The Open-Source software development paradigm: Nothing new to scientific research

John Daues (jdaues@nmsu.edu)
USDA-ARS-Jornada Experimental Range
Las Cruces, NM

Abstract

In this paper I demonstrate that - because of its propensity for innovation and flexibility in both the traditional scientific research and corporate research communities - the Open Source Software (OSS) development paradigm does indeed apply to scientific research.

Open-Source and Scientific Research

Basic scientific research has traditionally been a community-based process. Society provides monetary subsidies for basic research, and the social structure of science operates with reciprocity and knowledge sharing among scientists to insure contributions to the public good (Hippel and Krogh, 2003). While many research institutions continue to operate in this manner, corporations that hold restrictive copyrights and licenses, which can inhibit innovation, have also entered the scientific research world.

Like the traditional scientific research community, the OSS development paradigm is a form of community-based innovation. OSS is software for which the source code is available to the users so that they may read it, make changes to it, and build new versions of the software incorporating their changes, which in turn are made publicly available . For a comprehensive definition of open source, see http://www.opensource.org/docs/definition_plain.php.

In the traditional scientific model, theories are developed collectively and their flaws are perceived and gradually removed by means of criticism provided by the entire scientific community. Similarly, when code is released in the open source community, enhancements are made and bugs are found and removed by the entire open source community (Kipp, 2005), In contrast, proprietary software does not allow public inspection and modification of its code and hence prevents innovation from the user community.

Community-based innovation has contributed significantly to science and technology. For example, in the early years of the automobile, drivers went beyond using their vehicles to modifying their individual cars and even inventing and patenting new components (Franz, 1999). This same practice has been seen with personal computers (Freiburger and Swaine, 2000), astronomy (Ferris, 2002), and medical devices (Goetz, 2003) and others.

Numerous corporations use and produce OSS, from largest of organizations like IBM (once entrenched in proprietary technology) to much smaller companies like Delano Scientific, a company creating software for the drug discovery industry (Delano, 2005). In this environment, the source code is, of course, available for all to see and the software is usually free of charge. Instead, services related to the software generate significant revenue for the company.

Computer programming in the 1960s and 1970s was dominated by the free exchange of software (Levy, 1984). This started to change in the 1980s when the Massachusetts Institute of Technology (MIT) licensed some of the code created by its employees to a commercial firm and also when software companies began to impose copyrights (and later software patents) to protect their software from being copied (Drahos ,2002). Motivated by the spirit of traditional scientific collaboration, Richard Stallman, then a programmer at MIT's Artificial Intelligence Laboratory, founded the Free Software Foundation (FSF) in 1985 (http://www.fsf.org/). The FSF is dedicated to promoting computer users' rights to use, study, copy, modify, and redistribute computer programs. The Open Source Definition was created by Bruce Perens and Eric Raymond in 1998

(Perens, 1998).

The OSS paradigm can produce innovative, high-quality software that meets the needs of research scientists with respect to performance, scalability, security, and total cost of ownership (TCO). OSS dominates the Internet with software such as sendmail, Yahoo, Google, BIND (DNS), PHP, OpenSSL, TCP/IP, and HTTP/HTML. Many excellent applications also exist including Apache web server, Mozilla Firefox web browser, the OpenOffice suite, and the GNU/Linux operating system (Wheeler, 2005).

OSS users have fundamental control and flexibility advantages. For example, if one were to write a model using ANSI standard C++ (as opposed Microsoft C++), one could easily move the code from one platform to another. This may be convenient for a number of reasons, from simply a preference from one developer to another, to moving from a desktop PC environment to a high performance computing (HPC) environment.

Open Standards, which are publicly available specifications, offer control and flexibility as well. Examples in science include Environmental Markup Language (EML) and Virtual Reality Markup Language (VRML). If these were proprietary, use would be likely limited to one propriety application to interface with one proprietary format or numerous applications, each with its own format. One need only imagine the limitations on innovation if commonly-used protocols like ASCII, HTTP, or HTML were proprietary.

Conclusion

The OSS development paradigm, which is similar to traditional scientific research methodology and has also been shown to be productive in a corporate environment, can produce innovative, high-quality software and offers OSS users fundamental control and flexibility advantages. Thus, the OSS development paradigm does indeed apply to scientific research.

References

Delano, W.L., 2005, The Case for Open-Source Software in Drug Discovery, *Drug Discovery Today*, Volume 10, Number 3, February 2005, pp213-217.

Drahos, P. and Braithwaite J., 2002, Information Feudalism: Who Owns the Knowledge Economy? New York: The New Press.

Ferris, T., 2002, Seeing in the Dark: How Backyard Stargazers Are Probing Deep Space and Guarding Earth from Interplanetary Peril. New York, Simon & Schuster

Franz, K, 2005, Tinkering Consumers Reinvent the Early Automobile University of Pennsylvania Press

Freiburger, P. and M. Swaine, 2000, Fire in the Valley, New York, McGraw-Hill

Goetz, T., 2003,. "Open Source Everywhere," *Wired,* volume 11, number 11 (November), http://www.wired.com/wired/archive/11.11/opensource.html

Hippel, E and G Krogh, 2003, Open Source Software and the “Private-Collective” Innovation Model: Issues for Organization Science. Organization Science, Vol. 14, No. 2, March-April 2003, pp. 209-223.

Kipp, M.E.I, ,2005, Software and Seeds: Open Source Methods http://www.firstmonday.org/

Levy, S., 1984, Hackers. Anchor/Doubleday, New York.

Perens, B., 1998, The open source definition
http://perens.com/Articles/OSD.html

Wheeler, D., 2005, Why Open Source Software/Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!
http://www.dwheeler.com