

Policy Implications of Open Source Software

Danielle Fallon
August 30, 2004

STS 395: Technology in Culture
Professor Russell

“On my honor as a student, I have neither given nor received aid on this assignment.”

Danielle Fallon

Introduction

The proprietary software model championed by Microsoft is the most familiar model of software development; however, lately attention is focusing on an alternative, the open source software model. Recently, open source software is gaining a significant market share of the software industry's 300 billion dollars in annual revenue, posing a direct threat to the proprietary model (E-commerce and Development, 2003, p. 95). Open source software grants users the freedom to access, alter, and redistribute programs by releasing not only the binary code of zeros and ones, but also releasing the source code. Proprietary companies only distribute the binary codes. By releasing the source code written in programming languages such as C++, Java, or Fortran users can modify the open source code to fit their individual or business needs. Recent international government action shows a preference for open source with certain local and national governments supporting or mandating the use of open source. Although deliberate preference for open source by the U.S. government is discouraged, the debate over open source software versus proprietary software goes beyond merely enhancing IT infrastructure or minimizing costs. It encroaches upon serious intellectual property, cyber security, and procurement concerns. The implications of open source are far reaching; therefore an understanding of the history, law, economics, and business is necessary to understand the public policy implications.

Background

The earliest computers that emerged in the 1950's were cumbersome mainframes with software written in binary designed to work only on a specific type of hardware. Early developers relied on sharing programming code because the task of coding on the complex and unreliable computers was insurmountable alone. The famous Moore's Law developed by Gordon Moore, the founder of Intel, states that the amount of data a microchip can hold doubles every eighteen months. This law accounts for the rapid improvements in hardware performance, scalability, and cost. In 1953, the first marketed commercial computer called the IBM 705 sold for an average price of \$1.6 million but by 1970 the popular computer called PDP-11 developed by Digital Equipment Corporation (DEC) sold for merely \$11,000 (Weber, 2004, p. 21-23).

Due to the declining cost of computers, Ken Thompson, a Bell Lab researcher, was able to buy in 1969 an earlier DEC computer, the PDP-7, and spent four weeks writing an operating system called UNIX. UNIX revolutionized the future of software. It was not only written in human-understandable source code but it was a step away from using a hardware-specific operating system into using software capable of being transported onto many different types of computer hardware. Because Thompson's employer, Bell Labs, was part of AT&T at the time of UNIX's development, neither Bell Labs nor AT&T could capitalize on Thompson's work. A Sherman antitrust suit settled by the Justice Department on January 24, 1956, under the Eisenhower Administration declared that AT&T could not manufacture or sell goods that were not "common carrier communications" services (Weber, 2004, p. 22). UNIX was not considered a communication device, thus AT&T decided to give away or license the UNIX code for a nominal fee to universities, research facilities, commercial users, and military organizations.

Robert Fabry, a computer science professor at the University of California Berkeley was interested in the UNIX software and received a copy from AT&T. Fabry's windfall came when in 1975 the UNIX creator, Ken Thomson, decided to take a sabbatical from Bell Labs and work at Berkeley in cooperation with Fabry marking the beginning of cooperation with Berkeley and Bell Labs. This collaboration at Berkeley was the first significant open source effort. Berkeley contributed and redistributed UNIX based on a license they developed with Berkeley graduate student, Bill Joy, the Berkeley Software Distribution (BSD). This license is one of the prominent open source licenses still used to today for the redistribution of open source software and allows individuals and corporations freedom to use, contribute, or redistribute software under certain restrictions.

Initial copies of UNIX were redistributed by sending floppy disks in the mail; however, the development of the Internet significantly aided the dissemination of UNIX. The Department of Defense developed the first network called ARPANET (Advanced Research Projects Agency Net) in 1968. ARPANET served to coordinate military research by linking universities, defense contractors, and researchers over a small network. The primitive network encountered problems in the late 1970s because most of the software running computers linked in ARPANET were becoming obsolete.

Compatible software, especially an operating system, was needed so that all the computers could run on the same infrastructure. UNIX with its portability features was the most viable solution and in 1979 Professor Fabry at Berkeley offered to develop a specific version of UNIX for ARPANET establishing UNIX as a central figure in the emergence of the Internet.

UNIX and open source software in general appeared to be on the rise but events surrounding a U.S. Government antitrust suit at the conglomerate AT&T, Western Electric, and Bell Labs stalled development. The suit alleging monopolistic behavior resulted in the 1982 decision by Judge Greene to separate AT&T's operating companies; the Baby Bells and Bell Labs were disconnected from AT&T while Western Electric was eliminated. AT&T, no longer bound by the earlier 1956 decision that limited the company from manufacturing and distributing UNIX, created a new division called the UNIX Systems Laboratory to quickly recover the losses in its prior inability to capitalize on UNIX. AT&T charged outrageous licensing fees between \$100,000 and \$250,000 and forced Berkeley to remove six key code sections that were originally part of the AT&T UNIX code (Weber, 2004, p.39). AT&T sought commercialization of UNIX for a profit diverging from Berkeley's open source effort. Berkeley's version of UNIX was incomplete without the AT&T core code sections. A computer programmer, Bill Jolitz, revived Berkeley's effort by writing the replacements for the removed AT&T code in Berkeley's version of UNIX. Even though he is often not credited for his work, many believe Jolitz to be the first to launch the open source UNIX-based operating system in the PC market via the Internet (Weber, 2004, p. 42).

In 1991 AT&T threatened Berkeley with a lawsuit stating that Berkeley's version of UNIX used AT&T code even though Berkeley removed the six code sections earlier. However, the University of California-Berkeley followed suit stating that AT&T used Berkeley code without crediting its source. The case never went to trial and AT&T sold its version of UNIX to another company called Novell. The events resulted in people being leery of using UNIX because of the lawsuit and the growing number of versions of Unix. The Berkeley version alone was fragmented into various versions including FreeBSD, OpenBSD, and NetBSD in addition to AT&T's versions. The battle between AT&T and Berkeley along with the fragmentation of UNIX allowed for the commercial success of the proprietary model.

