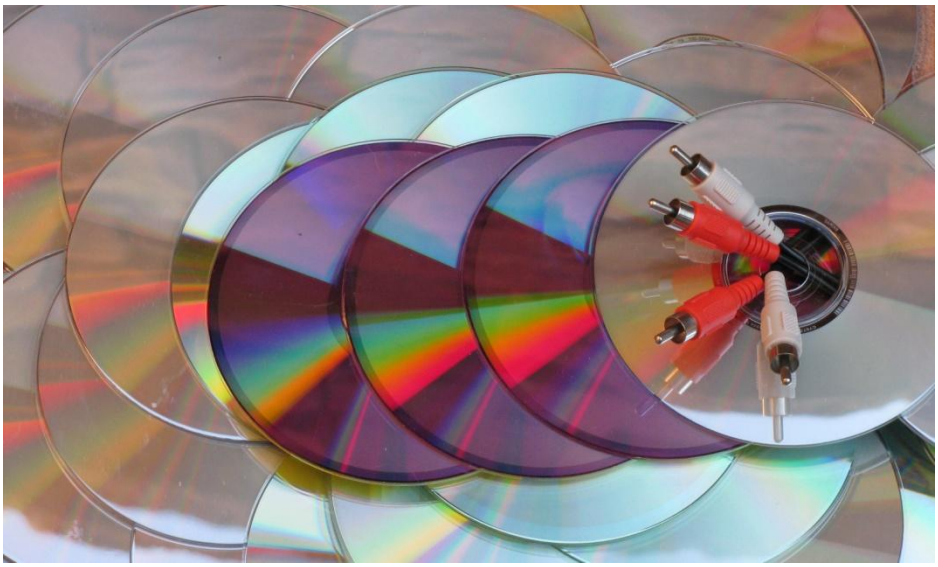
2010 ISSUE
B>Quest
BUSINESS QUEST
1996 - 2010



# *Free Software: Uses of Free Software and Its implications in the Software Industry*

## *by Vivek Shah and James Keefe*

Peer Reviewed

*Vivek Shah is a Professor of Quantitative Methods and James Keefe jk07@txstate.edu is a Senior Lecturer in the Department of Computer Information Systems at Texas State University at San Marcos.*

2010 ISSUE
B>Quest
BUSINESS QUEST
1996 - 2010

## Abstract

Throughout the world, millions of ordinary consumers as well as businesses now use “freeware” - open source software applications that are available either for free or at a small price. This paper presents pros and cons of using free software either for personal or business use. The economic impact of free software on proprietary software developers is explored, as well as the business case for developing or servicing freeware for public use.

## Introduction

The information economy is a vast market and includes the provision of infrastructure and services. This definition includes software, databases, music, video, book content, designs, genetic information, human and organic memories, and other entities that may eventually be represented, stored, and communicated as bits. Software is the key element driving the information economy, and this economy is going to be effected by developments in the software industry, particularly the development of free software.

“Freeware” are open source software (OSS) applications that are either available for free or a comparatively low price. Freeware is different from free software in that the source code that is usually proprietary or not publicly available. Freeware is also distributed for free (or at a very low price) and is usually a light version of commercial software. Many “Free” software applications, on the other hand, have open source codes; as a result, anyone can make modifications (including updates) to the code. “Free,” in this sense, is not only about price but also about the freedom to access and modify the code as well as the freedom to distribute the software.

To address the ambiguity between “free as in free speech” versus “free as in free beer,” a number of alternative terms have been suggested. The expression “free and open source software” (FOSS) includes both “free” aspects of free software. Software that is distributed at no cost is usually referred to as “gratis,” whereas OSS is usually referred to as “libre”, which means that anyone has the freedom to read, modify, and redistribute the source code (http:// GNU.org/philosophy/free-sw.html).

This paper will provide an overview of free software as well as a discussion of the pros and cons of using free software for either consumer or business use. Factors considered include software performance, security, reliability as well as associated costs. Further, the impact of free software on proprietary software developers as well as the business case for developing or servicing FOSS for public use is discussed.

