DoD ESI White Paper

Considerations for Open Source Software Use

Contractual protections to consider before taking advantage of popular Open Source solutions

April 2013

www.esi.mil

About DoD ESI

The DoD ESI was formed in 1998 by Chief Information Officers at the DoD. To save time and money on commercial software, a joint team of experts was formed to consolidate requirements and negotiate with commercial software companies, resulting in a unified contracting and vendor management strategy across the entire department. Today, DoD ESI's mission extends across the entire commercial IT life-cycle to include IT hardware products and services. DoD ESI has established DoD-wide agreements for thousands of products and services.

www.esi.mil

Table of Contents

Introduction and Background

Free software and open source software are helping to bring back the collaborative model of code sharing in the commercial world.[1] In January 1998, a group associated with free software (software where the users have freedom to access the source code, not necessarily where the software is provided for no cost) introduced the term "open source." Chris Peterson and others in attendance at a strategy session in California did this to differentiate themselves from other free software projects, such as the GNU projects launched 15 years earlier to develop the GNU "free" software system. The intent was to create a new emphasis in the community for extending the free software model to the commercial world.[2] These new open source projects would exist in the mainstream of the commercial software market and include projects such as the Apache web server and the Mozilla browser.[3] (It is important to note that free software and open source software is distinguished from "freeware." Although freeware does not require payment for use, the authors or copyright holders of freeware may retain all rights to the software, and it is not necessarily permissible to reverse engineer, modify, or redistribute freeware.)[4]

The term "open source" commonly refers to a software program or set of software technologies that are made widely available by an individual or group in source code form for use, modification and redistribution under a license agreement having very few restrictions.[3] Source code is a computer program in its original programming language before translation into object code. Think of source code as the blueprint for a program that is generally written and stored in a textual, human-readable form, which must then be "compiled" into machine-readable form before it can be executed. The compiled code is known as object code. Merriam-Webster defines object code as "a computer program after translation from source code usually into machine language by a compiler." The DoD defines open source software as "software for which the human-readable source code is available for use, study, reuse, modification, enhancement, and redistribution by the users of that software.[5]

One of the fundamental differences between open source software and proprietary software is that the source code of open source software must be made available with the software. (This does not mean that the source code must be physically delivered with the software, just that is must be available at a freely accessible location.)[6] Anyone should be able to download the source code, view it, and alter it as they see fit. With proprietary software, you generally cannot view or edit the source code. It is also important to mention that when software is "open source" it does not necessarily mean that it is provided at no cost. The community or provider can still charge a fee for distributing the software and to ensure you receive updates and support to the source code.[7]

The History of Open Source Software

GNU's History

"GNU" is a recursive acronym for "GNU's Not Unix!", chosen because GNU's design is Unix-like, but differs from Unix by being free software and containing no Unix code. The GNU Project, announced by Richard Stallman in 1983, was an early effort in the free software movement that served as a precursor to open source software. The plan was to bring a wholly free Unix-like software operating system into existence.

Software development for GNU began in 1984 when Stallman quit his job at the Massachusetts Institute of Technology's Artificial Intelligence Laboratory so that they could not claim ownership or interfere with distributing GNU as free software.

In October 1985, Stallman set up the Free Software Foundation (FSF) to help raise funding. In the late 1980s and 1990s, the FSF hired software developers to write the software needed for the GNU Project. Nevertheless, most of GNU has been written by volunteers—some in their spare time and some paid by companies, educational institutions, and other non-profit organizations. A major gap in the GNU effort was filled in 1992, when Linus Torvalds released his Linux kernel.[8]

Overall, Stallman wanted software users to have freedom and for software programs to be "free" to users, as most were in the 1960s and 1970s. Today, a program is free software if its users have the "four essential freedoms:"(i) free to run the program for any purpose; (ii) free to study the source code and change it; (iii) free to share the unmodified software with other people; and (iv) free to distribute modified versions of the software.[9]

U.S. Government

The adoption of open source software is becoming more prevalent in the U.S. government. One example of this is the Navy. The Navy has been working with the Open Source Software Institute (OSSI) since 2001. The OSSI is a membership-based, non-profit organization comprising corporate, government, academic and open source development community representatives whose mission is to promote the development and implementation of open source software solutions within U.S. Federal, state and local government agencies.[10] In June of 2007, the U.S. Department of the Navy, Office of the Chief Information Officer (DONCIO) announced the issuing of the Navy's Open Source Software Guidance memorandum addressing the use of open source software within Navy and Marine Corps information technology (IT) systems. A thorough summary of the "History of open source in government" can be found in an article written by Gunnar Hellekson[11], which includes the following more recent historical highlights.

