The Open-Source Movement: An Introduction for Forestry Professionals

Patrick Proctor, Paul C. Van Deusen[1], Linda S. Heath, and Jeffrey H. Gove[2]

Abstract.—In recent years, the open-source movement has yielded a generous and powerful suite of software and utilities that rivals those developed by many commercial software companies. Open-source programs are available for many scientific needs: operating systems, databases, statistical analysis, Geographic Information System applications, and object-oriented programming. Using "real world" examples, including applications employed by Federal agencies, we address the concerns associated with open-source software deployment: cost, security, software availability, and usability. The potential for application to U.S. Department of Agriculture Forest Service Forest Inventory and Analysis data is discussed.

The growing availability of open-source software is causing many businesses and organizations to consider its adoption. Open-source software has advanced to the point where it has become a viable alternative. "Open source" does not just mean free software that is distributed with its source code. For software to be considered open source, it must comply with 10 criteria of the Open Source Definition (Open Source Initiative 2004). The Open Source Initiative, a registered nonprofit organization, broadly oversees the certification of software distributed under a license that conforms to the Open Source Definition. This article will explore the nature of open source, compare it with similar proprietary corporate platforms, and address many of the concerns voiced by today's information technology (IT) user. We believe the merits of open source allow for a formidable and attractive platform.

The Open-Source Philosophy

Raymond (2000) describes the major differences in the development paradigms between closed- and open-source software. He compares the former to the building of a cathedral, where the design, progress, and management of a software project are conducted under strict regiment in a group that is closed to nonmembers. Such models normally apply to corporate projects, although in the past they have also been applied to open-source software projects. By contrast, the development of the popular GNU Emacs editor (Free Software Foundation 2003a) exemplifies the open-source approach. This latter model is compared to a bazaar, which seems at first appearance to be chaotic and uncontrolled, but when the model is viewed with scrutiny, it more closely resembles the working of a diverse yet controlled system. Linus Torvalds was the first to popularize this open-source model with his "release early and often, delegate everything you can" (Raymond 2000, 2) philosophy. Torvalds is the creator of Linux, currently the most accessible and widely used open-source operating system. In this developmental model, users are often also contributors. One of the major keys to success of such ventures is that people contribute not because they were assigned to but out of love for the project.

Central to the open-source model and considered the core difference between the cathedral and bazaar models is Linus's Law: "Given enough eyeballs, the bugs are shallow" (Raymond 2000, 9). In the cathedral model, bugs are insidious and often difficult to correct, if found at all, because of the limited number (and often high turnover) of programmers with access to the code. The bazaar model, however, draws on the talents of often thousands of "hackers"; with such a base to draw from, an insidious bug becomes something simply fixed not by the group as a whole but by the one or two people out of the many with the specific talent. "Release often," then, becomes the vehicle for rapid development and evolution toward an unbreakable system.

[1] Programmer and Senior Scientist, respectively, NCASI SMDG Group, 600 Suffolk St, Fifth Floor, Lowell, MA 01854. Phone: 978-934-1948; email: paul_vandeusen@uml.edu.
[2] Supervisory Research Forester and Research Forester, respectively, U.S. Department of Agriculture, Forest Service, PO Box 640, Durham, NH 03824. Phone: 603–868–7612; e-mail: lheath@fs.fed.us.

The best example of the bazaar model, as Raymond (2000) points out, is Linux itself. The Linux platform is available in a number of "distributions" made by various software groups or companies that include Red Hat, Debian, and Yellow Dog. Although groups may package and sell the code, the source code is free and available to be compiled, and contributions are considered and encouraged from all. The Linux kernel—which is stable and often termed unbreakable—can scale from embedded devices (Embedded Linux Consortium 2003) to clusters running at supercomputer speed, including the world's third fastest supercomputer as of June 2003 (TOP500 2003). Finally, as evidenced by the visionary GNU Project (Free Software Foundation 2003b) and the thousands of tools produced directly by members of the Free Software Foundation or under the GNU General Public License (Free Software Foundation 2003c), a large community of users have based their work on a Linux platform.