## History of Free Software

There has been a recent surge of interest in “open source” software development, which helps developers at various different locations and organizations share code to develop and refine programs.

In 1980, the US patent office included software as patentable art. Prior to 1980, software development was a collaborative effort between colleagues in the computer industry. With the new patent law in place, companies began developing proprietary software that they could license to users for a fee. This proprietary software gave companies a competitive advantage, but did not allow for public access to the source code, thus eliminating the possibility for collaboration. In 1983, Richard Stallman, a programmer at MIT, was not happy with the way software development was headed. By 1984, Stallman began a mission to ensure that FOSS is available to anyone who desired it.

“GNU” (a recursive algorithm for “GNU is not UNIX”) is a free and open source operating system, first developed by Stallman in 1985. Since UNIX was a popular proprietary operating system, the idea was to create a UNIX-compatible operating system that would be made available to the public.

Stallman also founded the nonprofit Free Software Foundation in 1985, which exists to promote the development of the GNU project and to protect the legal status of free software (www.fsf.org). Although the foundation, in principle, is opposed to software patents and copyrights, the foundation holds the copyrights for the GNU operating system in an attempt to prevent it from becoming proprietary.

Another example of FOSS is Linux, a free operating system. Developed in the early 1990s by Linus Torvalds, the Linux “kernel” (key software code component) is used as a part of the GNU operating system (http://www.gnu.org/gnu/gnu-history.html). Now, there are other software applications available that are compatible not only with

UNIX-based but also Microsoft-based systems such as the Firefox web browser and openoffice.org, which competes with the Microsoft Office package.

## Economics of Free Software

A number of open source products, such as Linux and the Apache web server, dominate their product categories. Linux running on corporate servers is one of the most common uses of open-source software among businesses. According to a September 2008 Gartner survey of 274 companies worldwide that either currently use open source or are planning to do so in the next 12 months. About 52% of the companies surveyed are already using open-source server software and another 23% plan to use it within the next 12 months (King, 2008).

Over the last decade, numerous major corporations, including Hewlett Packard, IBM, and Sun, have launched projects to develop and use OSS. Meanwhile, a number of companies specializing in commercializing Linux, such as Red Hat and VA Linux, have completed initial public offerings, and other open source-based companies such as Cobalt Networks and Sendmail have received venture capital financing. Also, commercial software developer like Oracle's bid acquire open-source database maker mySQL was a clear sign of the profound changes commercial software giants are willing to make as they adapt to the increasingly significant collaborative programming philosophy (Shankland, 2006).

Free Software applications are first, second, and third-running products in terms of market share in several markets, including web servers, server and desktop operating systems, web browsers, e-mail, and other infrastructure applications.

Some of these market developments are easy to account for. For example, software vendors support open source platforms because failure to do so would significantly limit their market size. Another factor driving change in the software industry's business model is that software applications fit into specific niches within larger systems and cannot be readily substituted for one another. Software is different from a sack of concrete or a reel of wire cable. Some vendors behave as if software prices are set by the market, but you cannot buy a quantity of software from one vendor and an equal amount from another vendor and expect them to be interchangeable—something you could expect to do with cable or concrete.

## Free Software in the Business Market

The impact of open source technology is expected to continue to be quite significant in the software industry and in society as a whole. It allows for novel development models, which have already been demonstrated to be especially well-suited to efficiently take advantage of the work of developers spread across all over the world. It also enables completely new business models, which are shaping a network of groups and companies based on OSS development. It has, in general, a very positive impact as an enabler for the creation of new markets and business opportunities (Working Group on Libre Software, 2000).

Apache is the premier example of a FOSS business enterprise network. “The Apache system emerged in 1995 and was derived from a set of patches that were applied to the then-popular NCSA (National Center for Supercomputing Applications) web server source code (leading to the name ‘Apache’ server, a play on the words ‘a patchy’ server)” (Boulanger 2008). In the beginning, the patches were contributed by volunteer users who were frustrated by the original software. By the end of 1995, the Apache software was entirely rewritten by the volunteers.

## Benefits of Using Free Software

We have identified six aspects of free software that contribute to competitive advantage.

