Open Source Software¹ - Alternatives to the Challenges of Software Procurement and Development?

Philip D. Gray and David Satola²

Information technology is key for development. Efficient provision of IT infrastructure and services will enhance production, which will contribute to growth and help in alleviating poverty. A growing portion of Bank projects have IT components, whether they be traditional telecommunications or informatics projects, or the new wave of e-government, e-education, e-health or even rural development projects. More and more, these project components have essential hardware and software requirements. However, procurement rules, combined with the exigencies of proprietary software, have made the provision of downstream software modifications, and even maintenance difficult. So-called “open source” software may provide alternatives to the old paradigm and unleash new development opportunities using open source software as a tool.

1. Introduction and Context

As the cost of information technology (IT) hardware falls, and reliability increases, mainly as a result of Moore’s Law³, software is becoming the most critical cost element in IT. Battles regarding software are spilling in all directions, whether into the realm of personal privacy, development of new forms of multi-media content, electronic payment etc. Software is an extremely powerful tool with the ability to affect peoples’ lives in fundamental ways. The public policy debate associated with what can or cannot be done with software is only just beginning. Part of this debate revolves around the appropriateness of open standards and open systems. One element in this debate is whether public policy should encourage the development of proprietary, or ‘open source’ software solutions. Indeed, many governments including China, Taiwan, Peru Tunisia, the UK and Germany to name a few, are now adopting policies encouraging the use of OSS solutions, especially as regards public procurement and in emerging e-government areas. In FY92, 69% of Bank and IDA projects had informatics components. By region, the percentage ranged from 43% in ECA to 82% in MNA. The sector with the highest percentage of projects was Health (92%), followed by Education (85%) and Urban Development (82%).⁴

¹ ‘Open source’ software (OSS) refers to software where the “source code” (the human-readable code which has to be ‘compiled’ into “object code” (machine-readable code) in order to be used by the computer) is available and freely manipulated, modified, improved and liberally redistributed. This is in contrast with proprietary software, where the source code is not generally available even to the purchasers of the software, who just purchase the ‘binary’ (object code) or compiled version of the software.

² Philip D. Gray is Senior Private Sector Development Specialist and David Satola is Senior Counsel at the World Bank. The views expressed in this paper are those of the authors and not necessarily those of the World Bank Group. The authors wish to thank, Eduardo Talero and Larry Wilder for their contributions.


The use of off-the-shelf versus custom software, combining hardware and software, or procuring the two separately, and, in particular, the relationship between initial licensing of software and downstream services (maintenance and support) indicate that further special attention be devoted to procurement in this area. While there is no prohibition in procuring OSS, consideration should be given to promoting its use for development projects.

The debate over the use of OSS as a tool for developmental is a subset of the broader debate of the role of intellectual property rights in development and open standards. The Bank has yet to formulate an institutional position regarding the broader developmental context within which the debate over OSS is occurring. It is recognized on the one hand that where intellectual property rights, such as copyright, are not protected by national law, holders of copyrighted software (or hardware with embedded software) will be reluctant to export to that jurisdiction. The same holds true for development of OSS. It is also recognized that there is a view held by some in the development community that continued protection of proprietary rights held by those in the developed world actually depresses local growth and development in developing countries—developing countries merely effect transfer payments to rights holders in the developed world while providing disincentives at home for innovation and growth of local high-tech industries.

With the undeniable worldwide trend towards open platforms and OSS, the Bank now has a window of opportunity define its institutional approach and shape the debate in the area of intellectual property rights and development. In addition, through focused procurement policy, the Bank could provide viable alternatives through focused open source procurement policy, including, possibly, the so-called “community open source” models.

2. Open Source Software - History

In the early days of computing, all software was ‘open’ because the cost of the software was effectively irrelevant to the overall cost of a computing package. Operating systems software was ‘bundled’ with mainframes and mini-computers and was essentially free for users to modify.

The evolution of software as a distinct, packaged service relates to the development of smaller computers, notably the PC. The advent of the PC led to the development of the consumer-oriented ‘shrink-wrapped’ software market that we are familiar with today, where software was sold (licensed) as a product, rather than as a form of after-sales service by a hardware vendor. While obviously of benefit to software vendors, a number of which have become some of the most valuable corporations on earth, the difficulties with proprietary, closed source software include vendor lock-in, inability to make small modifications to software, ‘planned obsolescence’ and security.