Bill Gates began establishing the proprietary model as early as 1976 with his letter to computer programmers called “open letter to hobbyists” where he accused coders of stealing software. Bill Gates wrote. “As the majority of hobbyists must be aware, most of you are stealing software” (Weber, 2004, p. 36). Gates was upset because he had written a computer language called BASIC and other programmers were merely copying the code instead of paying his company, Microsoft, founded the prior year, 1975. Many programmers enticed by the potential profit rewards from the commercial model joined Gates’ effort. The arrival of the personal computer (PC) in the early 1980s provided the demand for the commercial model and Microsoft is still the leader today in PC operating systems with Windows.

The growth of the Internet in the 1990s and the emergence from the recession during the Cold War brought computing and the Internet to the “front and center to corporate investment, economic growth, and an intellectual trajectory of arguments about a ‘new economy’” (Weber, 2004, p. 45). Open source software not only provided much of the infrastructure for the Internet, for example, the vast majority of e-mail routed over the Internet is done through open source software, but the Internet itself provided a development platform allowing programmers to collaborate over the web in developing open source projects. The Internet along with the work of a particular programmer incited the reemergence of open source software.

In 1990, a 21-year-old University of Helsinki computer science student named Linus Torvalds, pictured in Figure One, was working on a UNIX clone called Minux developed by Professor Andrew Tanenbaum in Amsterdam. Torvalds decided to develop his own UNIX-like operating system apart from Minux and in the autumn of 1991 released the first version of his operating system, Linux. The program was distributed as open source under a license developed by Richard Stallman called the General Public License (GPL), an open source license with different restrictions than the Berkeley Software Distribution (BSD). Torvalds’ innovative Linux operating system, also referred to as GNU/Linux is arguably the first successful open source effort and led to the development of open source programs worldwide as well as successful business ventures such as VA Linux and Red Hat Inc.,



Figure One: Linus Torvalds
Source: www.linux.org/info/linux.html

companies that had two of the most successful initial public offerings on NASDAQ in 1999 (Weber, 2004, p. 55). The success of Linux along with other open source projects is drawing the attention of not only programmers worldwide but the attention of the United States government as it seeks to understand what open source is and its implications.

Understanding Open Source

Open source software or OSS is often confusingly called free software because users have the freedom to run, improve, and distribute code freely, but open source software is not necessarily cost free. Software business ventures like Red Hat Software charge for packaged software with product support; however, most open source projects available via the Internet have no cost. The belief is that instead of an exchange for a price, there is an exchange for the new user to contribute and improve the code. The only restrictions are that any modified code must be distributed under the same license as before. For example, programmers contributing to Linux under the GPL License must in turn redistribute their source code modifications under the same license. Proprietary models only grant you the ability to use the software you license “just like a rental from Hertz gives you the right to use the car you’ve leased” (Hahn, 2002, p. 53). With open source, one could not only use the Hertz rental car but could modify the car perhaps by painting the outside a different color.

A common analogy to Coca-Cola helps elucidate the distinction between open source and proprietary software. The list of ingredients including carbonated water, citric acid, and natural flavors in Coca-Cola is displayed on every can and bottle. This list, the fundamental make-up of Coca-Cola, compares to binary code, the fundamental zeroes and ones that instruct the computer. By distributing only the ingredients or the binary code, Coca-Cola and proprietary software companies prohibit others from being able to modify and redistribute their work. Someone drinking Coca-Cola cannot “understand it in a way that would let you reproduce the drink, or improve upon it and distribute your cola drink to the rest of the world” (Weber, 2004, p. 3-4). On the other hand, the software recipe, or source code, is distributed with open source software so that users worldwide have the freedom to run, modify, and redistribute the software. If people had

the recipe for Coca-Cola they could make soda, modify the ingredients, and also redistribute the soda.

Model of Production

By distributing their recipe or source code, open source software is relying on the contributions of programmers worldwide to read the code and improve upon it resulting in a very different model of production than the Fordist style of production used in commercial software. In the Fordist style of production there is a clear division of labor and segmentation of tasks. The structure is hierarchical relying on management to oversee and ensure the success of the project. Open source software, on the other hand, relies on a much broader and international support base primarily programmers who volunteer their time to OSS projects. Eric Raymond, author of the famous *The Cathedral and the Bazaar* compares the building of a cathedral, extensively planned, hierarchical, with a distinct division of labor to commercial software production. He states that the open source model of production is “a great babbling bazaar of different agendas and approaches” that happens at a “speed barely imaginable to cathedral-builders” (p. 21-22). Instead of a manager assigning employees to distinct tasks, OSS relies on a peer-to-peer structure where human capital is allocated according to the individual programmers contributing. Individuals can choose where to contribute based on their talents and interests. Yochai Benkler in the *Yale Law Journal* believes:

[I]f peer production has a sufficient advantage in terms of its capacity to process information about who the best person is for a given information production job over firm and market based mechanisms to outweigh the costs of coordination, then peer production will outperform firms and markets (Healy & Schussman, 2003, p. 6).

The myriad open source software programs to date reflect the success of peer production and the less hierarchical model of open source production as described by Benkler and Raymond.

Many believe open source will have disruptive changes in the current U.S. model of production in the information technology sector as well as other sectors of the economy much like the disruptive effects of the Internet on commerce, government, and security. Debates in the policy arena about possible effects of open source are just

beginning while many companies have already moved towards using open source software in their enterprises. Policymakers must be aware of what open source software programs exist, their threat to commercial software, as well as their growing deployment in the marketplace.