- **Lower cost of ownership**: The lower cost ownership is one of the main reasons for enterprises to adopt Free Software. The companies spend heavily on commercial software products for licenses. OSS offers low license and maintenance costs, which helps to reduce the technology cost for a company. Forrest Research found that 87 percent of their respondents have gained expected cost savings from using OSS (Forrester Consulting, 2007). E*Trade is one of the companies that have used open source technology to reduce their costs and they were able to save $13 million a year through the use of open source applications (King, 2008).

Its success is no longer limited to basic software, such as Linux or Apache, a program that powers web servers. Free Software also helps companies to save on areas such as collaboration, customer relationship management, and supply chain management. Open-source firms are flourishing in databases (Ingres, for instance), business intelligence (JasperSoft), customer-relationship management, and other business applications (SugarCRM, Alfresco). In addition, open-source firms have

started to move into new markets without proprietary rivals. For instance, a company called Cloudera distributes a version of Hadoop, a program that helps firms process and analyze the unprecedented volumes of data generated by large websites. The small businesses are using Asterisk, which is an open source telephony engine and platform, to reduce their communication cost (Rupley, 2009).

When companies are operating under a tight budget, more companies are looking for open source technologies. Usually, open source is typically a tenth of the cost of traditional enterprise software (Howells, 2009).

- **Better performance**: All too often, commercial software strives to create unnecessary complexity in software products in order to generate revenue. OSS developers, on the other hand, usually strive to deliver only core features. Hence, companies are able to achieve better performance when using OSS. Many of the OSS programs such as Linux, MySql, Apache, and Eclipse are responsible for the increased efficiency in product and service development (Ebert, 2008). Office Depot is enjoying the benefits of using open source technology. They replaced their proprietary software with Novell's Linux operating system. They found it to be the most cost-effective mechanism to standardize their system to one platform. The cost-effective, standardized platform has given more flexibility because it runs on off-the-shelf servers and the company can build the system when needed, rather than upgrading an entire mainframe all at once (Ebert, 2008).

- **Better quality**: More people are involved in reviewing OSS source code than proprietary software. Issues such as security breaches are fixed and communicated relatively fast to the user community. Therefore, OSS has better quality than commercial software. Forrest Research showed that 92 percent of respondents found the quality of OSS to exceed their quality expectation levels (Forrester Consulting, 2007). E*Trade's Thompson has found that their systems are more reliable under OSS. On January 22, 2009, when the interest rate was changed by the government, 55,000 customers logged into their website at once. Not only did the site perform well under the heavy load, it performed better than its competitors (Ebert, 2008).

- **Decreased management workload**: The software source code is publicly available and managers have the opportunity to integrate the software to support company needs. Managers do not need to invest resources in writing code. They can instead allocate the saved time on developing other useful features for business needs. In addition, the free software can be reused for other similar tasks or projects. Using OSS

reduces work time and provides predictably reliable results. E*Trade's Thompson found that their engineers spend less time on contract negotiation and more time on the technology when they use OSS.

- **Good support systems**: Since OSS development involves a community of programmers and computer experts, the situation allows the OSS user to experience rapid implementation of new features and security fixes. Moreover, the community provides answers relating to troubleshooting and suggests enhancements.

- **Unlimited upgrades**: Commercial software sometimes does not share its code, and therefore, upgrades are not possible. Some commercial software companies might provide limited upgrades for a specific time. On the other hand, upgrades can be extended indefinitely.

## Concerns Regarding the Use of Open Source Technology in Business

The popularity of OSS is increasing in software development during this current financial downturn. It is tempting to switch to OSS for cost savings since the source code of OSS is available for free. However, this may necessitate ongoing maintenance or lead to other undisclosed hidden hassles for the company. Some of the OSS-related concerns that have been addressed are as follows:

- **Poor Code**: From the business perspective, OSS may involve issues and risks for Program Managers. Considering that the size of the OSS community is not sufficiently large, the people who are involved in it may be unreliable. Thus, this situation often leads to poor code development and fails to attract skilled workers to work with the systems.

