

OPEN TO DEVELOPMENT: OPEN-SOURCE SOFTWARE AND ECONOMIC DEVELOPMENT

Bibek Debroy* and Julian Morris**

* Director, Rajiv Gandhi Institute for Contemporary Studies, New Delhi, India

**Director, International Policy Network, London, UK (www.policynetwork.net); Visiting Professor, University of Buckingham, UK (www.buckingham.ac.uk). Corresponding author, email: jmorris@policynetwork.net (replace =at= with @)

I. INTRODUCTION

Economic growth and development is often an elusive goal. Even newly industrialized nations with years of solid growth under their belts can find themselves backsliding—witness South Korea in the late 1990s.¹ Thus, when a new source for economic development is offered, it justifiably garners attention. In recent years, several policy advisors have offered open-source software (OSS) as just such a source, a kind of incubator of technology and economic growth. For example, a 2003 report by the Finnish Ministry for Foreign Affairs concludes, “if the adoption of [OSS] in developing countries is done wisely, it can help stimulate an indigenous software industry and create local jobs.”² Similarly, a 2003 report by one United Nations agency, the Conference on Trade and Development, posits that OSS “makes it possible to keep IT expenditures, as well as experts and promising young talent, at home and contributing to a nascent local software industry.”³

¹ Tzong-Shian Yu, *Can East Asia Rise Again?*, 13 JOURNAL OF ASIAN ECONOMICS 6, January 2003.

² Niranjan Rajani, “Free as in Education: Significance of Free/Libre and Open Source Software for Developing Countries,” for the Ministry of Foreign Affairs, Finland, May 2003.

³ United Nations Conference on Trade and Development, “E-Commerce and Development Report 2003,” Chapter 4: Free and open-source software: Implications for ICT policy and development, p. 110.

Open source is, in essence, a way of creating and distributing software.⁴ Rather than keep the human-readable program instructions (called source code) hidden from users, as traditional software companies such as Microsoft, Intuit and Apple typically do, open-source programs give away the source code to one and all. That access enables users skilled in programming to become de facto software developers by adding to or modifying the software code and then redistributing it. Nor is open-source software licensed to single users or companies in the typical fashion—users can pass on the software to others if they choose. Instead of preventing the software from being shared, open-source licenses dictate how the software is shared. For example, software licensed under the widely used GNU General Public License (GPL) allows any user to redistribute a program without charge, but requires that any redistribution be accompanied by the source code.⁵

In many developed nations, most notably Western European countries and the United States, open-source and proprietary software have coexisted for quite some time.⁶ Starting in the mid-1980s, “free software” emerged in the U.S. for mostly back-office applications—web servers, email operations, and the like.⁷ While it has made some inroads into front-office programs, even today it remains strongest in applications used primarily by software developers, or at least by technically adept people.⁸ Applications requiring user-friendly touches are not typically found in open-source, which is one of the reasons proprietary software continues to dominate certain software categories.

⁴ We use the term “open-source” to refer to software that is made readily available in the form of source code. This includes open-source software, free software, “FOSS,” software libre, and “FLOSS.”

⁵ GNU is a recursive acronym that stands for GNU’s Not Unix. For a full text of the GPL, see *GNU General Public License* (v.2, 1991), available at <http://www.gnu.org/copyleft/gpl.html> (last modified 5/26/03).

⁶ We refer to software generally sold by for-profit software developers as proprietary.

⁷ The roots of open-source software actually extend farther back, to when developers in academic settings exchanged programs and source code with their colleagues. See Richard Stallman, “Free Software: Freedom and Cooperation,” speech delivered at New York University, May 29, 2001, <http://www.fsf.org/events/rms-nyu-2001-transcript.txt>, downloaded June 7, 2001. Stallman ardently rejects the label “open source” software, preferring instead “free software.” While the semantics of the movement tend to hold a great deal of meaning for movement participants, for my purposes there is little distinction between the two monikers. We therefore refer to all software that is distributed with its source code (and with users allowed to redistribute that source code) as open source.

⁸ As an operating system, Linux does not neatly fit in the back-office box. Nonetheless, Linux is used most frequently on servers and back-office computers, not consumer desk-tops.

Many developing countries are grappling with what the appropriate level of open-source and proprietary software coexistence should be for a still-growing economy. Open-source advocates argue that government supported open-source programs can be an important aspect of an economic development program because they can aid nations in creating a local software industry. Since open-source requires so little in the way of initial investments (that is, there are often no up-front licensing fees), proponents claim that the software model's promise is welcome news for resource constrained developing nations seeking an entrée into the technology and computing sector. But are the promises supported by the facts?