In October of 2009, DoD CIO David Wennergren, released a DoD memo titled, "Clarifying Guidance Regarding Open Source Software (OSS)." The memo followed the Navy's declaration two years earlier of reminding procurement officials that open source software is commercially available off-the-shelf (COTS). The appendices to the memo went into much more detail about the potential advantages and risks of open source software. The memo specifically encouraged the DoD to take advantage of its ability to modify software to suit a mission's need. Later in 2009, CENDI, an organization of government managers, issued a FAQ document addressing copyrights and open source software to help agency lawyers understand open source licenses and the intellectual property questions that they pose.[12]

In 2011, the Office of Management and Budget (OMB) refreshed its open source guidance with the "Technology Neutrality" memo[13], reminding agencies that competition in software is important, and that they are forbidden from discriminating against software based on its development method. Once this memo was published, most of the barriers to open source adoption had been diminished or eliminated in the US government. Open source continued its growth in government applications in 2011. Eben Moglen, a Professor of Law and Legal History at Columbia University Law School, and head of the Software Freedom Law Center, released "Government Computer Software Acquisition and the GNU General Public License[14]" which explains the provisions of that very popular open source license in the context of government procurement regulations.

Licensing Components

Copyright law grants the copyright-holder significant control over the copying and distributing of a work, but FSF wrote a license for GNU software, which grants recipients permission to copy and redistribute the software under highly permissive terms. For most of the 1980s, each unique GNU software project had its own GNU license: the Emacs General Public License (Emacs is a family of text editors), the GCC General Public License (a compiler system that supports various programming languages), etc. In 1989, FSF published a single license they could use for all their software, and which could be used by non-GNU projects: the General Public License (GPL). The GPL is one of the most commonly used open source licenses.[15] Since 1998, it has been updated twice, most recently to add protections against software patents.[16]

Not all open source licenses must comply with the GPL requirements. The Open Source Software Initiative, a California public benefit corporation founded in 1998, claims "open source" software (as compared to free source or free software) must comply with the following ten criteria:[17]

1) **Free Redistribution.** The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

   (Editor's Note: In this context, "free" refers to the "freedom" to redistribute. Companies or communities may charge support fees or fees for the value-added programs and services. Also, the GPL gives licensees broad rights to sell open source software, so long as licensees grant to downstream licensees the same rights to sell, copy and modify the modifications to the original program).[3]

2) **Source Code.** —The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost—preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

3) **Derived Works.** The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4) **Integrity of The Author's Source Code.** The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5) **No Discrimination Against Persons or Groups.** The license must not discriminate against any person or group of persons.

6) **No Discrimination Against Fields of Endeavor.** The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7) **Distribution of License.** The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8) **License Must Not Be Specific to a Product.** The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9) **License Must Not Restrict Other Software.** The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10) **License Must Be Technology-Neutral.** No provision of the license may be predicated on any individual technology or style of interface.

Why Use Open Source Software

In addition to easy access to source code and a broad community of developers some overall benefits of open source from the DoD's 2009 Memorandum include the following:[5]

Reliability

"The continuous and broad peer-review enabled by publicly available source code supports software reliability and security efforts through the identification and elimination of defects that might otherwise go unrecognized by a more limited core development team."[5] This includes bug fixing, since when a bug is spotted in proprietary software, the only people who can fix it are the original developers, as only they have access to the source code. Instead, open source software normally has a large number of users who can access and change the code, so bugs tend to be more rapidly corrected.[18]

Unrestricted Users

"Open source licenses do not restrict who can use the software or the fields of endeavor in which the software can be used. Therefore, open source software provides a net-centric licensing model that enables rapid provisioning of both known and unanticipated users."[5]

Rapid Modification

"The unrestricted ability to modify software source code enables the Department to respond more rapidly to changing situations, missions, and future threats…open source software is particularly suitable for rapid prototyping and experimentation, where the ability to 'test drive' the software with minimal costs and administrative delays can be important."[5] If the open source software you are using adheres to the GPL, you are not required to release your modified version, or any part of it. You are free to make modifications and use them privately, without ever distributing them.[19]

No Distribution Requirements for Private Use

A white paper co-authored in 2011 by members of the Software Freedom Law Center and the Navy titled, "Government Computer Software Acquisition and the GNU General Public License," states "the GPL's source code distribution terms are only applicable when the modified executable is distributed to the public or "outside" the government. Distribution of GPL software source code made "inside" the government within classified programs is a private modification and the resulting executables and source code may only be redistributed to individuals and contractors with the appropriate access."[14] Therefore, modification to GPL source code solely distributed "inside" the government does not need to be redistributed to the public.