---

6 See, fn 9, infra, and accompanying text.
7 While not taking sides in this debate, it is important to point out the existence of the debate surrounding proprietary and “open” approaches, and related issues of protection of intellectual property rights, as it potentially affects growth and development of e-commerce in the developing world. For a recent apologetic from the developing world, see, e.g., “Open Software & Open Standards in South Africa: A critical Issue for Addressing the Digital Divide”, available at http://www.naci.org.za/docs/opensource.htm.
8 Security problems include potential ‘hidden’ back doors in the software, poorly debugged code, slow updates of potential problems etc. While vendors of closed source software tend to argue that there is a benefit to ‘security by obscurity’, computer security experts have tended to conclude that open, peer-reviewed software is more secure. (add reference)
The origin of so-called “open source” software was the Berkeley Software Distribution (BSD) of the Unix operating system originally developed by Bell labs (owned at the time by AT&T) in the late 1960s. Unix was distributed by BSD under the BSD license, a relatively open license allowing developers to take the original source code, modify it, and create proprietary software packages on this basis, and indeed this formed the basis of a series of proprietary Unix operating systems including Sun’s SunOS (later Solaris), Hewlett Packard’s HP-UX among others.

The difficulties with the increasing prevalence of proprietary packages (particularly frustrations related to prohibited access to source code to improve or customize it) led one computer scientist, Richard Stallman of MIT, to develop software programs and ensure their continued openness through the licensing mechanism dubbed ‘copyleft’, which prohibits the establishment of proprietary rights in the software by those who access its source code. This license, commonly called the GPL, provides, inter alia, that the user of the software is free to use and modify the software. If, however, the user modifies the software in some way, any modification of the software which is either sold or otherwise publicly released has to ensure the source code is also publicly available, on the same terms.

The GNU Project developed many tools, including, crucially a ‘C’ compiler, text editor and other utilities, made available under the GPL. Linus Torvalds developed an operating system kernel in the early 1990s and also released it under the GPL. It was rapidly modified and improved due in no small part to the openness of the code provided under the GPL, creating the Linux operating system, which, despite measurement problems, is probably the fastest growing operating system in the world, and identified by Microsoft as its key competitive threat.

3. The Benefits of Open Source

The principle benefits of OSS are examined from both a technical and a developmental perspective. As will be seen, sometimes it is the technical qualities of OSS that make it attractive from a developmental perspective. The benefits of OSS include:

- Third-party improvements/No “vendor lock-in” - upgrades can be made after the fact by third parties, and not at the discretion (some would say to ensure a steady revenue stream) of the proprietors of “closed” software code is available to be modified, upgraded or maintained, internally, by the original vendor, or by a third party.

---

9 This license is called the GNU Public License (GNU is a recursive acronym meaning Gnu’s Not Unix), or more commonly, the GPL. See http://www.GNU.org.

10 There are numerous licenses which are considered ‘open source’ such as the BSD, Mozilla etc. The differences tend to be on the requirements on redistribution of the software, for instance with GPL software requiring that any modifications to have the source code available as well (the so-called “viral” provision), where BSD-style licenses impose few restrictions on modifications and –re-distribution, hence allow for proprietary software to be developed using the BSD code base.

11 The openness of open source is preserved through a more or less traditional license. Moreover, the GPL also includes and preserves copyright in the author of the code, essential in order to assert an action under the license to prevent “proprietization” of the open source code. However, there are no known published cases construing the enforceability of the GPL’s “copyleft” mechanism, although general principles of contract and intellectual property rights protection could be used to support the proposition that the GPL is enforceable.

12 The GNU Project refers to the operating system as GNU/Linux.
• Security/Reliability – open source code tends to be more secure than closed source software, related in part to the analysis of bugs and security flaws. The developer of the Apache\textsuperscript{13} Web server, points out that some of the security woes of proprietary software lie in the development process itself. Open source is different in so far as changes made are open to scrutiny.

• Control of Data – data is stored, and hence is under the control of the owner of the data, not subject to the whims of the vendor or subject to being lost if the vendor goes out of business.

• Scalability – because the source code is open, the product can be scaled to the particular use.