- **Lack of vendors**: About sixty-five percent of commercial software users stated their preference of commercial advantages over open source because of the lack of vendor professional services (Asay, 2007). This option is considered one of the main reasons why people refuse to use OSS. As for commercial software, the user does not have to take any responsibility for system integration, deployment and support, since the vendor can take care of it all. Vendors provide sales support and engineering staff trained to make this happen.

- **Complex legal restrictions**: According to Koohgoli, the CEO of Protecode, the issue is not about the use of open source, but with the unmanaged copyright and licensing issue (Pearlman, 2009). Open source has complex legal restrictions that can create copyright and patent compliance issues and corporate transaction challenges for companies that rely heavily on customized software or that distribute software to partners or customers. There are two categories of licenses. The first is an attribution-type license. This acknowledges that the authorship of the software is included in some manner, such as source code comments and help files. The second is the reciprocal-type license, known as "copyleft", which requires users who make modifications or extractions of copyleft software keep the derivative products free as well (Free Software Foundation). The idea is to ensure compliance with the applicable licensing requirements.

- **Availability of new features**: Since the open source community is based on volunteerism, new features for software cannot be expected. This could absolutely have an effect on business performance, especially in a dynamic competitive environment where businesses fight over market share. Companies have to adopt with the current market; OSS might support them in the short term, but probably not in the long term.

- **Difficult initial adoption**: Commercial software might be costly, but the systems are easier to adopt. Meanwhile, OSS is associated with an indirect cost. This means more salary and other labor costs could be wasted due to uncommon knowledge required to use OSS. Organizations that do not have experience with OSS projects will find it easier to get commercial software running. Moreover, if the previous project or assignment was created using commercial software, moving to OSS could cause inefficiencies within the whole business process (Rangaswami, 2008).

## Business Implications

**Software industry implications**: FOSS has had financial impacts on commercial (proprietary software providers). According to Boulanger (2008), "In several recent 10-Q quarterly filings with the Securities and Exchange Commission, Microsoft, one of the world's largest software publishers, has stated that the popularization and adoption of FOSS systems pose a significant challenge to its business model." The Standish group has quantified the impact to proprietary software companies. A Standish Group article (2008) states that "FOSS is the ultimate in disruptive technology, and while it is only 6% of estimated trillion dollars IT budgeted annually, it represents a real loss of $60 billion in annual revenues to software companies." So, even though FOSS is still a small niche

market segment, there are significant financial and strategic impacts on the entire industry.

**Security and reliability**: Whether free software is used within a business or by individual consumers, the software needs to be both secure and reliable. Another important attribute is the speed of software fixes and updates. Software vendors have touted proprietary software as inherently more secure or reliable than OSS. Experience by many industry and government users, however, has not borne out these claims.

There are some who think that free software is easier to hack into because everyone has access to the code. Likewise, the notion has been advanced that since the source code is hidden, proprietary software is therefore inherently more secure. This has proven false, since programmers can hack into proprietary software just as easily as OSS. If hackers really want to get access to proprietary source code, there are also software packages that can decode the source code from the binary executable files. Though not 100 percent accurate, programmers can get a good idea as to what the source code looks like. So, just having access to the source code does not make it possible to hack into systems. Source code availability and security are independent.

On the other hand, FOSS developers claim that since a larger community has access to the source code, there are more people available to look for vulnerabilities than there are hackers who seek to exploit software vulnerabilities. In the case of proprietary software, the community of hackers is likely larger than the potentially small community of developers who have access to the source code. Although the absolute number of developers versus the number of hackers is not a perfect metric for measuring software security, it is nonetheless a factor to consider when evaluating the probability of security breaches.

Some may think that proprietary software is more reliable since there are paid professionals who are updating the code. This theory too is not entirely true since FOSS is updated regularly by professionals. Just because FOSS programmers are not paid by large corporations does not mean that the quality is any less. In addition, some FOSS developers have a business model that includes updating and maintaining business FOSS for a fee. Although the software is free and open source, a company can still create a business model by maintaining the software for others, such as Red Hat and the Free Software Foundation. One test by Bloor Research in 1999 (Boulanger 2005121) compared the reliability of a Windows NT server and a GNU/Linux server. The test showed that Windows system had an uptime of 99.26%, while the FOSS system had and uptime of 99.95%. This may not seem like too much difference, but

over the one-year test, the Windows system had 60 more hours of downtime. This test shows that a FOSS system can be at least competitive with a proprietary system.