No Supplier Lock-In

"Reliance on a particular software developer or supplier due to proprietary restrictions may be reduced by the use of open source software, which can be operated and maintained by multiple vendors, thus reducing barriers to entry and exit."[5] Additionally, by relying upon a community rather than one proprietary software supplier, the risk of supplier financial instability and discontinuation of support and future enhancements is reduced.

Cost

Many open source programs can be obtained at no cost or at a very low cost. For example, in November 1998 the Mexican government decided to use Linux in 140,000 primary and secondary schools. By choosing a lower cost alternative (Linux) instead of Microsoft Windows products (Windows 98 and Windows NT), the government projected a savings of 124 million USD in license costs.[20] Since open source software typically does not have a per-seat licensing cost, it can provide a cost advantage in situations where many copies of the software may be required, and can mitigate risk of cost growth due to licensing in situations where the total number of users may not be known in advance. By sharing the responsibility for maintenance of open source software, the Department as a whole can benefit by reducing the total cost of ownership for software, particularly compared with software for which the Department has sole responsibility for maintenance (e.g. Government Off-the-Shelf).[5]

Who is Using Open Source Software

A total of 80 percent of various entities, including the government, use open source software.[21] By 2016, it is estimated that 99 percent of Global 2000 companies will use open source in their mission-critical applications.[22] In 2005, 10 percent of enterprises' IT portfolio was made up of open source and it is estimated by the end of 2012, that figure will have increased to 30 percent.[22] The breakdown of involvement with open source software is as follows:

- 34 percent only consume
- 45 percent consume and contribute directly back to projects
- 12 percent consume and contribute back via a third party
- 9 percent contribute to open source projects even though their company's policies prohibit it

Linux, Apache web server, and MySQL were the top open source projects in 2011.[23] Some of these and other popular open source examples are listed here below:

A) **Alfresco.** Alfresco is an enterprise content management system created in 2005 by John Newton, co-founder of Documentum, and John Powell, former COO of Business Objects. Alfresco's open source free license is licensed under the Lesser GPL, which provides for less freedom for end-users and lessens the software provider's requirements to share what is built on top of GPL-covered software.[24] Alfresco also offers a commercial version of their free open source software that is identical, but the commercial version is provided for under a support subscription which provides users access to: (i) Alfresco enterprise software, including certified stress-tested builds; (ii) technical support services and escalations, including an end user portal; and (iii) maintenance fixes, patches, and updates.[25]

B) **Tomcat** is an Apache Software Foundation open source web server and servlet container project that powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations.[26] Tomcat started off as a servlet reference implementation by James Duncan Davidson, a software architect at Sun Microsystems. He later helped make the project open source and played a key role in its donation by Sun to the Apache Software Foundation.[27] The Apache Software Foundation was incorporated in 1999 as a membership-based, not-for-profit corporation in order to ensure that the Apache projects continue to exist beyond the participation of individual volunteers.

C) **Mozilla Firefox** is a web browser that originated from the 1998 Mozilla project that released the Netscape browser suite source code. Within the first year, new community members from around the world had already contributed new functionality, enhanced existing features and became engaged in the management and planning of the project itself. After several years of development, Mozilla v.1.0 was released in 2002. In 2002, the first version of Phoenix (later renamed to Firefox) was also released by Mozilla community members. In 2003, the Mozilla project created the Mozilla Foundation, an independent non-profit organization supported by individual donors and a variety of companies. Firefox 1.0 was released in 2004 and became a big success. By 2008, Firefox had 20 percent worldwide market share.[28]

D) **Red Hat Enterprise Linux** is an operating system for open source computing. It is sold by annual subscription, runs on seven system architectures, and is certified by top enterprise software and hardware vendors.[29]

E) **OpenOffice** is another Apache project that provides an office application suite. OpenOffice has been in existence since 2000, and was first released in 2002.[30] Contributors to this open source project include Oracle, RedFlag CH2000, and IBM. Additionally, more than 500,000 people—unaffiliated individuals, business and government employees, and students—are part of the project.