• Localization – because the source code is open, it can be easily and legal customized to accommodate local language requirements, that may not be commercially viable for vendors of proprietary software. In addition, local software engineers can compete for procurement contracts decreasing developing countries’ reliance on imported skills, decreases potential exposure to exchange rate-risk from procuring hard-currency-denominated proprietary software, and decreases barriers to entry to the global software market place by software engineers from developing countries.

• Replicability – if a Bank-funded open source project is successful, it can be easily and legally replicated in other countries without wasteful, duplicative funding. Rather, funding can be provided for customizing the software to local circumstances.

4. Open Source and Software Procurement

The logic contained in thousands, if not millions of lines of code, make the preparation of reliable, secure, useful and user-friendly software extremely difficult. Because of this complexity, software projects often do not work as planned, are full of bugs, are difficult to maintain, and maintenance and repair are becoming the most difficult and expensive element of any form of large-scale service delivery. Indeed, as companies and governments start to engage in ‘e-service’ delivery, software is becoming the main point of contact between an organization and its consumers, suppliers, citizens, etc. The difficulties in procuring software are legion, and endless amounts have been mis-spent on ill-conceived, or overly complex software “solutions”, both by the private and the public sectors. Problems in software procurement include treating software like the purchase of any other product (ignoring essential after-sales servicing), overly rigid contracts, overly ambitious projects etc. Vendor lock-in is also a significant problem, for instance in the UK, over 50% of IT service contracts are with EDS, and where there is often relatively little competition in tendering. The history of the software industry is rife with high profile examples of procurement waste. In 1997, the U.S. Internal Revenue Service scuttled a software upgrade to the tune of $3.5 billion. A survey in 2001 by the publication Computing revealed that the catalogue of over-budget or cancelled contracts in the UK since 1997 topped £1bn and the bill continues to escalate. These problems are not the public sector’s alone, however. A study by KPMG Consulting estimated that UK businesses are wasting £17 billion a year through inefficient IT expenditure.

More and more countries, both developed and developing are opening their procurement policies priorities toward including open source. Simultaneously, public procurement is emerging as an instrument of public policy.\textsuperscript{14} As governments move towards more inclusive OSS procurement policies, there is ample argument for the Bank to keep pace with its donors and clients.

\textsuperscript{13} Apache software runs on approximately 60% of “http” (i.e., Internet) server websites. http://www.netcraft.com/survey/

\textsuperscript{14} Hunja, R. “Obstacles to Procurement Reform in Developing Countries”, in ________________.
Procurement of software is procurement of services. The most typical problems encountered with procurement of software using World Bank procurement rules are maintenance and support. Unless the terms of reference for the procurement package include *ex ante* that the consultant will be available for “after-sales” services such as maintenance and repairs of the software, the general procurement principles could, in theory, preclude that consultant from being engaged to perform the maintenance or repair the software (*i.e.*, to access and manipulate the source code). In the case of proprietary software, this can pose problems where software “fixes” are necessary and can only be performed by the vendor who is the “proprietor” and is not bound under its contract to provide those services or is prohibited under the rules form being engaged using Bank funding to do so. In the case of OSS, by contrast, no such conundrum exists, since the source code is available to all and may be freely modified, etc. In these cases, the services of any qualified software engineer could be procured, not necessarily those of the firm who drafted the code.

Bank procurement is already sensitized to the vagaries and complexities of complex IT procurement. Procuring OSS may provide a means to avoid the problem altogether. Procurement of software services should not discriminate between closed/proprietary systems and open source systems. Because a robust licensing and copyright regime in a country is required for procurement of either proprietary or OSS, whether the Bank provides funding for these service should be predicated on the existence of a credible and robust intellectual property and licensing regime reflecting international best practice in order to give certainty to both vendors and users.

5. **Directions for the Future**

What role can the World Bank play, if any, in the increasingly important debate over the use of intellectual property in the development process generally, and with respect to OSS in particular, through its own purchases, grant facilities such as *infoDev*, procurement policies, and even possibly through the development of community open source alternatives through, perhaps, the vehicle of the Development Gateway? The benefits outlined, and the fact that OSS is dramatically growing in use and popularity, already indicate that this is an area that the Bank will continue to confront. A proactive, development-oriented approach for the Bank is recommended, which is at a minimum, amenable to open source alternatives. Regarding software procurement, the Bank should be neutral with regard to proprietary and open source, allowing clients to procure OSS where competitive and where it fits their needs.