## Speed of Software Updates

Another important factor that affects both security and reliability is the speed of software updates and fixes. All software code will be written with errors, usually measured in the number of errors per number of lines of code (usually defects per 100 lines of code. For example a common defect rate is about 1 percent. So, based on the total number of lines of code, "Windows XP and Red Hat Linux would be estimated to have approximately 40,000 and 30,000 undiscovered defects, respectively" (Boulanger 2008). When an FOSS release is made, all users serve as a test-bed to verify the software. In contrast, a proprietary software release has a much smaller test capability, since fewer people have access to the code, and there are limited test resources within the firm. Since the FOSS community is much larger than proprietary software developers, FOSS system flaws are typically patched much faster than proprietary software vendors.

Another argument against proprietary software is that since the code is hidden, the application may have additional functions that the user is not aware of. For example, seemingly benign software could be designed to gather data from the user. The data gathered may be personal or simply non-personal trend data. Regardless of the type of data gathered, however, it should not be done without the user's knowledge. With OSS, it is much easier for users to know the type of information that the software is gathering.

Overcoming the perception of FOSS software as inferior in quality to proprietary software is a task that has been taken seriously by the European Union (EU). The EU has sponsored research initiatives into providing metrics for OSS quality, which would thus permit direct comparisons with proprietary software.

## Perceived Costs of Software

For individual and small business users, software cost of purchase is not the only cost. Just as relevant is immediate usefulness of the software, the ability to produce documents that can be read by a wide variety of recipients, and to do this without incurring additional training costs. For instance, school systems from high school through college teach Microsoft Office, and hiring people who are ready to be productive using these applications is much easier than finding people who are comfortable with OpenOffice, Lotus Symphony, or other free offerings. Training can be

arranged, but this involves extra effort and time. This is as true for individuals (who may not have to be trained on MS-Office) as it is for businesses. Conversion tools for converting OpenOffice documents and Office are readily available (some built into OpenOffice, others feely available online), but users may prefer the convenience of software that does not require any additional downloads, plug-ins, conversion tools, or the extra learning time (however small) involved in mastering them.

Thus, proprietary software enjoys many of the marketplace advantages held by convenience stores. Consumers are willing to pay a premium for convenience and will not forego savings in time and effort for reductions in price.

The idea that the "free" software reduces software cost is due to a conflation of the term "cost" with that of "price." Just as consumers at the convenience store are acting rationally when paying higher prices because they factor time and effort into their overall cost calculation, software purchases must factor in convenience and time (including training and search costs), into their cost model.

However, cost is not the only reason for the growing popularity of open source. Open source software offers more flexibility than proprietary programs, the licenses for which often include restrictions on how they can be used, and companies no longer perceive free software as riskier. Getting sued for running programs that inadvertently violate somebody else's intellectual property, for instance, has proven not to be as big an issue as once feared (Anonymous, 2009).

Software adopters who do not want to invest resources of time and personnel into a FOSS project may find that freedom (from initial software purchase outlays) is outweighed by the convenience of using products in which training has already been invested.

## Economic Logic

Businesses can save money by using free software, as make money, by developing and maintaining free software. Since free software is created and maintained for free, it does not seem possible for business success by selling or distributing software. Clearly, software companies are concentrating on OSS in the hope that it will generate revenue even though they are giving away something that they previously would have sold. Oracle, Microsoft, and other proprietary vendors offer no-cost versions of their software to make it easier for programmers, developers, and implementers of all types to choose their platforms from those vendors. The proprietary software vendors also aim for a business model based on the free razor/profitable razor

blade sales stream, where frequent upgrades and software maintenance contracts are the razor blades.

When Red Hat, MySQL AB, Pentaho, or any other open source-based vendor gives away some or entire software portion of their product, they do so in the expectation that potential customers attracted by the free software will then purchase support contracts, consulting and training services, and other value-added services. Making the software open tends to attract members of a community that were never likely to purchase the software or related services, such as independent developers and companies that are too small, for now, to afford enterprise-level solutions.