Security

As expected, security is a primary concerns when switching to an open-source platform. The security measures available in open-source operating systems are comparable to those available in proprietary, closed-source operating systems such as Microsoft Windows (Microsoft 2004). Remote access to machines is controlled by a series of "ports," each of which is assigned to a particular function (e.g., HTTP, FTP, Telnet). Access to these ports, in turn, is controlled by a firewall that blocks outside users and illegal ports. This user control (available in Linux distributions and in Microsoft Windows versions 2000 and XP only) is accomplished through a user name/password-based access system, which requires users to be verified by a system administrator before gaining access.

Because Microsoft Windows is the most used desktop operating system in the world, its exploitation by hackers is more likely for a number of reasons. First, more users in the form of individual desktop systems exist to "attack," which makes an attractive target. Also, viruses and worms can spread more rapidly because of the large user base. Second, on Microsoft Windows systems, software, such as web browsers, are allowed to run scripts that, if the author is clever enough, can directly access the operating system files—something that is not allowed on open-source Linux. Third, patches must go through a corporate testing and clearance process before being released to the public. This results in a long lag time until a resulting virus "cure" is built into the system itself. Typically, a patch, when finally released, is available exclusively through Microsoft servers. The code cannot be checked by outside sources because of its unavailability to the general public, and the reliability of the patch is based entirely on internal Microsoft control mechanisms. Although Microsoft has a full staff of software testers and security analysts, hackers consistently exploit Windows system vulnerabilities before these "holes" are discovered internally. Some recent examples include the Blaster and SoBig viruses (Cable News Network 2003). Unfortunately, Microsoft provides no means for users to assist in solving this problem other than to be aware of and follow Microsoft advisories. If Windows users want to address these security concerns, they often are required to look to third-party providers.

As mentioned above, the open-source community has a far less restrictive management system for vulnerabilities. Bugs are often discovered and patched by any of the numerous users involved in open-source development. Before code is put into practice, the code is checked and rechecked by a literally world-wide network of developers. Patches are quickly and freely distributed to anyone who wants them. Because the patches are open source, they can be hosted on any server, provided the server abides by the GNU General Public License (Free Software Foundation 2003c). The open-source community is always searching for new vulnerabilities, and community groups, such as the Linux Security Audit Project, exist for the sole purpose is finding and patching Linux vulnerabilities (Linux Security Audit Project 2003). In addition, efforts such as the National Security Agency's Security-enhanced Linux project (National Security Agency 2003) provide even more protection if desired (Coker 2003).

Although debates occur about which distribution and patching system is more efficient or desirable, open-source solutions are in no way less secure than their proprietary counterparts. They clearly offer a well-documented and tested security alternative to proprietary operating systems.

Cost

Cost often is cited as a significant factor in the success of the open-source movement. Although prices of retail software continue to rise, open-source software remains entirely free or affordable to license and install. As table 1 shows, a number of retail closed-source packages have open-source counterparts, and the savings in using them can be immense (Newegg 2003). Although a cost advantage clearly exists to using open-source products, the argument can be made that the savings in retail cost is eclipsed by the time cost of retraining employees on new and/or unfamiliar applications.

Five to 10 years ago, when Linux was largely text-based, training users may have been costly. The Linux user interface has been redefined to be accessible to any user, however. A number of graphical interfaces are available to choose from, e.g., GNOME (GNOME Foundation 2003) and KDE (KDE e. V. 2003), all of which draw on industry-standard interfaces as their inspiration. Any user familiar with the Microsoft Windows operating system's graphical user interface could switch to the current Linux environment and find similar functionality. The same holds true for vital applications such as office productivity and photo-editing programs. Linux user interfaces will be familiar to Microsoft Windows users, and they also feature extensive online help. Also, databases based on the structured query language (SQL) must adhere to the SQL standard. Queries and databases written for retail programs, such as those from Oracle, can easily be migrated to the popular open-source database MySQL (MySQL AB 2003, Oracle 2004). Developers familiar with Oracle database products will find MySQL to be a similar, if not almost identical, environment. One major corporation that made the switch to open source was the Ernie Ball Guitar String Corporation. Ernie Ball's CEO, Sterling Ball, disputed analysts' predictions of tremendous cost and user transition difficulties when migrating from Windows to Linux:

> It's the funniest thing—we're using it for e-mail client/server, spreadsheets and word processing. It's like working in Windows. One of the analysts said it costs $1,250 per person to change over to open source. It wasn't anywhere near that for us. I'm reluctant to give actual numbers. I can give any number I want to support my position, and so can the other guy. But I'll tell you, I'm not paying any per-seat license. I'm not buying any new computers. When we need something, we have white box systems we put together ourselves. It doesn't need to be much of a system for most of what we do. (Becker 2003)

Availability

One possible downside of open-source software is its lack of retail availability. Although more popular open-source packages are becoming available in stores and catalogs, most open-source software must be downloaded from the Internet. This often requires a high-speed connection or a long time waiting for downloads to complete. As high-speed Internet access continues to proliferate, this issue is becoming less of a problem. In fact, the online availability of open-source programs is actually becoming a benefit: no packaging materials are used, no shipping time is required to get the latest version of a program, and no money is wasted on programs that do not meet the user's needs. For users with high-speed connections, the available delivery mechanisms, such as apt-get (Chiba Industries 2003), RPM (RPM Community 2002), or yum (Duke University 2002), are superior to those of their retail counterparts.

Table 1.—*Popular closed- and open-source software packages and their retail prices (Newegg 2003).*

Closed source	Price ($)	Open source equivalent	Price ($)
Microsoft Windows 2000 Server	870	Linux	0
Adobe Photoshop	565	GIMP	0
Oracle (1 computer)	15,000	MySQL	0
Microsoft Office XP	297	Open Office	0
Total:	16,732	—	0

Code Accessibility

As mentioned in the first section, because of the open-source principles, code for open-source projects is freely available. Source code is the software component that is readable by humans before it is compiled into machine-readable code. Source code, considered intellectual property, is the component of software to which software copyrights apply. In retail products, source code is not openly available. For users adept at programming, being able to view the source code offers many distinct advantages: bugs can be fixed, features can be added, modules can be enhanced, and security features can be checked by outside sources.

"Real World" Applications

The real world applications of open-source software are numerous and diverse. Organizations and individuals are adopting the open-source platforms for a number of reasons: costs are reduced, capacity for customization is increased, licensing maintenance is eliminated, and security is easily maintained. A short list of organizations that use open source indicates the widespread acceptance of the technology. The following is a list of organizations that have given open source a central role:

- Amazon.com (Adelson 2002).
- Toyota USA (IDC 2001).
- Massachusetts Institute of Technology.
- Harvard University.
- U.S. Department of Energy (Weiss 2001).
- U.S. Navy (Orlowski 2003).

Each of these organizations cited reasons along the lines of those previously mentioned for switching to open-source software. Security, cost, software availability, and customization were all contributing factors. In some of these cases, immediate cost savings were as high as $17 million (Adelson 2002). In the case of the U.S. Navy, the open-source code enabled the security customization required for specialized projects aboard nuclear submarines (Orlowski 2003). The Department of Energy has used open-source programs to create clustered supercomputers at an affordable price (Weiss 2001). On college campuses, open-source software enables students to work with the source code and generally function on the leading edge of technology.

These real world success stories also are contributing to the viability of open source as a retail offering. Many hardware retailers, including Dell, IBM, and Target, are offering open-source-based hardware solutions to their customers. These solutions can range from "clean" systems with no retail software installed to default open-source installations to customized open-source platforms created for customers. These examples and the increasing demand for availability clearly indicate open-source software's success.