This paper examines the role that open-source software can play in an economy and its development, with a focus on empirical evidence and economic logic. Unfortunately, while open-source can clearly be a viable part of a developed software industry, the available evidence does not support the position that open-source software can form the basis of an industry on its own, especially in nations where the technology sector is still embryonic.

Section II puts the open-source/development debate into perspective by providing some statistics on how software has contributed to the economies of developed nations. It also provides some background on the differences between open-source and proprietary licenses, distinctions that are crucial for the creation of an income-generating industry. Given the size of the software industry in the United States, it is easy to understand developing nations' interest in spurring a local market. However open-source licensing restrictions imply that no profits can be made on pure open-source software. Section III then discusses recent government forays into open-source software. Some of these efforts have been sensible, attempting to increase awareness of open-source solutions. Others have been less sensible, attempting to dictate software solutions regardless of the individual circumstances. Section IV examines the various claims made in favor of open-source as a development tool. Many of these arguments fall flat because their chain of logic relies on one or more leaps of faith that do not square well with actual experience. Section V concludes the paper.

II. HOW SOFTWARE FITS INTO AN ECONOMY

Before delving into the claims that the open-source software model can aid a nation's economic development, some basics are in order. The next section provides some statistics on the U.S. software industry, intended to indicate an upper bound on the industry's potential economic impact. The section following then briefly describes the differences between how open-source and proprietary software are licensed. These differences have a profound impact on open-source's ability to contribute to a nation's economy.

A. Software's Potential

In developed nations, especially in the United States, software—and high technology in general—has made a significant impact on the economy. If the U.S. experience could be translated to other countries, even on a smaller scale, the potential impact could be large. The interest in a low-cost means of developing a local software industry is easily understandable.

In the 30 years between 1970 and 2000, the U.S. software industry grew at an astonishing rate of 38 percent annually.⁹ By the year 2000, annual software sales to end-users in the United States were more than \$100 billion (USD). Even after leveling off a bit in the mid-1990s, the software industry grew at roughly three times the rate of the economy as a whole—even while the general economy itself was booming.¹⁰ As a percentage of total GDP, the software industry in the United States is actually relatively small. Nonetheless, as a percentage of total exports the effect of the software industry on the economy is much higher. The software industry ran a trade surplus of \$13 billion in 1997; without software's contribution, the U.S. trade deficit would have been 36 percent higher.¹¹ In addition to direct effects on national product and exports, the software industry has provided far reaching

⁹ Martin Campbell-Kelly, *FROM AIRLINE RESERVATIONS TO SONIC THE HEDGEHOG: A HISTORY OF THE SOFTWARE INDUSTRY*, 14 (2003).

¹⁰ Nagla Rizk, *Information Technology and Growth: Will the Software Industry Lead Egypt into a New Economy*, 2002 PROCEEDINGS OF THE MIDDLE EAST ECONOMIC ASSOCIATION, available at <http://www.sba.luc.edu/orgs/meea/volume4/RizkRevised.htm>.

¹¹ *Strong Software Industry Growth Results in More Jobs, Economic Increases*, BUSINESS SOFTWARE ALLIANCE, 15 June 1999, available at <http://www.bsa.org/usa/press/newsreleases/Strong-Software-Industry-Growth-Results-in-More-Jobs-Economic-Increases.cfm>.

spillover benefits to other sectors of the economy, from airline reservation systems to the control systems that guide automated factories.¹²

The benefits of the software industry have not been restricted to the world's wealthiest nations. India is often held as an example of a developing nation that has successfully latched on to the global software boom. Currently, the proprietary software industry in India accounts for 16 percent of all exports, not including related non-software Information Technology (IT) typically boosted by advancements in software.¹³ For example, the recent growth in customer service and support call centers located in India can be traced to the infrastructure and technology improvements that grew out of the software industry.¹⁴

Despite India's apparent success in cultivating a for-profit software industry, empirical evidence thus far does not support the view that a vibrant software industry automatically leads to general economic growth and poverty reduction. In particular, the impact on national employment is not as significant as one might imagine: one study observes that "even optimistic estimations project that no more than another 2.2 million new jobs [in the tech sector] will be created within the next eight years."¹⁵ To put that 2.2 million in perspective, more than 75 million workers in India are either looking for work now or will be entering the labor force over the next three years.¹⁶ These figures are national, though, and while the software industry may not be able to single-handedly rescue India from its current unemployment situation, tech sector job growth appears to be at the fore in several booming Indian cities, such as Bangalore..¹⁷ Neither has the software industry growth thus far been

¹² Even in the U.S., the industry leader and largest adopter of IT, the question of whether the IT sector led to the observed dramatic productivity growth in the late 1990s is still a matter of considerable controversy.