F) **Subversion** Founded in 2000 by CollabNet, Inc., Subversion is an open source version control system. Subversion has now also become a project of the Apache Software Foundation. Although CollabNet (and other companies) pay the salaries of some full-time Subversion developers, the software carries an Apache License.[31] In 2011, Carahsoft Technology Corp. and CollabNet, were awarded a five-year DoD Enterprise Software Initiative (ESI) Blanket Purchase Agreement (BPA) for CollabNet software and consulting services, including the CollabNet TeamForge application lifecycle management platform, CollabNet Lab Management, training, and consulting services. CollabNet TeamForge powers the DoD's Forge.mil initiative, which is focused on improving the ability of the DoD to rapidly deliver dependable software, services and systems in support of net-centric operations and warfare. Forge.mil consists of two offerings – SoftwareForge and ProjectForge. SoftwareForge provides a free collaborative environment for shared development of DoD community source and internal open source software. ProjectForge provides teams that require greater access control with their own secure, private, web-based collaborative software development environment. ProjectForge is a fee-for-service capability delivered on-demand by DISA Computing Services and hosted in a Defense Enterprise Computing Center.[32]

Government Use Considerations

Obtaining Open Source Software

There are three major ways that open source software can be obtained and used:

1) **As a component** of a software developer's toolkit, where the open source software may be used internally as one tool within a developer's toolbox to create an overall solution;

2) **As a standalone product** that is marketed as open source software (e.g. Alfresco is an "open source content management platform"); or

3) **As part of a commercial off the shelf (COTS) solution** that is not marketed as containing or using open source software (or other potential third party components), but has open source software code embedded in its overall solution. We will come back to this use type later in the article, as this may pose the most risk to an entity regarding whether or not the terms of open source licenses are being violated.

Most open source licenses can be downloaded directly from websites. Open source software is just like any other software—the license defines the rights, restrictions and obligations. Open source software licenses generally grant broader rights than traditional software licenses.[33] In the case of open source software, that means that you are obtaining the software either from a reputable company that includes an indemnity for the code they distribute (OpenGeo is one such example) or software that is distributed with the support of an organization (such as Apache.org or OSGeo.org) that requires contributors to sign agreements ensuring only unencumbered code is contributed, meaning that such code does not have proprietary or copyrighted materials incorporated without the permission to do so.[34]

Ownership of Code Base and Enhancements

Ensure that the open source license permits your entity to modify open source software for internal use without being obligated to distribute source code to the public. It is important to note that if the end-user chooses to distribute the modified open source software outside the user's organization (e.g. a Government user distributes the code outside the Government), then some open source software licenses (such as the GNU General Public License) do require distribution of the corresponding source code to the recipient of the software. For this reason, it is important to understand both the specifics of the open source license in question and how the Department intends to use and redistribute any DoD-modified open source software before choosing to use open source software.[5]

Supporting Open Source Software

Before approving the use of software (including open source software), system/program managers, and ultimately Designated Approving Authorities (DAAs), must ensure that the plan for software support (e.g. commercial or Government program office support) is adequate for mission need.[5] While community or internal support is still significant, there's an increasing interest in third-party support.[23] The government can also consider creating a government user community within the open source communities by which they can support the specific open source software needs of the government. This community-within-a-community approach can help supplement the use of third parties' support for open source software. Leverage new government community portals such as Forge.mil, which enables users to share knowledge, experience, and lessons learned on how to improve and accelerate software development and deployment via group blogs, discussions, wiki, documents and polls.[35]

Mitigating Tactics for Use of Open Source Software

A) **Open Source Policy.** Almost half of entities surveyed have an open source policy or pre-approval process in place.[21] It is important to ensure this policy is reviewed and updated on a regular basis, and that internal audits are performed to ensure the policies are being followed. For example, a survey conducted found that 71 percent of 635 leading mobile applications using open source software licenses under GPL, Apache and Lesser GPL did not comply with the applicable open source licenses.[36]

B) **Code Scanning Tools.** Open source scanning, the scanning of compiled binaries and source code for adherence to open source license requirements, is compliance management. The adoption of scanning tools boils down to how much a company is willing to tolerate, with regard to the risk of violating the open source license under which the software is distributed if they do not have scanning tools in place.[22] Entities should consider investing in secure code scanning tools to help eliminate security vulnerabilities in their software code. Additionally, there are open source scanning products and services that help entities understand how much open source they use, how and where it's used, and what needs to be done to comply with open source licenses.[37]

