Institute For Policy Innovation IPI

ISSUE BRIEF

Synopsis: ***Open source is not necessarily the best way to develop software. While it may fill a useful role in specialized computing environments, open source does not translate to the mass market for software. Proponents of open source are vested interests who have substituted myth for reality. Policy makers should not mistakenly assume that this essentially derivative process is any substitute for innovation.***

Has Open Source Reached its Limits?

By Tony Healy

Is it possible that, despite all the hype, open source is not necessarily the best way to develop software? That it's not about to take over the software industry, and that it's no more a threat to Microsoft than were Netscape, the Macintosh or Word Perfect?

Several important distinctions are slowly starting to become obvious in software. The most important is the distinction between simply writing a program, which any computer science student can do, and creating a software product for the mass market, which requires much more expertise, time and work.

The mass market now numbers around 600 million computer users, and it demands that programs be easy to install, reliable to operate and useful. Those three criteria are deceptively simple, and generally not understood by the growing number of non-software people advocating for open source software.

The site access records of search engine site Google provide a useful marker into this debate. According to those records, Linux has only around 1 percent of the mass market.[1] This poses some serious questions for open source advocates, particularly their demands for preference in government purchasing. If people don't want open source, why should they be forced to use it?

This is not to say that there are not valid uses for Linux and the release of source code. Linux increasingly fills a useful role in specialized computing environments such as those associated with academic and technical research; and sharing of research findings, including source code, is standard practice in academic and scientific research.

The issue that needs to be addressed is whether those other environments translate to the mass market for software. In this paper, I argue that they don't, and that the composition of various open source advocacy groups masks fundamental weaknesses of open source. I also argue that, contrary to claims by political advocates, open source is the worst choice for nations seeking to build their local software industries.

Examples don't support claims

Many of the success stories of open source aren't relevant to what seems to be the main thrust of open source advocacy—the capturing of the mass consumer market. This is important, because the mass consumer market is qualitatively different from other markets. It demands a much higher level of software engineering in order to provide the requisite ease of use, robustness and flexibility.

This point is nicely illustrated in the games market, where innovation is at a premium, and technology changes rapidly. In that environment, the open source model, of copying existing code bases from someone else, fails dismally. The computer game market is dominated by commercially developed games.[2]

It is also illustrated by the fact that most of the successful open source products tend to be for technical users or for running on servers. This type of software is easier to write because the user can be relied on to carry out any necessary installation or operating procedures as instructed, or to understand the need for particular technical restorative actions.

Similarly, most of the environments cited as evidence of the merit of open source development are in academic and scientific computing, which have different motives and success factors from those for mass-market software. For academics and scientists, the writing of software is simply the manifestation of research that will be published separately. It is the research, not the software, which constitutes their primary output, and the criterion by which success will be judged. By comparison, the work of a software developer, whether an individual or a firm, is to develop and market successful software products. Actions that undermine competitive standing of software have little impact for academics, but can cripple software developers.

The mass consumer market is qualitatively different from other markets. It demands a much higher level of software engineering in order to provide the requisite ease of use, robustness and flexibility.

Allied to this is the fact that academics' pay comes from teaching students, or from government or private grants, whereas developers' pay comes from the software they produce, whether directly or as part of a software firm. Academics gain nothing from protecting their source code, whereas commercial developers do. Together, these grounds render academic and scientific software irrelevant as arguments for the open source process.

Third, it's common in open source advocacy to see figures describing the number of projects at open source site sourceforge.net or similar sites, with the implications this represents a mass of useful products. In actual fact, most of the projects are of poor quality, are unfinished and are certainly not comparable with the polished products of the commercial software development model.

Fourth, the firms often presented at open source conferences as evidence of the virtues of releasing source code are usually not software developers at all, but web developers, and their much vaunted "products" usually include very little original intellectual property. In other words, protection of source code is generally not important to web developers.

The new breed of detached observers who are now starting to examine open source from cultural perspectives has noticed the divergence between myth and reality in the open source movement. For example, University of Arizona sociologists Kieran Healy and Alan Schussman found open source to be an essentially derivative process, rather than an innovative one, and found claims about collaboration to be exaggerated.[3]

Agendas of advocacy groups mask weaknesses

In the same way that many examples of open source activity aren't relevant to mass market software, so too the agendas of many open source advocates hide weaknesses in the concept. Most communities pushing for the release of source code are vested interests who gain from open source at the expense of software developers, but this is not usually acknowledged.

This raises questions as to whether the software development industry has a place in the economy and, if so, whether it has a right for its interests to be acknowledged. I argue that the software industry is incredibly useful and productive, deserves its place in the economy and needs the freedom to decide whether to provide source code to customers.

Communities advocating for open source fall into four main groups — IBM, hardware makers, commodity firms and some types of lawyers.