Application of Open Source to Forest Analysis

Where possible, gradually replacing corporate software packages with their open-source counterparts would be a beneficial and exciting option. The result would be a decrease in cost, an increase in security and stability, and a more flexible computing environment. The easiest initial change would be to upgrade servers to open-source software. They could continue to interface with Microsoft Windows desktops for file sharing through Samba (Samba Team 2003) and act as servers for various FIA operations. This change would be largely transparent to the end user, especially because Forest Service servers are currently Unix-based. It would yield numerous benefits for the organization. Funding could be saved on software licenses for Oracle database software, Microsoft Windows operating systems, and other retail software. Additionally, use of the Linux kernel increases server stability and eliminates viruses, worms, and Trojan horses written to exploit Microsoft system and application vulnerabilities. Although upgrading systems to open source can be a significant and possibly daunting step, it can decrease IT overhead for an entire organization. Such an upgrade also establishes a niche at the forefront of a movement on the verge of changing the world of computing forever.

A clear example of open source being implemented successfully in a forest analysis project is Carbon On-Line Estimation (COLE) (Proctor *et al*., in press). For this project, open-source development tools and practices are used exclusively. The result is a comprehensive data analysis solution produced at a fraction of the cost of using retail tools. Additionally, as COLE comes into

its own, it, too, will become a registered open-source project. This step will allow other developers to contribute to the development of COLE and enhance it to suit their own research. In short, the open-source development cycle will come full cycle.

Conclusion

The open-source movement is a useful and viable option in today's computing world. In nearly all areas, open source either meets or exceeds the features and quality of proprietary retail software. Most importantly, open source presents owners and managers with an alternative that alleviates many of the problems that currently plague the IT infrastructure of many organizations: security, licensing costs, viruses, and scalability. Perhaps the advantages of open source are best summarized by Sterling Ball on his company's transition to an all open-source office:

> I'm not making calls to Red Hat (Linux) [for support]; I don't need to. I think that's propaganda…. What about the cost of dealing with a virus? We don't have 'em. How about when we do have a problem, you don't have to send some guy to a corner of the building to find out what's going on—he never leaves his desk, because everything is server-based. There's no doubt that what I'm doing is cheaper to operate. The analyst guys can say whatever they want. (Becker 2003)

The open-source revolution is clearly becoming a dominant force in computing, and the more its user base increases, the more it will gain power. Only time will tell if organizations will have the vision to take this powerful option to the next level.

Acknowledgments

This work was made possible by a grant from the USDA Forest Service, RWU 4104, and the Forest Health Monitoring Program.

Literature Cited

Adelson, A. 2002. Amazon.com: migration from UNIX to Red Hat Linux. Framingham, MA: IDC. http://www.redhat.com/whitepapers/services/Amazon_case_study.pdf. (26 September 2003).

Becker, D. 2003 (20 August). Rockin' on without Microsoft. CNET Networks. http://news.com.com/2008-1082-5065859.html. (26 September).

Cable News Network. 2003. Microsoft offers virus bounty. http://www.cnn.com/2003/TECH/biztech/11/05/microsoft.bounty/index.html. (5 November).

Chiba Industries. 2003. Unofficial APT repositories. http://www.apt-get.org. (26 September).

Coker, F. 2003. Kernel Korner—NSA security enhanced Linux. Linux Journal. July 7. http://www.linuxjournal.com/article.php?sid=6837. (26 September).

Duke University. 2002. yum, Yellow dog updater. http://linux.duke.edu/projects/yum. (26 September).

Embedded Linux Consortium. 2003. Embedded Linux Consortium home page. http://www.embedded-linux.org. (26 September).

Free Software Foundation, Inc. 2003a. GNU Emacs - GNU project. http://www.gnu.org/software/emacs/emacs.html. (26 September).