C) **Intellectual Property Indemnification and Disclosure.** Some companies, like Red Hat, offer programs to protect customers developing and deploying open source solutions. Red Hat's Open Source Assurance program features an Intellectual Property Warranty for Red Hat users. The warranty is a promise by Red Hat to replace software if there is an intellectual property issue so that customers may continue to use the solution without interruption.[38] However, use of improper code in open source projects is small. In the case where this does occur, legal mitigation is always clear: open source projects that have mistakenly embedded encumbered code simply remove it when notified, the software is automatically updated, and the end users move on to the unencumbered version.[34]

Additionally, it would be prudent to demand that COTS software providers disclose any open source software used (as well as potential other third-party components) as part of their overall software solution, and to provide warranties and indemnification with their software. Such indemnification for the open source software contained in their COTS solution may be limited to the modifications, if any, that the COTS software provider made to the source code. Additionally, the COTS provider should warranty that, since their overall software solution is being made public, that any and all open source software must be accompanied by disclosure of the source code related to the applicable components so as to not infringe upon any open source licenses. Finally, ensure that there will be no additional future fees requested by the COTS software provider related to their use and support of the open source code.

D) **Lawsuit Examples.** The American IP Law Association estimates that defense against a single software patent lawsuit costs between $2 and $5 million dollars.[39] According to the End Software Patents coalition, $11.4 billion is wasted each year on software patent litigation. This $11.4 million figure was calculated from estimations of 55 software patent suits being filed every week and the cost to litigate a single mid-sized patent suit costs $4 million dollars.[40] Here are some examples of lawsuits regarding free software or open source software.

1) **FireStar Software vs. Red Hat.** In June 2008, Red Hat paid up $4.2 million to settle a patent infringement suit that had been brought by FireStar Software.[41] There were warning signs in 2004 when the Open Source Risk Management (OSRM) study announced that it had found 283 software patents issues in the Linux kernel.[42] That is just one program out of the thousands that make up a Linux distribution.[40]

2) **FSF vs. Cisco.** In December 2008, the FSF filed a copyright infringement lawsuit against Cisco, the designer and seller of electronics, as well as networking and communications technology and services. Cisco allegedly was distributing Linksys products, which violated the licenses of programs like GCC, binutils, and the GNU C Library. The FSF holds copyrights for these programs, and as such they should have been made open to users to share or modify the software, and by denying users access to the source codes, Cisco was brought to court. Five months later the two parties settled when Cisco agreed to appoint someone to ensure that all Linksys products are compliant with the free software licenses and to make a donation to the FSF.[43]

3) **Oracle vs. Google.** When Sun Microsystems released its Java technology and made its programming language open to all in 2006, it left most of it free. However, J2ME, the mobile version of Java, was under license whereby any commercial closed source development of it was under obligation of payment. Meanwhile, Google was using J2ME to develop their Android operating system. They never paid Sun Microsystems for use of J2ME. Oracle was very aware of this when they agreed to buy Sun Microsystems in 2009.[44] Oracle, after discussing these issues with Google, filed suit in 2010 against Google for copyright and patent infringement.[44] The trial started in April 2012 and although in some cases Google had infringed on Oracle's copyrights in building Android, Google is not required to pay any money for this infringement.[45] Additionally, Oracle had to pay Google more than $300,000 in legal fees.[46]

One strategy by commercial copyright holders and companies to help avoid future litigation is to cross license their patents to each other. These agreements allow each party to develop against the other party's patented software without the threat of being sued for patent infringement. The ability to negotiate cross-licensing agreements is a major reason that many software companies, including ones providing open source software, file patents. For example, Microsoft formed a strategy back in 2004 to start cross licensing. Microsoft believed that, "if you get agreements…with the big 30 to 40, you can provide a tremendous amount of development freedom to yourself going forward. Probably about 80 percent of the relevant patents out there are really held by those 30 to 40 companies."[47] Microsoft has cross-licensed with companies such as Amazon, Casio, Lexmark, Panasonic, and Samsung.

Recent Open Source Software Experiences within the U.S. Government

As stated above, use of open source software should be approached by addressing key considerations before using it. The government is focusing on developing more guidance and protections around the use of open source software. For instance, the Homeland Open Security Technology (HOST) program was created in order to investigate open security methods, models and technologies and to identify viable and sustainable approaches that support national cyber security objectives. The foundational technology for the purposes of HOST is based on open source software.