Open Source Programs

Linux, the popular open source operating system developed by Linus Torvalds, runs on everything from wristwatches, mainframes, hand held devices, cell phones, to the home video recorder Tivo. As quoted in a 2003 Business Week article, “Nearly 40 percent of large American companies use GNU/Linux in some form” and GNU/Linux runs on 29.6% of world’s Web servers which handle tasks like printing and storing on a network (E-commerce and Development, 2003, p. 104-105). Linux is also expanding to the personal computer operating system as other home and office open source programs like OpenOffice, a business application suite that provides word processing, spreadsheet, and drawing capabilities much like Microsoft Office, are being perfected. Open source programs other than OpenOffice and Linux have a significant presence in the software industry as well. For example, the open source Apache web server has over 65% of the web server market worldwide and Sendmail, a e-mail transfer and management program, routes the vast majority of the world’s e-mails over the Internet (Weber, 2004, p. 6). In addition, other common open source programs include MySQL, a popular database system, Gaim, an instant message service, and GIMP, an image editor like Photoshop.

Open Source Threat and Users

Open source software products, including the before mentioned, pose a threat to many of the existing proprietary models because many of the open source products mirror proprietary counterparts. For example, the database system MySQL poses a threat to Oracle’s database system and the open source instant messenger service, Gaim, poses a threat to AOL’s instant messenger. Commercial software operations recognize the risk of open source, especially Microsoft. On October 31, 1998, a memo from Microsoft later termed “the Halloween Memo” from a high-level member recognizing the threat of open source to Microsoft stated that “Linux and other OSS advocates are making a progressively more credible argument that OSS is at least as robust- if not more- than

commercial alternatives” (Weber, 2004, p. 126). The memo also recognizes the innovative possibilities of having hundreds of volunteer programmers contributing to the open source effort. As stated by Microsoft employees and quoted by Weber in *The Success of Open Source Software*:

The ability of the OSS process to collect and harness the collective IQ of thousands of individuals across the Internet is simply amazing. More importantly, OSS evangelization scales with the size of the Internet much faster than our own evangelization efforts appear to scale (Weber, 2004 p. 126).

In addition to Microsoft, many companies including software purchasers are recognizing the possibilities with OSS and are using or switching to open source software.

Financial giants Merrill Lynch, Morgan Stanley, Credit Suisse First Boston, and Goldman Sachs all deploy open source software. According to the Open Source Software Initiative, a non-profit organization that promotes open source, “84 percent of Fortune 100 companies rely on open source as a critical component of their infrastructure” (2003, p. 4). Many companies run on the Linux operating system. In fact, “Nearly 40 percent of large American companies use Linux in some form” (Weber, 2004, p. 6). British Airways, Cisco Systems, Nikon, Regal Entertainment, the American Movie Theater Company, Home Depot, Toyota, Fidelity, Verizon, and Reuters all use Linux. Movie companies, Disney, Dreamworks, and Pixar use Linux for the special effects seen in movies such as Titanic, Lord of the Rings, Monsters Inc., and Shrek. Companies such as Amazon, E-trade, Reuters, and Merrill Lynch also use the open source Apache web server while Yahoo, Motorola, and Texas Instruments use the open source database, MySQL. These companies chose open source based on its benefits to their enterprises.

Benefits of Open Source Software

The benefits realized by corporations as well as individuals and governments using open source are primarily a result of the open source model of production. By relying on an international and predominantly volunteer programmer base, OSS offers a clear cost advantage that allows companies to achieve significant economy of scale by deploying open source software on multiple computers. For example, Sabre Holdings, a company that supports Travelocity, decided to use open source operating system, Linux,

and JBoss open source applications because additional licensing fees were not required on each additional computer installed. The CTO of Sabre Holdings, Craig Murphy, comments that he's "expecting at least an 80 percent reduction in running cost" considering open source's economy of scale (Wheatley, 2004).

Open source software also reduces an organization's dependence on a single software vendor. Due to the low or nonexistent licensing fees for open source, companies can easily switch to open source without paying the high licensing fee to commit to one version of software. Proprietary companies also often force high priced upgrades whereas open source updates are available for free to download off the Internet. In addition, proprietary companies usually discontinue all support for outdated legacy programs whereas open source has a wide based community support that is more viable for continued service and support.

Another obvious benefit of open source is its transparency. Programmers can debug code along with tailoring it to certain needs. Thus many believe that open source is more secure because it is being tested by more people. Lastly, open source taps into a broad base of international talent and motivation to foster innovation that is not always seen under the commercial model.

Drawbacks of Open Source

A result of the wide programming base is that open source software is geared toward the technophile, IT professionals that understand programming languages, instead of the technophobes or people with little knowledge in computers or computing. Many believe open source software is still behind the proprietary model in its interface and ease of use because the end user is one generally more concerned with the program itself and not aesthetics. As Hahn mentions in *Government Policy toward Open Source Software*, "Open source has found acceptance where the software is generally used by information-technology specialists...or where the software does not require interaction with consumers, as is the case with Linux on cell phones" (p. 40). Commercial software companies, on the other hand, gear software to the end user and perform extensive research on the wants and needs of its customers. One must consider the newness of open source software, however, for initial versions of Microsoft were not user friendly. As Hahn notes, "Microsoft took over a decade to deliver products easily grasped by the

untutored” (2002, p. 17). Current OSS programs are relatively new and are still developing a niche in the software industry.

The informal programming base of OSS also does not guarantee development of certain software because programmers decide where they want to contribute. Many open source projects lack the needed support. As stated in *Government Policy toward Open Source Software*, “Until the project reaches a critical mass, it may be unattractive to many developers; developers may choose to wait for others to serve as pioneers” (Hahn, 2002, p. 24). Programmers also may lose interest in a project or are only drawn to exciting projects, not necessarily ones that fit the needs of end users. In regards to security, many argue that the transparency of open source increases the security of OSS software; however, opponents argue transparency leads to code more vulnerable to security breaches because the source code is exposed.

The informal programming base of OSS also poses support and legal responsibility concerns. Opponents argue that the lack of vendor or technical support for open source is a significant failure of open source; however, Bob Bickel at JBoss believes this is an argument only commercial companies use because JBoss and many other companies are committed to primarily open source support (Bickel, 2004). In addition, there are some legal support structures for open source. On November 17, 2002, the JBoss Group announced it would defend JBoss customers from legal action, including possible copyright infringement lawsuits, along with HP, Red Hat, and Novell (Wheatley, 2004). Opponents argue that the accountability of OSS is still vague. According to Microsoft Chief Technology Officer and Senior Strategist, Craig Mundie, it is uncertain whether responsibility “lies with the collective of programmers who contribute the code, the service providers who help customers implement and maintain the software, or the customers themselves.” (p. 5). Microsoft and others are also concerned about the intellectual property and open source licensing, specifically the GPL license.