Free Software Foundation, Inc. 2003b. The GNU operating System. http://www.gnu.org. (26 September).

Free Software Foundation, Inc. 2003c. GNU general public license. http://www.gnu.org/licenses/licenses.html. (26 September).

GNOME Foundation. 2003. The GNOME project. http://gnome.org. (26 September).

IDC. 2001. Toyota Motor Sales USA: Linux across the enterprise. http://www.redhat.com/whitepapers/services/IDC_Toyota_CS.pdf. (26 September 2003).

KDE e. V. 2003. The KDE project. http://www.kde.org. (26 September).

Linux Security Audit Project. 2003. Linux security audit project. http://lsap.org. (26 September).

Microsoft, Inc. 2004. Microsoft Windows product home page. http://www.microsoft.com/windows/default.mspx. (26 September).

MySQL AB. 2003. The world's most popular open source database. http://www.mysql.com. (26 September).

National Security Agency. 2003. Security-enhanced Linux. http://www.nsa.gov/selinux/index.html. [Date accessed unknown].

Newegg. 2003. Price listings. http://www.newegg.com. (26 September).

Open Source Initiative. 2004. Open Source Initiative home page. http://www.opensource.org. (13 January).

Oracle Corporation. 2004. Oracle home page. http://www.oracle.com. (26 September).

Orlowski, A. 2003 (6 August). Navy buys Linux on Apple kit. The Register. http://www.theregister.co.uk/content/39/32211.html. (26 September).

Proctor, P.; Heath, L.S.; Van Deusen, P.C.; Gove, J.H. [In press.] COLE: A web-based tool for interfacing with forest inventory data. In: Proceedings of FIA science symposium; 2002 November 18–20; New Orleans, LA. St. Paul, MN: U.S. Department of Agriculture, Forest Service, North Central Research Station.

Raymond, E.S. 2000 (August). The cathedral and the bazaar. http://catb.org/~esr/writings/cathedral-bazaar/. (26 September 2003).

RPM Community. 2002. The RPM package manager. http://www.rpm.org. (26 September 2003).

Samba Team. 2003. Welcome to the Samba web pages. http://www.samba.org. [Date accessed unknown.]

TOP500. 2003. TOP500 supercomputer sites. http://www.top500.org. (20 October).

Weiss, T.R.; Vijayan, J. 2001. Compaq, DOE, biotech firm to build $150 million Linux supercomputer. Computerworld. http://www.computerworld.com/softwaretopics/os/story/0,1080,56666,00.html. (26 September 2003).

Protctor, Patrick; Van Deusen, Paul C.; Heath, Linda S.; Gove, Jeffrey H. 2005. The open-source movement: an introduction for forestry projessionals. P. 203-208, in McRoberts, Ronald E.; Reams, Gregory A.; Van Deusen, Paul C.; McWilliams, William H., eds. Proceedings of the fifth annual forest inventory and analysis symposium;

2003 November 18–20; New Orleans, LA. Gen. Tech. Rep. WO–69.Washington, DC: U.S. Department of Agriculture

Forest Service. 222 p.

Disclaimer

Papers published in these proceedings were submitted by authors in electronic media. Editing was done to ensure a consistent format. Authors are responsible for content and accuracy of their individual papers. The views and opinions expressed in this report are those of the presenters and authors and do not necessarily reflect the policies and opinions of the U.S. Department of Agriculture. The use of trade or firm names in this publication is for reader information and does not imply endorsement by the U.S. Department of Agriculture of any product or service.

United States
Department of
Agriculture

Forest Service

Gen. Tech. Report
WO–69

December 2005

Proceedings of the Fifth Annual Forest Inventory and Analysis Symposium

Edited by
Ronald E. McRoberts
Gregory A. Reams
Paul C. Van Deusen
William H. McWilliams

New Orleans, LA
November 18–20, 2003

Sponsored by
Forest Inventory and Analysis, USDA Forest Service
National Council for Air and Stream Improvement