One government agency, the Consumer Financial Protection Bureau, recently had the advantage of being able to build its systems from scratch. As such, it took advantage of that flexibility to build open systems on open source software that could grow with the agency over time. Matthew Burton, the CIO at the Consumer Financial Protection Bureau said the goal of establishing the agency IT infrastructure was not only to use open source, but also to share code with the public (whenever it didn't compromise system security).[48]

In 2011, extensive interviews were conducted with a wide range of state, local and federal government information technology professionals, industry experts and others to gain a fuller understanding of how open source is being used in U.S. government today and where the opportunities and challenges lay. The formal report is in review and anticipated to be released by the Department of Homeland Security. The report is expected to be titled, "Lessons Learned: Roadblocks and Opportunities for Open Source Software (OSS) in U.S. Government."[49]

In order to help obtain further guidance surrounding the use of open source software, the DoD held a public meeting in January of 2012 regarding the Defense Federal Acquisition Regulation Supplement (DFARS) to gain insight on the following areas:[50]

- What are the risks that open source software may include proprietary or copyrighted material incorporated into the open source software without the authorization of the actual author, thereby exposing the Government and contractors who use or deliver the open source software to potential copyright infringement liability?
- Are contractors facing performance and warranty deficiencies to the extent that the open source software does not meet contract requirements, and the open source software license leaves the contractors without recourse?
- To what extent should the DFARS be revised to specify clearly the rights the Government obtains when a contractor acquires open source software for the Government, and why?

Some of the feedback obtained during this public meeting was from Eddie Pickle of OpenGeo who stated, "On the question of the risks of copyright liability, open source does not differ from proprietary software. Open source is no more likely to have proprietary or copyrighted materials incorporated than proprietary software. Therefore, the guidance for selecting open source solutions is quite similar to what it would be for prudent procurement of proprietary software: make sure you know your source."[34]

Conclusion

Remember, just like any software solution, open source software requires enterprises to apply policies, approvals, and controls around its deployment. Although typical use of open source software can have a lower total cost of ownership, poor management of open source use can also bring out hidden costs related to license compliance issues and software vulnerabilities.[22] The key to successful use of open source software and keeping costs low is to ensure there is an effective and enforceable open source software policy in place. Create a return on investment framework for open source software adoption and use it in the procurement process.[36] In addition to having strong governance policy for open source software, ensure use of scanning tools to maintain insight into the inventory of open source software. Also, consider creating a government user community for open source software to foster self-support for the specific needs of the government.

About the Author

Gretchen Kwashnik currently serves as the IT Supplier Management Team Leader at ING DIRECT, USA in Wilmington, DE. Her responsibilities include technology acquisition, IT contract management, IT supplier management, and the integration and management of IT project contractors. Gretchen began her ING DIRECT career in the Procurement department, drafting and negotiating IT and Marketing contracts. In2008, she transitioned to the IT department, where she has since concentrated on technology acquisition, the software development life cycle, and the facilitation of IT projects. Gretchen has a Bachelor of Science in International Business from King's College and a Juris Doctorate from Widener University Law School.

References

[1] Article, "A New Open-Source Model For SaaS" by Dan Woods on November 9, 2010. Published by Forbes. Found at: http://www.forbes.com/sites/ciocentral/2010/11/09/a-new-open-source-model-for-saas/(Last accessed August 20, 2012)

[2] Website: http://www.opensource.org/history/ (Last accessed August 20, 2012)

[3] Report, "An Overview of "Open Source" Software Licenses" by The Software Licensing Committee of the American Bar Association's Intellectual Property Section. Found at: http://www.americanbar.org/groups/intellectual_property_law/resources/an_overview_of_open_source_software_licenses.html (Last accessed August 20, 2012)

[4] "Legal battles that shaped the computer industry" p.175 by Lawrence D. Graham in 1999 by Greenwood Publishing Group.

[5] DoD Chief Information Officer (CIO) Memorandum, "Clarifying Guidance Regarding Open Source Software (OSS)," October 16, 2009. Found at: http://dodcio.defense.gov/Home/Topics/UseofFreeOpenSourceSoftwareFOSS.aspx (Last accessed August 20, 2012)

[6] Website: http://oss-broschuere.berlios.de/broschuere/broschuere-en.html (Last accessed August 20, 2012)

[7] Website: http://www.gnu.org/philosophy/selling.html (Last accessed August 20, 2012)

[8] Website: http://www.gnu.org/gnu/linux-and-gnu.html (Last accessed August 20, 2012)

[9] Website: http://www.gnu.org/philosophy/free-sw.html (Last accessed August 20, 2012)