For IBM, open source is a Trojan horse that gives its consulting business access to lucrative government accounts around the world. The consulting fees charged by outsourcers for the switch to open source are often comparable to the license fees that would have been paid to Microsoft. The inconsistency in IBM's open source advocacy can be seen in the tight hold it exercises on the source code for its own profitable software products, such as the expensive Websphere application server.

For hardware makers such as Sun, HP, IBM and some makers of embedded devices, open source is a way to reduce the cost of software and thus expand the market for computers. While this is a perfectly legitimate aim for those companies, it is not in the interests of software developers or of developing countries that might have a chance of building useful software industries.

For web firms and some support businesses, open source represents a reduction in costs. A common mistake in policy analysis is to see those firms as representing software developers, when they are better seen as customers of software developers. These firms will naturally advocate for software to be cheaper, while charging top dollar for their own services.

For law firms and lawyers, open source represents a rich opportunity to benefit from the increased complexity of licensing and copyright agreements. Only lawyers benefit from this.

Rationales are false

Several rationales used to promote open source do not stand up to examination. Those rationales fall into three main areas—that open source assists countries to develop valuable software industries, that open source is a better way to develop software, and that it's better to use public software rather than Microsoft software.

Examples of industry development motivations can be seen in Peruvian Congressman Villanueva Nuñez' famous 2002 letter to Microsoft[4] and in recommendations of the United Nations Conference on Trade and Development.[5]

However, industry development requires strong intellectual rights protection, according to a United Nations report on developing nations. This is especially so for producing valuable packaged software. In India, the lack of such protection prevented well-known IT companies Wipro and Sonate from producing packaged software for the local market.[6] Similar findings were made in a paper by Sunil Kanwar and Robert Evenson, which used cross-country panel data on R&D investment, patent protection and other country-specific characteristics over the period 1981-95 to conclude that intellectual property rights unambiguously spur innovation.[7]

So-called free programming is often funded by taxpayers in one form or another, and open source essentially represents a distortion of the market.

In any case, for developing nations, it seems odd to concentrate on the costs of software when many other factors are more important. For example, whereas a computer costs a month's wages for an average American worker, it costs eight years' wages for a Bangladeshi.[8] Similarly, the University of Namibia had only 15 computers for 2,000 students in 1998.[9]

Australian politician and open source advocate Ian Gilfillan claims that open source projects help train local developers, presumably by making the source code available.[10] But that claim makes no sense. Good developers create their own designs; they don't need to copy other peoples' source code. Further, developers already have access to extensive source code in samples and software development kits if they wish to see how particular techniques are implemented.

In terms of arguing that open source is a better way to develop software, one popular rationale is that open source spares the developer from having to reinvent the wheel. But all modern software platforms provide this benefit. Microsoft platforms probably provide it better than open source, because they expose functionality via precisely defined hooks that continue to work in upgraded versions of the platform, allowing properly engineered third party applications to work seamlessly across all required versions of Windows, including future versions.

A related argument holds that access to the source code allows greater customization and that this can contribute to innovation. Again, development of custom functionality and third party applications does not need access to source code of the underlying platform. There is extensive development of customized functionality and third party applications for the Windows platform.

Finally, the arguments that it's better to use public software rather than Microsoft software rely on alleged freedom from lock-in, and avoidance of the alleged dangers of a software monoculture.

The reality is that open source can trap a customer into an outsourcer relationship more readily than commercial software. This is because commercial platforms expose standard API's for third party applications and any consultant can develop for them.

For example, respected open source developer Hans Reiser of the ReiserFS file system has complained that controllers of different versions of Linux have started threatening to invalidate support contracts if customers stray from their own versions. He describes this behavior as being intended to achieve market leverage and exclude competitors. "By doing this they can exclude mainstream official kernels from being used, exclude rival file systems, exclude whatever might lead to less customer lock-in," he writes in Slashdot.[11]

The arguments about a monoculture can apply both ways. Just as having consistent platforms makes for a bigger target, it also simplifies the task of securing platforms and issuing updates. Establishing 100 percent security in software and in large installations of that software is an enormous task.

Having multiple different products would simply multiply the effort, not reduce it.

Conclusion

Pushing the open source concept too far into areas where it's not applicable will lead to universities and taxpayers shouldering the cost of software development for business, and doing it less capably than specialist software development firms. This is a point made by Bertrand Meyer and Nikolai Bezroukov, who contend that so-called free programming is often funded by taxpayers in one form or another, and that open source essentially represents a distortion of the market.[12, 13]

Already, a few practical realities have emerged from open source experiments. Munich staff will continue to use many Windows programs, except they will be running them on emulators on Linux. When Australia's largest telecommunications company, Telstra, considered open source desktop products, it exempted 6,000 managers, who would continue to use Microsoft products.