Open Source Licenses

The two most prominent licenses are the General Public License (GPL) and the Berkeley Software Distribution (BSD) with the main distinction being the restrictions on redistributing the software. The General Public License requires that all GPL code stay in the public domain; therefore it is often termed “viral” because the presence of GPL

code in any program requires that the whole program be placed in the public domain. Proprietary companies cannot use GPL code in their work without distributing all the source code. The GPL is often termed copyleft as a pun on copyright because instead of inhibiting people from distributing work, the GPL forces people to distribute work to all. The Free Software Foundation founded in 1989 under MIT researcher Richard Stallman decided to make the GPL “restrictive” based on the belief that all software should be free. The restrictive clause inhibits fragmentation because all code is publicly available. Andrew Morton, one of the lead maintainers of Linux, states that the GPL license is “the whole thing that keeps the system running” (Council, 2004, p.13).

An alternative to the GPL, the Berkeley Software Distribution only requires acknowledging the contributors to the source code but allows users to freely modify and distribute the code as they wish. In other words, users can put BSD code in a commercial program and distribute the code for a profit instead of in the public domain. The BSD license is therefore termed a “permissive” license. Although the BSD license is not considered viable on large scale projects like the GPL with Linux, it has more benefits for commercial companies and companies that wish to significantly profit from open source software.

When the GPL or BSD does not fit the needs of companies or individuals, new open source licenses are made. Currently there are over seventy recognized open source licenses including the Artistic License, the Mozilla Public License, MIT License, IBM Public License, and the Common Public License. Another common open source license is the LGPL License also termed Library GPL or “Lesser” GPL written by Richard Stallman. JBoss uses the LGPL License because it is more “business friendly” allowing the company to call GPL licensed libraries of code while keeping the core code proprietary (Bickel, 2004). The LGPL has allowed JBoss to create a successful open source application business.

Intellectual Property

The impact of open source software especially GPL licensed software raises concerns about implications on our intellectual property system. The proprietary model uses patents and copyrights to ensure a return on their knowledge creation which has not only allowed the U.S. to become a world leader but has ensured the success of

commercial software companies. Craig Mundie of Microsoft states, “Whether copyrights or patents or trade secrets, it was this foundation in law that made it possible for companies to raise capital, take risks, focus on long term and create sustainable business models” (Lohr, 2004). To support large research and development costs, companies have relied on intellectual property protection to get a return on their investment. Software companies spend billions of dollars on research and development. “For instance, in 1998 the U.S. software and computer services invested an estimated \$14.3 billion in R&D which exceeded the level of R&D spending by the U.S. motor vehicle, pharmaceutical, and aerospace industries” (Hahn, 2002, p. 78).

Microsoft has also benefited from government funded research under the Bayh-Dole Act, also termed the Trademark Act of 1980, which promotes technology transfer from government funded research to the commercial marketplace by offering ownership of inventions researched in universities and exclusive licenses to industry (The Bayh-Dole Act, 1999). Software industries rely on a significant amount of governmental funds for research and development and are concerned about governmental spending on open source GPL projects. Bill Guidera, Policy Counsel with Microsoft, along with many opponents argue that government funding of open source projects under the GPL limit the ability to profit from government-funded research; commercial companies would be discouraged from making investments if their intellectual property must be made freely available to competitors (Hahn, 2002, p. 47). Bradford Smith supports this belief commenting that the “use of the GPL in publicly funded research projects would drive an impenetrable wedge between the public and private sectors, thereby undermining the innovation and economic growth that has resulted from such public-private collaboration in the past” (Hahn, 2002, p. 79). Morgan Reed from the Association of Competitive Technologies argues that representatives must seriously consider if they should use taxpayers’ money to fund projects under the viral GPL because it “does not necessarily return intellectual property to the United States” (Council, 2004, p. 42). There is a concern that U.S. support of GPL licensed projects could hinder instead of encourage intellectual property creation in the United States.

Many open source software proponents argue that the current intellectual property (IP) system stifles innovation and that the open source model of development is more innovative because it focuses on protecting property of the user not the developer. As

Weber mentions, “Property in open source is configured fundamentally around the right to distribute, not the right to exclude” (Weber, 2004, 16). The growth of digital technologies in general aids copying and distributing digital work including software via the Internet. Napster, although violating copyright laws, created a successful process that provided the framework for individuals to share songs but relied on the contributions of others. Hahn believes, “Rather than encouraging creativity by limiting distribution, Napster encouraged distribution and assumed that creativity would take care of itself” (2002, p. 230). Proponents of IP that revolve around encouraging distribution like Napster and open source believe that restrictive distribution laws limit innovation, arguing that we are “going towards having books with embedded chips that charge you for every page you read” (Hahn, 2004, p. 2444). Many proponents of open source believe that a reexamination of our intellectual property system is necessary.

Patent Reform

President Lincoln stated, “The patent system added the fuel of interest to the fire of genius” (House Subcommittee, 2002, p.1). Indeed patents have spurred innovation that has led to U.S. dominance; however, concerns over patent thickening and the effect of patents on information technology should encourage U.S. policymakers to reconsider the patent process. Since drastic patent reform in the 1980s that led to the extension of patent terms, there has been a drastic increase in the number of patents. As shown in Figure Two, the “number of patents soared to almost 20,000 per year, and a total of nearly 100,000 have been accumulated to date” (Hahn, 2002, p. 27).