[10] Website: www.oss-institute.org (Last accessed August 20, 2012)

[11] Article, "History of Open Source in the Government" by Gunnar Hellekson on May 2, 2012. Published by opensource.com. Found at: http://opensource.com/government/12/5/history-open-source-government (Last accessed August 20, 2012)

[12] Paper, "Frequently Asked Questions about Copyright and Computer Software: Frequently Asked Questions about Copyright and Computer Software" on October 1, 2010. Found at: http://www.cendi.gov/publications/09-1FAQ_OpenSourceSoftware_FINAL_110109.pdf (Last accessed August 20, 2012)

[13] Memorandum "Technology Neutrality" by Office of Management and Budget on January 7, 2011. Found at: http://www.cio.gov/documents/Technology-Neutrality.pdf (Last accessed August 20, 2012)

[14] Paper, "Government Computer Software Acquisition and the GNU General Public License" on October 1, 2011. Found at: http://www.softwarefreedom.org/resources/2011/government-acquisition-and-gpl.pdf (Last accessed August 20, 2012)

[15] Website: http://www.gnu.org/licenses/gpl.html (Last accessed August 20, 2012)

[16] Article, "Is GPL licensing in decline?" by Simon Phipps on April 27, 2012. Published by InfoWorld. Found at: http://www.infoworld.com/d/open-source-software/gpl-licensing-in-decline-191816 (Last accessed August 20, 2012)

[17] Website: http://www.opensource.org/docs/osd (Last accessed August 20, 2012)

[18] Website: http://www.oss-watch.ac.uk/resources/whoneedssource.xml (Last accessed August 20, 2012)

[19] Website: http://www.gnu.org/licenses/gpl-faq.html (Last accessed August 20, 2012)

[20] Article, "Mexican Schools Embrace Linux" by Leander Kahney on November 28, 1998. Published by WIRED. Found at: http://www.wired.com/news/technology/0,1282,16107,00.html (Last accessed August 20, 2012)

[21] Sonatype's 2012 Open Source Software Development Survey. Found at: http://www.sonatype.com/people/wp-content/uploads/2012/03/2012-sonatype-survey-findings-PDF.pdf (Last accessed August 20, 2012)

[22] Article, "Scan open source use to minimize risks, optimize benefits" by Jamie Yap, ZDNet Asia on March 20, 2012. Found at: http://www.zdnetasia.com/scan-open-source-use-to-minimize-risks-optimize-benefits-62304238.htm (Last accessed August 20, 2012)

[23] Presentation, "Predictions & Trends for Open Source in the Enterprise in 2012" by Steven Grandchamp of OpenLogic at the 2012 POSSCON Conference. Found at http://www.posscon.org/presentation/predictions-trends-open-source-enterprise-2012/ (Last accessed August 20, 2012)

[24] Website: http://www.gnu.org/licenses/gpl-faq.html#WhySomeGPLAndNotLGPL (Last accessed August 20, 2012)

[25] Website: http://wiki.alfresco.com/wiki/Open_Source_Licensing (Last accessed August 20, 2012)

[26] Website: http://tomcat.apache.org (Last accessed August 20, 2012)

[27] Website: http://en.wikipedia.org/wiki/Apache_Tomcat (Last accessed August 20, 2012)

[28] Website: http://www.mozilla.org/en-US/projects/ (Last accessed August 20, 2012)

[29] Website: http://www.esi.mil/contentview.aspx?id=200&type=2 (Last accessed August 20, 2012)

[30] Website: http://www.openoffice.org/about/ (Last accessed August 20, 2012)

[31] Website: http://subversion.apache.org (Last accessed August 20, 2012)

[32] Press Release, "Carahsoft Awarded $29M U.S. DoD Blanket Purchase Agreement for CollabNet Software and Services" in 2011. Found at: http://www.open.collab.net/news/press/2011/carahsoft.html (Last accessed August 20, 2012)

[33] Article, "Open Source Licenses: Greater Rights, Different Responsibilities" by Jennifer Venable on April 5, 2011. Published by LinuxInsider. Found at: http://www.linuxinsider.com/story/72274.html (Last accessed August 20, 2012)

[34] Article, "Defense Federal Acquisition Regulation Supplement; Open Source Software Public Meeting: Remarks Delivered by Eddie Pickle of OpenGeo" on January 12, 2012. Found at: http://www.acq.osd.mil/dpap/dars/docs/OpenGeoDFARSmeetingstatement-Jan2012.pdf (Last accessed August 20, 2012)