So, there are good reasons for much of the ongoing embrace of open source solutions by enterprise software vendors, but we have yet to grapple with the problem of why programmers would develop OSS in the first place. Clearly, it is time for a new economic model. Programmers acting as individuals in a labor market could expect some reasonable payment for the time and effort they expend on developing OSS. They decline that remuneration, and that seems to indicate that they do not operate under the free market model.

## Why do People and Companies Develop Free Software?

There are many benefits of Free Software for non-programmers, such as freedom of choice, peer review of code, price (free!), open standards, and more. However, while those benefits also apply to programmers, there are some problems for programmers such as not getting paid for work (remuneration is zero or minimal), making it more difficult for commercial entities to make money, reducing available jobs for paid programmers (as open source programmers are donating their time).

A majority of software developers who write open source code do it as a part of their jobs (Mobily, 2007). Apache, previously mentioned, was originally written and is still maintained primarily by network administrators and programmers who need reliable, low-cost web server software and believe it's better to pool their efforts than do it alone. Firefox, an open source web browser, is one of the most used freeware software in the history of computers. The Mozilla Foundation, an owner of Firefox, makes most of its money from Google for referrals from the default Firefox home page and built-in search box.

There are many reasons besides charity or the pure joy of creativity that prompt people to write OSS. There's also a substantial and growing developer contingent working on free or open source software that serves as the basis for a commercial software product. OpenOffice development is sponsored by Sun Microsystems. OpenOffice is free, but Sun rolls OpenOffice improvements into its commercial StarOffice suite. MySQL is available either free or in a commercial version with added configuration tools and other proprietary bells and whistles, and at least half a dozen popular web content management and ecommerce packages also fall into the dual-licensed, dual-branded category.

## Trojan Horses?

Michael Tiemann (2008) observes with dismay that Microsoft's embrace of the Open Source concept is actually a corruption of the concept, since its license to open software for noncommercial use is antithetical to the generally agreed-upon definition of Open Source, which clearly rules out restrictions against commercial use.

One issue with software customization is license agreements. For example, a company can create a proprietary version of OSS using a lesser general public license (LGPL). Byfield (2008) mentions an example of how to create a proprietary version of FOSS, where "A company can release a free software version of a product under the LGPL, and use a more restrictive license for a proprietary version of the same product, as Sun Microsystems does with OpenOffice.org and StarOffice" (Byfield 2008). By using this type of license, free software can be turned into proprietary software. Although this was not the original intent of free software, it can provide a path to create a competitive advantage for a firm.

Sun Microsystem's sponsorship of OpenOffice illustrates another aspect of strategic use of FOSS. It has been maintained that Sun, whose main line of business is not desktop software, has used its sponsorship of "free" OpenOffice as a weapon against Microsoft, not as a head-to-head competitor in desktop software, but as a competitor in the server business (Wichmann, 2002). Therefore, FOSS software sponsorship by for-profit companies may be used as indirect as well as direct attacks against competitors in the software industry.

## Closing Observations

This paper has attempted to identify areas of competitive advantage associated with the use of free software as well as areas that seem problematic. The idea of FOSS has been around since the invention of computer hardware, well before the software patent laws of 1980. However, since the implementation of the software patent law, the purpose of FOSS has been to allow for continued software collaboration to develop software to meet the needs of personal and business computer users. Proprietary software is not transparent, so it is possible for it to have hidden functions that the user is not aware of. FOSS can often compete with proprietary software in terms of performance, security, and reliability. In addition to being a viable option for personal use, free software can be a viable business solution for inexpensive information systems infrastructure, no matter what the size of the business is.

So far, we have considered adoption of FOSS in the context of businesses that are large and sophisticated enough to become active participants in the FOSS community, but future research would do well to consider individual and small businesses software users of software (such as database or desktop applications as well as larger businesses that do not wish to participate in the ongoing development of software platforms or sophisticated applications. Cost, for these users, may entail more than the initial software price, and restrictive license agreements that prohibit source code modification may have no relevance to their adoption decisions. These users have no desire to modify the software they purchase, but instead may value other factors such as ease of use and the ability to share documents with other users over the Internet.

