Open Source in Government

O

David Berry
University of Sussex, UK

OPEN SOURCE

Open source software (OSS) is computer software that has its underlying source code made available under a licence. This can allow developers and users to adapt and improve it (Raymond, 2001). Computer software can be broadly split into two development models:

- Proprietary, or closed software, owned by a company or individual. Copies of the binary are made public; the source code is not usually made public.
- Open-source software (OSS), where the source code is released with the binary. Users and developers can be licenced to use and modify the code, and to distribute any improvements they make.

Both OSS and proprietary approaches allow companies to make a profit. Companies developing proprietary software make money by developing software and then selling licences to use the software. For example, Microsoft receives a payment for every copy of Windows sold with a personal computer. OSS companies make their money by providing services, such as advising clients on the GPL licence. The licencee can either charge a fee for this service or work free of charge.

In practice, software companies often develop both types of software. OSS is developed by an ongoing, iterative process where people share the ideas expressed in the source code. The aim is that a large community of developers and users can contribute to the development of the code, check it for errors and bugs, and make the improved version available to others. Project management software is used to allow developers to keep track of the various versions.

There are two main types of open-source licences (although there are many variants and subtypes developed by other companies):

- **Berkeley Software Distribution (BSD) Licence:** This permits a licencee to "close" a version (by withholding the most recent modifications to the source code) and sell it as a proprietary product;
- **GNU General Public Licence (GNU, GPL, or GPL):** Under this licence, licencees may not "close" versions. The licencee may modify, copy, and redistribute any derivative version, under the same GPL licence. The licencee can either charge a fee for this service or work free of charge.

Free software first evolved during the 1970s but in the 1990s forked into two movements, namely free software and open source (Berry, 2004). Richard Stallman, an American software developer who believes that sharing source code and ideas is fundamental to freedom of speech, developed a free version of the widely used Unix operating system. The resulting GNU program was released under a specially created General Public Licence (GNU, GPL). This was designed to ensure that the source code would remain openly available to all. It was not intended to prevent commercial usage or distribution (Stallman, 2002). This approach was christened free software. In this context, free meant that anyone could modify the software. However, the term "free" was often misunderstood to mean no cost. Hence, during the 1990s, Eric Raymond and others proposed that open-source software was coined as a less contentious and more business-friendly term. This has become widely accepted within the software and business communities; however there are still arguments about the most appropriate term to use (Moody, 2002).

The OSMs are usually organised into a network of individuals who work collaboratively on the Internet, developing major software projects that sometimes rival commercial software but are always committed to the production of quality alternatives to those produced by commercial companies (Raymond, 2001; Williams, 2002). Groups and individuals develop software to meet their own and others' needs in a highly decentralised way, likened to a Bazaar (Raymond, 2001). These groups often

Table 1. A summary of open-source applications and technologies

• Web sites (e.g., Apache)
• Network infrastructure (e.g., BIND, Sendmail)
• Operating systems (e.g., GNU/Linux, FreeBSD)
• Applications software (e.g., GIMP)
• Group network ssoftware (e.g., Drupal)
• Business systems (e.g., Amazon.com's Web site)
• Distribution/peer-to-peer filesharing

Table 2. A list of reasons for utilizing open source in government

- To enable transparency
- To save money
- To maximize alternative providers interoperability
- To improve software standards
- To provide timely public interest data
- To facilitate a competitive software market
- To avoid vendor lock-in
- To ensure open standards are implemented
- To allow security and coding checks to be made visually
- To optimize equipment
- To connect to other government to government (G2G) systems

make substantive value claims to support their projects and foster an ethic of community, collaboration, deliberation, and intellectual freedom. In addition, it is argued by Lessig (1999) that the FLOSS community can offer an inspiration in their commitment to transparency in their products and their ability to open up governmental regulation and control through free/libre and open source code.

CRITICAL ISSUES OF OPEN-SOURCE SOFTWARE

Advocates of OSS argue that, because it harnesses a large team of developers, bugs and errors can be rapidly spotted and fixed, thus increasing reliability and security. They also say that having a large team means that OSS is by necessity "modular" (made up of discrete units, each with a specific function). Modularity simplifies software design and can increase the reliability as well as flexibility of software (Statskontoret, 2003). Advocates also argue that, by making the source code available with the software, there is no danger of "lock-in" because document formats are transparent. However, critics point out that proprietary software can also have a high degree of reliability, flexibility, and security and can also conform to open standards.

Many commentators argue that OSS projects can suffer from weak project management (because of their products' complex development structure) and that OSS can be difficult to use. The OSS community point out that new project management tools are being introduced and that efforts are being made to increase the user-friendliness of OSS desktop applications. There are often concerns that OSS is unsupported and contains unauthorized intellectual property (IP) belonging to third parties. However, the OSS community say this can also be the case with proprietary software. Moreover, large firms such as IBM and Hewlett Packard now manage open-source projects and indemnify users to give them added insurance (e-Government Unit, 2004).

There is broad acceptance that OSS and proprietary software are comparable in terms of software quality. It is acknowledged that switching costs can be high, whichever software model is used. There are conflicting reports on how total cost of ownership (TCO) varies for the two models. It is widely agreed that TCO should be evaluated only on a case-by-case basis. Many analysts believe that there is increasing symbiosis between the two models. For example, modularity is now seen as an important factor in the development of both proprietary and OSS. New project management tools are being used to manage both types of software projects.

Worldwide, the uptake of open source has been variable, with some countries taking a more proactive approach encouraging such use (e.g., Brazil) and others a more neutral position (e.g., the United Kingdom) (CSIS, 2004; OGC, 2004). With government budgets increasingly stretched and with increasing needs for interoperability both internally as joined-up government and externally as government-to-government (G2G) data systems, open

Table 3. A summary of critical issues in open source

Total Cost of Ownership Debates are still ongoing about the best method of comparing the costs of proprietary vs. open source.
User Ignorance and Perceptions Lack of adequate understanding of the open source and it usefulness. User reluctance to retrain on the new software.
Support There is some reluctance to trust anarchic groups on the Internet for the future support of a software product, although this is belied by the amount of corporate support (e.g., IBM, HP) that now exists.
Software Patents Software patents are widely seen to be detrimental to the open-source development method. Currently software is not patentable, although the EU has been discussing a new directive to clarify its position.
Government Support It is difficult to persuade departments to move away from "known" products to lesser known open-source ones.
Project Management Dispersed development means that care has to be taken with the management of open-source projects in the public sector.
Maintaining and Integrity of Data Maintaining up-to-date and accurate information on the site for viewers to use. Open standards and open source are seen as contributory to this effort.
Security Maintaining secure and safe systems and keeping unauthorized user access out.
System Incompatibilities Cross-platform incompatibility that prevent particularly between proprietary and open-source components.