¹³ <http://www.oecd.org/dataoecd/30/35/2401567.ppt>

¹⁴ Ashish Arora & Suma Athreye, *The Software Industry and India's Economic Development*, WORLD INSTITUTE FOR DEVELOPMENT ECONOMICS RESEARCH, Discussion Paper no. 2001/20, June 2001.

¹⁵ S. Rothboeck & V. Gayathri, *Labour and the New Economy: The Case of the Indian Software Labour Market*, INTERNATIONAL LABOR ORGANIZATION, 2001

¹⁶ Jason Overdorf, *Boom Times—But No Jobs*, NEWSWEEK INTERNATIONAL EDITION, 15 March, 2004.

¹⁷ *In India, the Job Numbers Don't Add Up*, INDIAN BUSINESS INSIGHTS, 20 April 2004; *India Confronts Backlash Against Outsourcing Jobs*, CHICAGO TRIBUNE, 19 February 2004; David Lynch, *Imported Jobs Building a New, Middle-Class India*, E-COMMERCE TIMES, 6 April 2004; *Jobs Abound in India's Booming Tech Sector*, REUTERS, 1 October 2003.

translated into a more equitable income distribution, one of the Indian governments ultimate goals.¹⁸ To the extent that the software industry potentially could promote more general economic development, studies on India's experience have identified a handful of factors: spillover to IT enabled industries, such as call centers; growth of IT consuming industries; liberalization and development in infrastructure, such as telecommunications and electricity sectors, necessitated by software industry growth; and emulation by other enterprises of the sound corporate governance and entrepreneurship demonstrated by the proprietary software companies.¹⁹

Despite the lack of clear-cut spillover to general economic growth, the fact that the software industry is labor intensive makes it especially attractive for less developed countries. With lower labor costs and no need for large capital outlays, less developed countries can gain a competitive advantage over established rivals. Since hardware and software production can be separated, developing nations can complement the progress of developed nations without necessarily entering into direct competition with them. Decreasing worldwide telecommunications costs and software's inherent portability make software creation a natural candidate for outsourcing to developing countries with a capable supply of software developers. In fact, this is how India began its local software industry: shipping English-speaking engineers for short-term outsourced engagements abroad.²⁰

The market for software is likely to continue expanding, with both the developed world finding more and newer uses for software and the developing world creating its own demand. Obviously, the market for software will not grow perpetually, but the opportunities are still vast. Whether these opportunities are available for developing nations is an open question; whether open-source is a viable means to access them is the question at hand here.

¹⁸ David O'Connor, *Of Flying Geeks and O-Rings: Locating Software and IT Services in India's Economic Development*, ORGANIZATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT, Working Paper No. 224, 2003. The recent growth of call centers could change that, however, since it typically employs a more diverse and less-educated work force.

¹⁹ O'Connor, *supra* note 18; Arora & Athreye, *supra* note 14; Nirvikar Singh, *Information Technology and India's Economic Development*, Working Paper, July 2002.

²⁰ Note that India was well positioned to take advantage of the software boom. It has a large supply of well-trained engineers, all of whom learn English as a matter of course in the Indian educational system.

B. Open-Source versus Proprietary Software

Now that we've established the potential for growth that a software industry can offer, we need to better understand whether open-source software has features that lend themselves to creating a local industry. Unlike physical goods, software is not sold outright. Instead, it is licensed to users with the copyright holders retaining "ownership" of their software. Licensees gain the right to use the software subject to certain restraints. Both open-source and proprietary software follow this model, with the kinds of restraints placed on users distinguishing the two forms of software.

One of the most frequently used open-source licenses was drafted by the Free Software Foundation as a means of promoting open-source at the expense of proprietary software: the GNU GPL.²¹ In particular, if a program is distributed under the GPL, all source code must be made available, essentially for free. The GPL also stipulates that any user can modify and distribute the program, either in original or modified form. Any redistribution, though, (whether of the original or modified program) must also come under the GPL. This condition has earned the GPL the label of "viral" because it typically means that once code is licensed under the GPL, any other program that incorporates that code falls under the GPL as well.

The GPL provisions are intentionally aimed at preventing open-source code from being incorporated into proprietary code. One result of the source code distribution requirements is that programmers can charge no more for programs than the cost of reproduction (which are typically quite small). If a programmer tried to charge license fees substantially above the reproduction costs for GPL software, anyone else could acquire the source code and redistribute it on their own, driving the price back down to reproduction costs. With license fees thus foreclosed, the only profit opportunities remaining are for additional services, such as software support or training, or for complementary proprietary programs that run with or on the open-source program.²²

²¹ For information on the Free Software Foundation and the GPL, see <http://www.fsf.org/>.