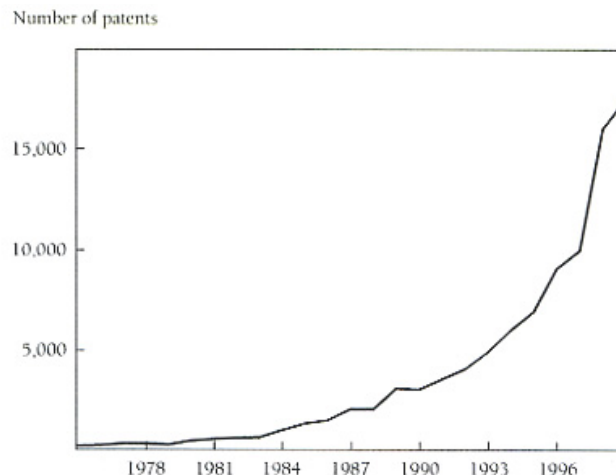


Figure Two: Number of Patents 1978-1996
Source: Hahn, 2002, p.27

Firms acquire many patents as a defense to prevent innovation and competitiveness from rival firms. As quoted in a memo written by Bill Gates, “A future start-up with no patents of its own will be forced to pay whatever price the giants choose to impose. That price might be high. Established companies have an interest in excluding future companies” (Hahn, 2002, p. 28). Individual open source programmers simply cannot develop large patent portfolios nor have the ability to pay high litigation costs.

Patents are often overly broad and are granted for insignificant advances that allow “firms to acquire large patent portfolios with expenditure of money but without commensurate invention” (Hahn, 2002, p. 29). A 1952 statute states that a patent should not be obvious to someone with “ordinary” skill in that arena, yet it is difficult to conceptualize the function of certain software programs, and often software compounds upon previous ideas and innovates rapidly. Mark Webbink, Senior Vice President & General Counsel for Red Hat Inc., states that the open source community believes “patents on software have actually stifled innovation, rather than promoted it, because software development occurs at a much more rapid pace than one finds in the other patent arts” (House Subcommittee, p. 22-23). The lifecycle of software is drastically shorter than the patent term, thus software is usually obsolete by the time the its patent expires.

The U.S. Government needs to consider revising the IP system to allow for equal ground for both software development models. As the Stanford economist, Lawrence Lessig believes “the government should not allow bloated intellectual property regimes to tilt the field of competition against one of the most vibrant competitors” (Hahn, 2002, p. 67). Through not only research funds but patent reform the U.S. can support open source software by perhaps making an organization that applies for open source patents or seriously monitors the patents being created. U.S. policymakers and representatives must also go beyond intellectual property and understand the security issues surrounding open source software.

Cyber Security

As mentioned earlier, the security of open source software is debatable, with opponents arguing that OSS is more vulnerable while proponents argue that open source is more robust because it is tested by many programmers on a regular basis. Recently security breeches in Microsoft such as the Blaster, Sasser Worm, and MyDoom show that

commercial software is not invincible to attack (Fallon, 2004, p. 4). Security is at the forefront of the U.S. Government's agenda as it handles important information such as social security, health, taxation, and voting records and as governmental agencies depend on the security of their information especially the Securities and Exchange Commission, Internal Revenue Service, Federal Reserve Board, the Federal Bureau of Investigation, Central Intelligence Agency, and the Defense Department. The U.S. Government must seriously consider security concerns when choosing to use certain software or in devoting money to software research and development.

While the commercial model follows the motto "security through obscurity," open source believes that the peer review nature of development allows for a wide programming base that can test and improve code under a variety of conditions. Proponents also believe that because of the international development base, users receive a fast response time because coders are working worldwide twenty-four hours a day. Eric Raymond, author of *The Cathedral and the Bazaar*, believes "given enough eyeballs, all bugs are shallow" which he dubs Linus's Law after the Linux creator, Linus Torvalds (E-Commerce & Development, 2003, p. 102). Open source "bugs" have more people examining the code, therefore, the coalescence of many people makes bugs easier to fix.

There is also a growing desire for more transparency in governance. As quoted in E-commerce, "The need for open public data formats is directly relevant to calls for increased accountability and transparency in public sector governance" (2003, p. 111). The transparency of open source, however, requires that code is available to all. There is no way to discriminate against mal-intentioned users who want to create bugs. Security goes beyond merely open source and commercial software, however, and has to do with implementation, use, and policy. Craig Mundie, CTO and Senior Strategist at Microsoft, concludes, "Security issues in software are an industry-wide problem, and flaws are no more or less prevalent in software produced under either of the models" (p.4). Many people doubt the security of open source because they are unsure who is contributing to the open source project and how the open source community is structured.

Open Source Motivations and Community

Andrew Morton, lead maintainer of the Linux kernel, stated at an Open Source Software Briefing in July on Capitol Hill that although the stereotype is the “male computer geek working in his basement writing code in his spare time, purely for the love of his craft” increasingly the core developers are employed professionals (Council, 2004, p.7). Morton’s employer, the Open Source Software Development Labs, an industry consortium, supports some developers but the majority comes from open source distributors such as IBM Corp, Red Hat, VA Linux, and SGI (Fallon, 2004, p. 4). The majority of contributors are professional programmers, IT managers, or system administrators. In regards to demographics, open source programmers tend to be between 22 and 37 (Weber, 2004, p. 135) and the average programmer has ten years of experience (Hahn, 2002, p. 17).

Programmers are motivated to contribute code in a community where status is gained from contributing impressive code to further the project. A 2001 survey by the Boston Consulting Group found that one-third of open source developers are motivated by the belief that source code should be open; one-fourth are “fun-seekers”; one-fifth say open source helps them professionally; and one fifth say open source programming helps them to learn coding (E-Commerce & Development, 2003, p. 108). Only 11.3 percent commented that they were motivated by a desire to beat proprietary software (Weber, 2004, p. 139). Andrew Morton, lead maintainer of the Linux code kernel, believes that “open source does tend to attract the very best developers” (Council, 2004, p. 10). Weber states that open source “suggests a mechanism for self-selection, because it is clearly the best programmers who have the strongest incentive to show others just how good they are” (2004, p. 143). In addition, a small group of the best programmers tend to write the majority of the code for open source projects.