As these factors become more apparent, open source will go the way of other IT industry fads that were once trumpeted as the way of the future, like Macintosh computers, business AI, 4GL programming languages and Y2K. Munich, the Australian Capital Territory and other locations will provide fascinating test beds for the claims of open source advocates. Indeed, there is already evidence that staffers at Munich are not as enamored of open source as the political advocates are.[14]

Endnotes

1. Google Zeitgeist - Search patterns, trends, and surprises according to Google, Sept. 2003, http://www.google.com/press/zeitgeist.html.
2. Solveig Singleton, *"FreeCiv" and its Discontents: Policy Lessons from Open Source Games: A Case Study*, CEI, 19 Nov. 2003, http://www.cei.org/pdf/3755.pdf.
3. Kieran Healy and Alan Schussman, *The Ecology of Open Source Development*, University of Arizona, 14 Jan. 2003 (Unpublished), http://www.kieranhealy.org/files/drafts/oss-activity.pdf.
4. Reply by Congressman Villanueva Nuñez to Microsoft, Lima, 08 Apr, 2002, translated into English by Graham Seaman 04 Aug. 2002, www.aful.org/politique/perou/english/rescon_en.html.
5. *United Nations E-Commerce and Development Report 2003*, Chapter 4, "Free and open-source software: Implications for ICT policy and development," United Nations Conference on Trade and Development, 2003, http://www.unctad.org/en/docs/ecdr2003ch4_en.pdf.
6. Zelkja Kozul-Wright and Jeremy Howells, *Changing Dynamics of Computer Software and Services Industry: Implications for Developing Countries*, United Nations Conference on Trade and Development, United Nations, New York and Geneva, 2002 pp 27-29.
7. Sunil Kanwar and Robert Evenson, "Does intellectual property protection spur technological change?" *Oxford Economic Papers 2003*; 55:235-264 (Department of Economics, University of Delhi, and Yale University).
8. *United Nations Development Program: Human Development Program*: 1999, Oxford University Press.
9. W. Wresch, *Information access in Africa: Problems with every channel*, The Information Society, 14, 1998, pp 295-300.
10. Ian Gilfillan, *Open Source Bill Gathers Momentum*, Democrats, 25 Jun 2003, http://sa.democrats.org.au/Media/2003/0625_a%20Open%20Source.htm.
11. Hans Reiser Speaks Freely About Free Software Development, *Slashdot*, 18 Jun 2003, http://interviews.slashdot.org/article.pl?sid=03/06/18/1516239.
12. Bertrand Meyer, "The Ethics of Free Software," *Software Development Magazine* Mar 2000, http://www.sdmagazine.com/documents/s=746/sdm0003d/0003d.htm (Needs free registration).
13. Nikolai Bezroukov, *Open Source Software Development as a Special Type of Academic Research*, Oct 1999, http://firstmonday.org/issues/issue4_10/bezroukov.
14. Michelle Delio, "Munich Open Source Plows Ahead," *Wired*, 11 Feb 2004, http://www.wired.com/news/infostructure/0,1377,62236,00.html.

About the Author

Tony Healy is a research software engineer and policy researcher in Sydney, Australia.

About the Institute for Policy Innovation

The Institute for Policy Innovation (IPI) is a non-profit, non-partisan educational organization founded in 1987. IPI's purposes are to conduct research, aid development, and widely promote innovative and non-partisan solutions to today's public policy problems. IPI is a public foundation, and is supported wholly by contributions from individuals, businesses, and other non-profit foundations. IPI neither solicits nor accepts contributions from any government agency.

IPI's focus is on developing new approaches to governing that harness the strengths of individual choice, limited government, and free markets. IPI emphasizes getting its studies into the hands of the press and policy makers so that the ideas they contain can be applied to the challenges facing us today.

About the IPI Center for Technology Freedom

The IPI Center for Technology Freedom is the technology division of the Institute for Policy Innovation (IPI), a non-profit, non-partisan, public policy research organization founded in 1987. The purpose of the Center is to sort out the policy challenges posed by technological change for both the American people and policy makers. The Institute for Policy Innovation is a public foundation, and is supported wholly by contributions from individuals, businesses and other non-profit foundations. IPI neither solicits nor accepts contributions from any government agency.

The IPI Center for Technology Freedom points the way for a society that encourages freedom to develop new technologies, and the freedom to access them. The Center places emphasis on delivering its studies into the hands of policy makers, press, the technology industry and other interested parties, so that the ideas they contain can be applied to today's challenges.

Editor & Publisher . . . . Tom Giovanetti

IPI *Issue Brief* is published by the Institute for Policy Innovation (IPI), a non-profit public policy organization.

NOTE: Nothing written here should be construed as an attempt to influence the passage of any legislation before Congress. The views expressed in this publication are the opinions of the authors, and do not necessarily reflect the views of the Institute for Policy Innovation or its directors.

Direct all inquiries to: **Institute for Policy Innovation**
1660 S. Stemmons Freeway, Suite 475
Lewisville, TX 75067

(972) 874-5139 (Voice) **Email:** ipi@ipi.org
(972) 874-5144 (FAX) **Internet Website:** www.ipi.org