²² That is, a company could charge a licensing fee for a proprietary applications program, such as a word processor, that runs on an open-source operating system like Linux.

The licensing provisions clearly have implications for firms hoping to earn a sustainable return on software production. Some for-profit firms specializing in open-source software distributions have attempted to distinguish themselves through the skill of their employees and the level of service offered by their support staffs, enabling them to charge recurrent subscriber support fees. Others offer more complicated open-source products that package multiple open-source programs together, such as a complete Linux operating system distribution, along with an easy installation program. These companies can charge for the convenience of their package, but of course, other open-source providers could easily replicate that package and lower the price. Another route taken by for-profit firms involves combining proprietary software with the open-source programs, allowing the company to charge higher licensing fees for the closed software. Regardless of the specifics, the underlying economics imply that pure open-source software production cannot generate sustainable profits—an important point to bear in mind when considering the arguments for governments to use open-source as a development tool.

III. GOVERNMENTS AND SOFTWARE POLICY

Governments have a dual role to play in terms of defining a local software industry. Not only do they establish the policy which influences industry players, they can also act as an important software consumer. In recent years, a number of governments have combined these two roles by introducing policies defining government software use. For example, several cities in Brazil have dictated that all municipal government offices must use open-source software instead of proprietary software.²³ On the surface, this might appear to be the government acting strictly as a software user, demanding uniformity across offices. However, the choice was largely motivated by politics.²⁴ As the founder of the Free Software Foundation observed, “I find in Brazil considerable awareness that free software is a social and political issue as well as a practical and economic one.”²⁵ One implication of the Brazilian stance is its foreclosure of

²³ David Evans & Bernard Reddy, *Government Preferences for Open-Source Software: A Solution in Search of a Problem*, 9 MICH. TELECOMM. TECH. L. REV. 313, 376 (2003)

²⁴ Paul Festa, *Governments Push Open-Source Software*, CNET NEWS.COM, 29 August 2001.

²⁵ Press Release, Free Software Foundation, Richard M. Stallman Addresses Brazilian Congress on Free Software and the Ethics of Copyright and Patents (Mar. 20, 2001) [hereinafter Stallman Addresses Brazilian Congress], at <http://www.gnu.org/press/2001-03-20-Brazil.txt>.

locally produced proprietary software. Were a national proprietary software industry to emerge in Brazil, it would be in spite of government efforts, not because of them. In less extreme policy-user decisions, some legislatures, including Germany's Bundestag, have adopted guidelines that require government offices to consider open-source when purchasing software.²⁶ On the pure policymaker front, a handful of governments have promoted open-source through education and awareness policies.²⁷

Some governments, acting purely as an influential software user, have chosen to license open-source software after evaluating both open-source and proprietary solutions. The city government of Munich, for instance, recently decided to migrate its 14,000 or so computers to the open-source operating system Linux. Munich is licensing the software from IBM/SuSE and is paying significant licensing and support fees. In fact, the move reportedly cost the city \$10 million (USD) more than its closest proprietary option would have cost, so clearly more than economics entered into the decision.²⁸ Aside from the additional cost, the transition has hit some snags. Many smaller software vendors dealing with the government lack open-source knowledge and are having difficulty making the switch.

While Munich may not be a clear-cut example, though, as a general economic matter it is difficult to find fault with a government acting as a profit maximizer choosing open-source software. There are legitimate business reasons to do so. The trouble lies with governments mandating open-source software use for all of its offices, regardless of migration costs, needs or particular circumstances. As explained earlier, pure open-source models cannot support a for-profit industry. Service, support, and add-on proprietary software (such as applications programs) can lead to sustainable profits for open-source companies, but if governments hope to use open-source policies to create and support local companies taking this route, foreclosing the use of all proprietary software in government offices runs counter to that goal. Government efforts to increase community awareness of open-source software could be welfare enhancing,

²⁶ *Supra* note 23, at 374.

²⁷ *E-Commerce and Development Report 2003*, UNITED NATIONS CONFERENCE ON TRADE AND DEVELOPMENT, 118.

²⁸ *Munich Open Source Plows Ahead*, WIRED, 11 February 2004, available at <http://www.wired.com/news/infrastructure/0,1377,62236,00.html>.

but once awareness is achieved individuals and administrative offices should be free to choose the software that best meets their needs. If choosing open-source makes sound economic sense, it will be chosen even if the local government has not endorsed or promoted it.