Morton mentioned that in the past three years Linux has had 38,000 changes to the core Linux kernel made by 1,000 contributors but about twenty of those contributors made half the changes (Council, 2004, p.7). The Apache Web server also had the majority of its code contributed from a small percentage of core developers. The Apache project “was driven by about fifteen core developers, who contributed 85-90% of the code, surrounded by a larger penumbra of participants” (Healy & Schussman, 2003, p. 8). The Orbiten Free Software Survey done in 1999 and 2000, arguably the most complete

survey taken, analyzed the credits in open source software concluding that “the most prolific ten percent of developers are credited with about 72 percent of the code” while “the top ten individual authors are responsible for almost twenty percent of the total code” (Weber, 2004, p. 71). The model of software production is not quite the bazaar that Eric Raymond described, but one that weighs heavily on a few core programmers devoted to the open source project.

The majority of open source projects are managed and organized on the web where developers can communicate via e-mail and forums or download code. SourceForge, the largest sites for open source projects, has over 600,000 participants registered and projects are classified in Planning, Pre-Alpha, Production/Stable, and Mature phases (Weber, 2004, p. 66). Other sites include Slashdot, Freshmeat, and Kuro5hin. All discussion and conflict is handled over the Internet and “ultimately the discussions get down to the technical issues, clarifying the trade-offs among design strategies and decisions that matter for the function and performance of software” (Weber, 2004, p. 102). Most projects are also divided in a hierarchical fashion. Linus Torvalds, the leader of Linux, delegates components and sub-systems to lieutenants who make up the “inner circle” and then their work is delegated to area-owners and maintainers. The vast majority of code, specifically for Linux, is “tested by hundreds or thousands of people” and all contributions to the Linux kernel are personally tested and approved by Andrew Morton (Council, 2004, p.13). The Apache Group makes code decisions by an e-mail voting system. Only votes cast by the Apache Group programmers are binding and the rest just serve as opinions.

International Procurement

Many international governments trust the security of open source software and are showing preference for or mandating open source software in procurement decisions. This has raised concern with U.S. policymakers and representatives. Some international governments choose open source to avoid dependence on American software companies under U.S. jurisdiction. As stated in *The Success of Open Source*, “No national government, if it had alternatives, would have chosen during the twentieth century to accept dependence for steel or petroleum on a single supplier or a small number of suppliers based in a potential rival nation” (Weber, 2004, p. 253). In addition, some

international governments are opting for open source because of reduced costs, ability to customize the software to local languages and cultures, and out of apprehension in using commercial American products like Microsoft with confidential e-government or military information (E-Commerce & Development, 2003, p. 110). International governments want to create IT exporting opportunities as well as cultivate a domestic software industry by choosing open source. The ability of domestic talent to contribute to open source “makes it possible to keep IT expenditures, as well as experts and promising young talent, at home and contributing to a nascent local software industry” (E-Commerce & Development, 2003, p.110) Many countries have already passed legislation or are considering legislation that preferences open source or encourages its development.

In May 2004 the city of Munich switched its 14,000 computers to Linux under a \$42 million dollar contract. The European Parliament called for its members to promote open source software whenever feasible (Hahn, 2002, p. 4) while the government of Singapore offers tax breaks to companies that use the open source Linux operating system and Germany offers discounts for using IBM Linux software (E-Commerce & Development, 2003, p. 116). In November 2001, the Government of Malaysia committed to using open source in Treasury and e-procurement. France’s ministry of Defense, Culture, and Economy along with Germany’s Federal Institute for Agriculture and Food uses open source operating systems (E-Commerce & Development, 2003, p. 116). In September 2003, the Chinese, South Korean, and Japanese governments agreed to collaborate on an open source project referred to as the Japan-China-Korea Open Source Software Promotion Partnership. India’s Department of Information Technology is encouraging standardizing academic institutions with open source software (E-Commerce & Development, 2003, p. 110). In Peru, Bill 1609, The Use of Free Software in Public Administration, introduced by Edgar Villanueva mandates the exclusive use of open source on all governmental computers. In addition, the governments of Italy, Spain, and Argentina are considering legislation that mandates or shows a deliberate preference for open source software.

Legislation passed in 2001 in Brazil mandates the use of open source software in six Brazilian municipal governments: Recife, Campinas, Solonopole, Amparo, Sao Carlos, and Porto Alegre (Hahn, 2002, p. 5). The Administration of Brazilian President, Luiz Inacio Lulu da Silva states that commercial licensing fees are “unsustainable

economically” for Brazil (Seifert, 2003, p. 8) where, according to Sergio Amadeu, the Head of Brazil’s National Information Technology Institute, “a mere 10 percent of the 170 million people have computers at home”(Brazil Gives Nod, 2003). For developing countries, open source is a viable option because of its inexpensive to nonexistent licensing fees where many of the average per capita income are less than the cost of a computer. Developing countries not only have limited IT budgets but are sidelined by the digital divide from technologically advanced developed countries. As Steven Weber mentions:

The digital divide between developed and developing countries is now a central feature of international politics and the global economy. The slogan captures a fundamental disparity in access to and the ability to use new technologies, a reflection of long-standing divides of poverty, education, and the freedom of choices (Weber, 2004, p. 248).

By engaging in the open source effort, developing countries can help direct the future of technology and can reduce dependence on economically developed nations. In the open source environment, software “is limited only by the knowledge and learning of the potential users, not by exclusionary property rights, prices, or the power of rich counties and corporations” (Weber, 2004, p. 250). International governments are recognizing the potential of open source in capturing innovation with worldwide collaboration.

U.S. Procurement

The current open source policy debate surrounds procurement, whether the U.S. should follow suit with many international governments and favor open source software. The United States government should not show a deliberate preference for open source software but should consider it on equal ground with commercial software. Some argue that open source is currently discouraged in procurement decisions, so legislation that requires the use of open source software is necessary; however, many U.S. governmental agencies already use open source software. The National Security Agency is working on a version of Linux called SELinux (Security Enhanced Linux) which develops software that improves computer security. The NAVY and the Defense Advanced Research Projects Agency (DARPA) are developing open source software for military purposes. In addition, the National Aeronautics and Space Administration (NASA), the Department of

Agriculture, the Federal Aviation Administration, Department of Defense, U.S. Air Force, and the Department of Energy make use of open source software. In debating whether the government should show preference for open source software in procurement decisions, one must examine the economics of the issue.

