefer the freedom of Free Software since they feel proprietary software caters to the lowest common denominator. As users grow more comfortable with Free and Open Source solutions they may be pleasantly surprised, especially if they're only ever used a single proprietary solution. OS/FS may not offer certain or more features as proprietary software, but it's not necessarily more difficult to deploy. Therefore Open Source and Free Software can also be called a disruptive technology.[7]

3.8 Service, Support and other costs

Any software solution requires some service and support. For both Free and proprietary software, experts and costumers can find help on email lists and community Web sites as well as in contracted support. As mentioned before the difference is that in OS/FS paradigm by default every user can be a tester whereas in proprietary development a software has to be tested **before** it is shipped to the end user – resulting in unstable releases.

The quality and availability of help is proportional to the interest and use, especially in OSS. If there is only some small group of developers, usually the costs for niche solutions are high – just the same as with proprietary products. On the back end, OS/FS is common so the community of developers is large and helpful.

As Open Source software usually is more modular than proprietary alternatives, customers can decides which of the modules they need for their questions. Modular software means each program is a discrete piece of a solution. If the user finds a better program, it can be swapped in without replacing the whole solution. OS/FS often offers several interchangeable, modular programs for various needs. This approach is summarized as "There's More Than One Way To Do It" (TMTOWTDI). Users can find the best components and then decide whether further development has to be financed. So Free Software solutions are usually scalable at little or no cost, while most proprietary solutions are incrementally more expensive with each new user or machine.

4 CONCLUSION

4.1 Open Source is community-driven & community-serving

As mentioned above the OS/FS movement exists because a large community of motivated, generous programmers work together. Not every programmer is a volunteer, some are corporate employees, but the movement succeeds through the participation of volunteers. Current users are often keen to help others solving their problems and give back to the community. Especially users without programming or other technical skills help by filing endless bug reports, writing documentation, or answering questions on

Further information: German Federal Office for Security in Information Technologies http://www.bsi.de or bundestux – http://www.bundestux.de

See: http://www.perldoc.com/perl5.8.0/pod/perlfaq1.html "Is Perl difficult to learn?"

After Christiansen disruptive technologies are simple and convenient-to-use innovations, which at the beginning have a marginal existence but then are continuously developed and start to capture the market (http://www.disruptivetechnologies.com/).

email lists thus exempting the real programmers to do their job. Current users report a sense of belonging and accomplishment by sharing and collaborating. This cooperation and focus on the common good accumulates in a new relationship between customers and consultants.

4.2 Open Source does not threaten intellectual property rights.

Opponents often argue that OS/FS is a menace to intellectual property rights. The CEO of Microsoft said, "Linux is a cancer that attaches itself in an intellectual property sense to everything it touches." (Linux / interview Ballmer). This argument refers to the most radical OS/FS license – the GNU GPL, and only points out the fact that source code released under the GPL can't be included in closed, proprietary software. The GNU GPL has been designed to protect its software from being looted by commercial interests – just as any software license in the world ever has. From the proprietary point of view this is the most normal thing that a software license would do.

Another argument against OS/FS is that governments should not invest into Open Source development because then this software cannot be used by commercial enterprises. This is simply not true as is impressively demonstrated by the constant growth of SuSE but also by giant companies like IBM, HP, SUN and many more smaller enterprises.

On another scale it is a legitimate question to ask where all the programmers who work for Microsoft have gone to school, high school and university? There is no return on investment for all the money which has been invested by parents and the government to build schools, universities and libraries – except to the benefit of the entrepreneur. Technology developed with OS/FS concepts potentially offers many advantages and challenges, but it's exagerated to compare it with the theft of intellectual property. New business models and social norms are developing, just as they have in the past (NETC / property).

4.3 OS/FS is a paradigm shift in information society – we are overnewsed but underinformed

Some old and new companies are quite successful using OS/FS. The movement represents a paradigm shift for the software technology, but it's not anti-business. Free Software may challenge some existing businesses, but doesn't competition inspirit the market? There's little reason to buy a proprietary program when a comparable Open Source alternative exists. The software industry is already converting from the product model to the service model for quite some time now, in Germany we comprehend ourselves as transforming into a “Dienstleistungsgesellschaft” - a service community. Companies like IBM are thriving with OS/FS because they offer quality service, other companies like SuSE sell OSS with added value, including service (Linux distributions).

Furthermore as shown in the previous chapters FS can protect users against paternalism and constraints of their civil freedoms. Important in this issue is the basic understanding about information: Is it permitted to deprive the public arbitrarily of some information or does the public have the right on special information? Some distinct information "wants" to be free of structure and arrangement and free of usage restrictions.

The idea of free information is a cultural perception known for hundreds of years. OS/FS is not a communist or anarchical movement. The concept behind the disclosure is one of the basic modules of our culture since the renaissance. Like libraries ensure and protect public available knowledge, our academic culture is based on the principle of the disclosure of sources (e.g. references, citations or documentations of experiments) (Buchholz / Argumente).

4.4 Standardization as conciliation between proprietary and open development models

Especially in the context of GIS software the use of standard interfaces, for example the OGC WMS specification and ISO norms also helps integrating Free and Open Source Software with other normed software which can also be proprietary. This opens the possibility to select the adequate software to solve a problem and not having to follow the largely commercial (and thus intransparent) interests of one software producer. Therefore standardization (hopefully) helps to reduce the currently growing antagonism between proprietary and Open Source development and business models.

In the GIS arena, Open Source and Free Software solutions are maturing and finding their place in a crowded commercial marketplace. There is no explicit need for proprietary software but there is also no reason not to adopt and integrate both concepts.

5 LITERATURE - EXTRACT

Buchholz, A.S. (2003, February 16): Argumente für Freie Software. Retrieved December 11, 2003 from http://www.kefk.net/Linux/Meinung/Freie-Software/Pro/index.asp

Frankfurter Rundschau (2003, December 06): Spaß am Programmieren. Retrieved December 15, 2003, from http://www.frankfurter-rundschau.de/uebersicht/alle_dossiers/politik_inland/die_alternative/die_alternative_themenausgabe/?cnt=352259&

Free Software Foundation: The Free Software Definition. Retrieved December 11, 2003 from http://www.fsf.org/philosophy/free-sw.html

Heise (2000, June 29): Wirtschaftsminister: Sicherheit und Innovation durch Open Source. Retrieved December 16, 2003 from http://www.heise.de/newsticker/data/odi-29.06.00-000/

Linux Online (2001, June 01): Experts from a Chicago Sun-Times interview with Steve Ballmer. Retrieved December 16, 2003 from http://www.linux.org/news/2001/06/01/0003.html

Lütge, G. (2003, October 23): Programmierer aller Länder vereinigt Euch! Die Zeit, retrieved November 24, 2003 from http://www.zeit.de/2003/44/Open_Source

Northwest Regional Educational Laboratory: Open Options. Retrieved November 24, 2003 from
http://www.netc.org/openoptions/pros_cons/features.html#customized
http://www.netc.org/openoptions/pros_cons/deployment.html#management
http://www.netc.org/openoptions/pros_cons/principles.html#property
http://www.netc.org/openoptions/pros_cons/features.html#reliable
http://www.netc.org/openoptions/pros_cons/features.html#secure

Raymond, E.S. (1998, February 04): The cathedral and the bazaar. Retrieved December 15, 2003, from http://fringe.davesource.com/Fringe/Computers/Philosophy/Cathedral_Bazaar/