Moreover, while lack of familiarity with low-cost software alternatives might cause this class of smaller-scale users to spend money on brand-name software that has been heavily promoted by expensive marketing campaigns, it is not at all obvious that FOSS presents a more attractive or even cost-effective solution when their priorities are considered. Indeed, familiarity with the use of FOSS as strategic weapons by companies such as SUN could be additional reasons for them to doubt whether the aim of corporate FOSS sponsors is to produce high quality software or whether the software merely represents a short-term ruse to sabotage marketplace competitors.

## References

Anonymous (2009), "Business Born Free; Open Source Software in the Recession," The Economist, 391:8633, May 30, pp. 69.

Asay, Matt (2008), "The Dying Embers of Microsoft's IP Claims against Open Source" October 20, http://news.cnet.com/8301-13505_3-10070008-16.html.

Boulanger, A. (2005). "Open-source versus Proprietary Software: Is One More Reliable and Secure than the Other?" IBM Systems Journal, 44:2, pp. 239-48.

Byfield, Bruce (2008). "Business vs. FOSS: Six Pressure Points," http://itmanagement.earthweb.com/osrc/article.php/3785311/Business+vs.+FOSS:+Six+Pressure+Points.htm.

Daffara, Carlo & Jesus M. Gonzalez-Barahona (Eds) (2000). "Free Software/Open Source: Information Society Opportunities in Europe?," http://eu.conecta.it/paper.pdf.

Ebert, Christof (2008). "How Open Source Tools can Benefit Industry," IEEE Computer Society, 26:2, March/April, pp. 50-51.

Forrester Consulting (2007). "Open Source Software's Expanding Role in the Enterprise," https://www.unisys.com/products/insights/insights__compendium/open__source__software__expanding__role__in__the__enterprise.htm.

Free Software Foundation, "What is Copyleft?" http://www.gnu.org/copyleft/

Howells, Ian (2009). "The Top 5 FUD Myths Used against Open Source," ComputerWorldUK, http://www.computerworlduk.com/community/blogs/index.cfm?blogId=16&entryId=2019, March 23.

King, Rachel (2008). "Cost-Conscious Companies turn to Open-Source Software," Bloomberg BusinessWeek, December 1, http://www.businessweek.com/technology/content/nov2008/tc20081130_069698.htm.

Mobity, Tony (2007). "So, why, why do People and Companies Develop Free Software?" Free Software Magazine, Issue 20, November 7, http://www.freesoftwaremagazine.com/articles/editorial_20.

Pearlman, M. H. (2009). "How Using Open-Source Software Can Affect Your Company's Value," *CIO*, December 16, http://www.cio.com/article/511115/How_Using_Open_Source_Software_Can_Affect_Your_Company_s_Value.

Rangaswami, J. (2008). "Learning about why People don't Adopt Opensource," October 21, http://confusedofcalcutta.com/2008/10/21/learning-about-why-people-dont-adopt-opensource/.

Rupley, Sebastian. (2009). Flexible, Open-Source Telephony Resources. Bloomberg BusinessWeek, October 25, http://www.businessweek.com/technology/content/oct2009/tc20091023_077445.htm.

Shankland, Stephen (2006). "Oracle Tried to Buy Open-source MySQL," February 15, http://www.builderau.com.au/news/soa/Oracle-tried-to-buy-open-source-MySQL/0,339028227,339234504,00.htm,

Stallman, Richard (2007), "Why Open Source Misses the Point of Free Software," http://www.gnu.org/philosophy/open-source-misses-the-point.html.

Wichmann, Thorsten, and Glott, Ruediger (2002). "Free/Libre and Open Source Software: Survey and Study," FLOSS FINAL REPORT, Berlecon Research GmbH Berlin, http://www.flossproject.org/report/FLOSS_Final4.pdf.

Working Group on Libre Software. (2000). "Free Software/Open: Information Society Opportunities for Europe?," Version 1.2, http://eu.conecta.it/paper.pdf

Note: Disc graphic designed by Carole E. Scott