Economics

Lawrence Lessig, a Professor at Stanford, believes that the government should favor open source because “free access brings the cost of information down to its marginal cost, and neoclassical economics favors price at marginal costs” (Hahn, 2002, p. 59). He states that, “Between two systems for producing a public good, one that releases the information produced by that good freely and one that does not, all things being equal, public policy should favor free access” (Hahn, 2002, p. 59). Others argue that the government should preference open source because it will encourage open source network effects which “causes a good or service to have a value to a potential customer dependent on the number of customers already owning that good or using that service” (Network Effects, 2004). Commercial software especially Microsoft’s software is successful because it has positive network effects; its wide use on business and personal computers is a value to potential customers. Proponents of favoring open source believe the government should show preference to encourage similar positive network externalities for open source software. The argument is also made that the government should encourage open source software because the government should support a public good. Open source follows the definition of a public good because it is both nonrival and nonexcludible; one is not prevented from getting open source and one can use OSS without diminishing the supply for the next person.

Economist James Bessen argues that there should not be a preference for open source software because open source meets a specific need that is not met by commercial software and therefore extends the software market. The success of open source evinces that there was a market failure in the software industry that open source satisfies. Many economists argue that the marketplace is the best location for handling procurement concerns and that the laissez-faire procurement approach by the government allows for the successful development of both models. In the capitalist market, the economy driven by private interests is believed to best serve the common good. Microsoft’s Policy

Counsel, Bill Guidera says, “Market imperfections like procurement preferences are just troubling obstacles towards people freely choosing which software best meets their needs” (Council, 2004, p.21) Along with Microsoft, the Initiative for Software Choice and the Business Software Alliance promote neutral grounds for procurement decisions. Even IBM, who distributes open source software, does not agree with preferential procurement practices. The Director of Public Affairs at IBM, Tim Sheehy, promotes “regulatory frameworks that treat open source equitably, instead of legislative mandates” (DelBianco, 2004, p. 4).

Instead, individual offices should make procurement decisions based on economical cost-benefit analysis with the belief that IT managers can successfully determine what technologies and licenses work best within the confines of their offices. There is no quantitative data that shows the government should favor open source over commercial software, especially because it is difficult to find data on OSS due to its vast and informal structure, and restricting software options will only limit competition and choice. As stated in E-commerce & Development:

No single software can be unambiguously ‘better’ than all others. Like any tool, software has certain characteristics of usability, reliability, flexibility, robustness, and costs. There is no optimal balance between these characteristics, and much depends on the distinctive needs of the particular user” (2003, p. 102).

Although open source licenses are usually at zero cost because of the large volunteer programming base, many costs are involved in switching to the open source software. Completely switching to open source software will also threaten IT workers skilled in commercial software or will cost significant retraining dollars. Most open source developers and companies make the majority of money from post purchase costs: training, support, data integration, and maintenance. Procurement decisions must be made with criteria including the total cost of ownership along the software features and its ability to meet individual needs. A flexible software environment that supports both the open source and commercial software model will overall benefit the public with increased competition and options.

Coexist

Designing software is often difficult because of rapid changes in hardware, operating systems, and networks such that one is writing software for a future computing environment that is unknown. As mentioned earlier, Moore's Law which states that the capability of hardware microprocessors doubles every eighteen months often results in a software development pace that lags hardware development. Supporting the development of open source and proprietary software will help the development of software match the development pace of hardware.

A recent Department of Commerce survey concluded that software has enabled firms "to create extraordinary efficiencies and improve decision making within their own operations and supply networks" (Hahn, 2002, p. 77). Software impacts our economy and nation in such a critical way that showing deliberate preference would result in fewer options and would impair U.S. capabilities.

Many individuals and firms recognize the potential of both software models coexisting and are even combining the benefits of both models instead of merely choosing one option. Microsoft's Policy Counsel, Bill Guidera, states there is a "move to the middle" between both models of software development (Council, 2004, p.16). As a Congressional Report states, "The rapidly changing computer environment may also foster the use of a combination of open source and closed source applications, rather than creating a need to choose one option at the exclusion of another" (Seifert, 2003, p. 1). With ever-changing developments in information technology infrastructure, greater flexibility with both models is preferable. The U.S government should ensure that procurement decisions are based solely on cost benefit analysis by building awareness of open source; to date government actions have followed this belief.

Recent U.S. Governmental Action

Instead of preferring open source, the U.S. government is correctly improving the awareness of open source software and state governments are following suit. On July 1, 2004, the Executive Office of the President issued a memorandum to Senior Procurement Executives and Chief Information Officers titled OMB 04-16 describing open source and reminding staff to consider "differences in licensing may affect the use, the security, and the total cost of ownership of the software and must be considered when an agency is

planning a software acquisition” (Adelstein & Hiser, 2004). The Office of Management and Budget has also called for agencies to be “technology and vendor neutral” when making IT decisions (Jackson, 2004). In October 2002 the MITRE Corporation released a report on open source in the Department of Defense (DoD) concluding that open source “plays a more critical role in the DoD than has generally been recognized” (Seifert, 2003, p. 7). In October of 2000 the President’s Information Technology Advisory Committee (PITAC) recognized the impact of open source on high performance computing affirming, “Open source software developments are a promising means to enable high end computing and should be considered an important infrastructure investment by the federal government” (Developing Open Source Software, 2000).

State governments have similarly encouraged the consideration of open source in procurement decisions. Alabama State Bill 276 states to “expressly allow any state entity to use open source software in lieu of proprietary software whenever feasible” (Alabama Latest State, 2004). A 2004 preference bill introduced into the Virginia House of Delegations mentioned that nothing prohibits “the utilization of free, open-source software available on the Internet as an alternative to competitively bid contracts” (DelBianco, 2004). In addition to Alabama and Virginia, California, Oregon, Oklahoma, Texas, and Massachusetts have made an effort to improve the awareness of open source. However, the debate is not merely one about open source versus the commercial model, but rather a discussion about the implications of open source, its possible applications, as well as the impact of its model of production.