[35] Website: http://www.disa.mil/Services/Enterprise-Services/Applications/Forge-Mil (Last accessed August 20, 2012)

[36] Posting "OpenLogic Scan Shows Open Source License Violations for iPhone and Android: More Than 70%+ Of Mobile Applications Containing Open Source Fail to Comply, posted by Aaron Mandelbaum on March 8, 2011. Found at: http://www.openlogic.com/news/bid/154650/OpenLogic-Scan-Shows-Open-Source-License-Violations-for-iPhone-and-Android-More-Than-70-Of-Mobile-Applications-Containing-Open-Source-Fail-to-Comply (Last accessed August 20, 2012)

[37] Website: http://www.openlogic.com/solutions/open-source-scanning.php (Last accessed August 20, 2012)

[38] Website: http://www.redhat.com/software/assurance/faq.html (Last accessed August 20, 2012)

[39] Article, "The Monster Arrives: Software Patent Lawsuits Against Open Source Developers" by Bruce Perens on June 30, 2006. Found at: http://technocrat.net/d/2006/6/30/5032/ (Last accessed August 20, 2012)

[40] Article, "$11.4 billion wasted on software patent litigation...and counting," by Matt Asay on February 28, 2008. Found at: http://news.cnet.com/8301-13505_3-9882152-16.html?tag=mncol;txt (Last accessed August 20, 2012)

[41] Article, "Red Hat feeds the patent trolls and fools the FOSS community" by FOSS Patents on March 14, 2011. Found at: http://www.fosspatents.com/2011/03/red-hat-feeds-patent-trolls-and-fools.html (Last accessed August 20, 2012)

[42] Article, "Linux's Patent Risk" by Sean Michael Kerner on August 2, 2004. Published by InternetNews.com. Found at: http://www.internetnews.com/dev-news/article.php/3389071(Last accessed August 20, 2012)

[43] Article, "5 Pivotal Open Source Lawsuits" on December 15, 2010. Found at: http://blog.sherweb.com/5-pivotal-open-source-lawsuits (Last accessed August 20, 2012)

[44] Article, "A Timeline of Oracle's Java Lawsuit Against Google" by James Niccolai on April 12, 2012. Published by PCworld. Found at: http://www.pcworld.com/businesscenter/article/253666/a_timeline_of_oracles_java_lawsuit_against_google.html (Last accessed August 20, 2012)

[45] Article, "Google Wins Patent Verdict In Oracle Trial" by John Letzing updated on May 23, 2012. Published by the Wall Street Journal. Found at: http://online.wsj.com/article/BT-CO-20120523-718152.html (Last accessed August 20, 2012)

[46] Article, "OUCH! Oracle Has Already Paid Google More Than It Can Win In Android Lawsuit" by Owen Thomas on June 4, 2012. Published by the Business Insider. Found at: http://www.businessinsider.com/oracle-google-android-lawsuit-legal-fees-2012-6 (Last accessed August 20, 2012)

[47] Article, "Microsoft, Autodesk in patent licensing deal" by Joris Evers on December 17, 2004. Published by IDG Communications. Found at: http://www.arnnet.com.au/article/5043/microsoft_autodesk_patent_licensing_deal/?taxid=620938001#closeme (Last accessed August 20, 2012)

[48] Article, "Government Agency Built From Ground Up with Open Source," written by Ron Miller on Apr 10, 2012. Found at http://www.real-user-monitoring.com/government-agency-built-from-ground-up-with-open-source (Last accessed August 20, 2012)

[49] Website: http://www.cyber.st.dhs.gov/host/ (Last accessed August 20, 2012)

[50] Notice, "Defense Federal Acquisition Regulation Supplement; Open Source Software Public Meeting" in the Federal Register Vol. 76, No. 233 on December 5, 2011. Found at: http://www.gpo.gov/fdsys/pkg/FR-2011-12-05/pdf/2011-31111.pdf (Last accessed August 20, 2012)

DoD ESI is an official
Department of Defense initiative
sponsored by the Department of Defense
Chief Information Officer (DoD CIO).

Your Preferred Source for
IT Acquisition Across the DoD

BEST VALUE

EFFICIENT

LOW RISK

VOLUME DISCOUNTS

UNIFIED VOICE

Visit DoD ESI online at **www.esi.mil**

Department of Defense Chief Information Officer
6000 Pentagon
Washington, DC 20350-6000