Open Source Implications in Government

Open source is being discussed not only as a software model but as a new post-capitalistic model of production that affects our knowledge economy along with our society. Open source represents a “real-world, researchable example of a community and a knowledge production process that has been fundamentally changed, or created in significant ways by Internet technology” (Weber, 2004, p.2). OSS can also provide insight on the political and economic implications of the growth of the Internet and how it has changed production, notions of intellectual property, and collaboration in the public domain (Weber, 2004, p. 2). As stated in the United Nations Conference on Trade and Development Report:

Internet or the rapid introduction into human affairs of extensive telecommunications bandwidth configured as a neutral and public network changes some very important things about the constraints and opportunities that individuals, organizations, and countries encounter as they move towards increasingly knowledge-intensive economies” (E-Commerce & Development, 2003, p. 119).

Not only has open source changed the process for information flow but it is also a different way of organizing a technocracy. The Information Age is altering or challenging our current power base and infrastructure such as the innovative capabilities of patents in protecting intellectual property. Economist Lawrence Lessig believes there is “an almost epochal battle over who will control what in the midst of a technological revolution and that open source is on the right side of the battle” (Weber, 2004, p. 9). Microsoft strains to compete with open source because there is not one central figure or organization to challenge; rather there is a vast and loosely structured support base.

The United States, like Microsoft, faces difficulties in competing with networks as problems arise in how to deal or defend oneself from a vast network of terrorist groups. Undoubtedly, much of the terrorist actions and planning have been abetted by the advent of the network and Internet. David Rondfeldt and John Arquilla in the foreign policy and security field note the rise of networks in foreign policy in what they term “netwar.” They believe “hierarchies have a difficult time fighting networks, it takes networks to fight networks, and whoever masters the network form first and best will gain major advantages” (Weber, 2004, p. 263). The open source model helps to elucidate the success and growth of networks which the government must understand to adequately provide national security.

Conclusion

Government policy should not restrict or show preference for either proprietary or open source software. One model is not inherently superior. For example, both models have security breaches and limitations. Therefore, procurement decisions by the government should be based on the specific needs of the as well as the total cost benefit analysis. Beyond procurement policy the United States government plays a critical role in funding software development and encouraging innovation through incentives such as

intellectual property rights. However, certain protections such as patents should be reconsidered as they limit software development. In addition, policymakers should consider the implications of funding research based on GPL licensed code as the intellectual property is distributed worldwide. This limits the ability of the United States to capitalize on its software knowledge creation.

Although open source is not the antagonist in the open source versus proprietary debate, it represents a growing shift towards a less hierarchical model of production centering on worldwide networks. Policymakers must understand the implications of this new production model as well as its effect on our government, commercial economy, and national security.

Works Cited

- Adelstein, T., & Hiser, S. (2004, July 23). Linux in Government: How to Misunderstand the Enterprise Linux Desktop. Linux Journal.
- Alabama Latest State to Present Open Source Software Bill. (2004, March 2). NewsForge. Retrieved August 20, 2004, from the World Wide Web: <http://business.newsforge.com/print.pl?sid=04/02/27/2329240>.
- The Bayh-Dole Act: A Guide to the Law and Implementing Regulations. (1999, October). Washington, DC: Council on Governmental Relations.
- Bickel, Bob. Vice President of Corporate Strategy and Development, JBoss. Interview by author, 11 August 2004.
- Brazil Gives Nod To Open Source. (2003, November 16). The Associated Press. Retrieved August 20, 2004, from the World Wide Web: <http://www.wired.com/news/infostructure/0,1377,61257,00.html>.
- Council on Competitiveness. (2004). Proceedings of the Forum on Technology and Innovation: The Policy Implications of Open Source Software. (2004). Washington, DC.
- DelBianco, S. (2004, April). No Such Thing As Free Software. Washington DC: American Legislative Exchange Council.
- Developing Open Source Software to Advance High End Computing. (2000, October). Washington, DC: the President's Information Technology Advisory Committee.
- E-commerce and Development Report 2003: Proceedings of the United Nations Conference on Trade and Development. (2003). New York: United Nations Publication.
- Fallon, D. (2004, July 15). Open Source Software: Report of the 15 July 2004 Forum on Technology & Innovation
- Hahn, R. (2002). Government Policy toward Open Source Software. Washington, DC: Brookings Institution Press
- Healey, K., & Schussman, A. (2003, January 14). The Ecology of Open Source Software. University of Arizona.

- House Subcommittee on Courts, the Internet, and Intellectual Property. (2002). Patent Reexamination and Small Business Innovation. Washington, DC: U.S. Government Printing Office.
- Jackson, J. (2004, July 16). Microsoft Applauds Open-Source Procurement Memo. Washington Technology.
- Mundie, C. Security: Source Access and the Software Ecosystem. The Microsoft Corporation. Retrieved August 20, 2004, from the World Wide Web: <http://www.net-security.org/dl/articles/mundie.pdf>.
- Lohr, S. (2004, July 5). Knowing Their Politics by the Software They Use. The New York Times.
- Network Effects. (2004). Wikipedia. Retrieved August 20, 2004, from the World Wide Web: http://en.wikipedia.org/wiki/Network_effects
- An open-source shot in the arm? (2004, June). The Economist.
- Open Source Software Institute: Open Source Software Position Briefing. (2003). Retrieved August 20, 2004 from the World Wide Web: www.oss-institute.org.
- Seifert, J. (2003, December 17). Computer Software and Open Source Issues: A Primer. Washington, DC: U.S. Congressional Research Service, the Library of Congress.
- Weber, S. (2004). The Success of Open Source. Massachusetts: Harvard University Press.
- Wheatley, Malcolm (2004, March 1). The Myths of Open Source. CIO Magazine.