IV. OPEN SOURCE AS A DEVELOPMENT PATH

We now turn to the main subject: whether open-source software can provide a viable platform for a local software industry and thus contribute to a nation's economic development. Here, we examine the arguments that proponents have put forth in support of open-source as a growth tool, presenting empirical evidence when possible. As we illustrate, arguments for using open-source software do not necessarily translate into arguments for employing open-source software as a means of development.

A. Open-Source Software is Less Costly

One of the most prevalent arguments made for open-source software is cost-based. Since no (or very low) licensing fees need be paid, the initial investment can be minimal. Moreover, open-source operating systems like Linux can be used on older, cheaper computers, saving on hardware costs. The low investment costs allow developing nations entrée to the growing hi-tech global market, a market that otherwise would be foreclosed.

On its face, this argument holds considerable appeal. Clearly initial costs can be lower for open-source software as compared to proprietary. Initial costs are not the full story, however. Several academic studies have investigated total cost of ownership, or TCO, a concept that includes initial costs as well as setup, training and support costs. These studies are less clear: for some applications open-source is likely to be the cheapest software option, but for others the labor costs involved in personnel training and support dominate initial investments and tilt the advantage towards proprietary software.²⁹ One drawback of applying this line of research to the debate at hand is that none of these studies were conducted in developing nations, where labor costs are typically quite low.

²⁹ For a survey of TCO studies, see Alan MacCormack, *Evaluating Total Cost of Ownership: Comparing Apples, Oranges and Cucumbers*, AEI-BROOKINGS JOINT CENTER, Related Publication 03-7, April 2003.

Focusing on the costs of open-source, though, distracts us from the real issue: relatively lower costs for software users do not imply that open-source can lead to economic development. Growth requires sustainable profits with paid employment positions, and often involves exportable products or services.³⁰ Low user cost says nothing about these necessary factors. As we touched on earlier in discussing software licensing, typical open-source terms (i.e., GNU GPL terms) make sustainable profits difficult to achieve. “Pure” open-source software, without adjunct services or add-on proprietary programs, is not capable of supporting a for-profit company at all.³¹ Because the relatively low-priced open-source software cannot provide a sustainable product differentiation, it cannot support developing country efforts to export (or even to license in-country) software for a profit. If open-source software cannot provide reasonable profit margins for its distributors, then it cannot support an entire industry and may end up comprising little more than a small handful of locally oriented service and support companies.

Even in the United States, which has one of the largest software markets of all nations, only a few for-profit open-source software companies have survived. The initial public offerings of many firms built around open-source software were heralded in 1999 and early 2000, but when the tech-bubble burst those companies were hit as hard—and perhaps even harder—than other hi-tech firms. Since then, several open-source companies have closed their doors altogether while others have abandoned their open-source offerings.³² Some U.S.-based Linux distributors remain, such as Red Hat, which had a good year in 2003, but the once-lauded open-source business model is no longer viewed as promising by investors or market analysts.

³⁰ See e.g., Bela Balassa, “The Lessons of East Asian Development: An Overview”, *Economic Development and Cultural Change*, 1988; World Bank, “The East Asian Miracle”, 1993. The empirical evidence of software industry in India is consistent with the importance of exports as well: 70% of India’s software industry revenue comes from export (Rothboeck & Gayathri, *supra* note 15).

³¹ Niranjan Rajani, “Free as in Education: Significance of Free/Libre and Open Source Software for Developing Countries,” for the Ministry of Foreign Affairs, Finland, May 2003.

³² *The Linux Uprising*, BUSINESSWEEK ONLINE, 3 March 2003.

B. Proprietary Software is Dominated by Western-Based Monopolies

Not only does proprietary software involve paying relatively higher licensing fees, some open-source proponents observe that those fees typically go to a foreign company. For some software, such as operating systems, the fees often go to a large “monopoly” company with whom the developing nation feels it has little leverage. Upgrades, alterations, and customization the developing nation might desire—such as a native language version of the software—are low on the priority list for these companies since they represent changes that hold little to no appeal to the firm’s other customers.

Open-source software, on the other hand, can be customized by local programmers. In fact, several countries have taken this route by starting with existing open-source software and adapting it to local language and customs.³³ Advocates argue that efforts of this kind provide developing nations a two-fold benefit. First, the government gets customized software in native languages. Second, the country can employ local workers to make the code changes and later to support the program’s users.

As with the argument reviewed above, this rationale for open-source software confounds two fundamental points: the reasons for using open-source software and the reasons that open-source software could aid economic growth. Adapting existing OSS technology to local language and customs can be beneficial for software users in some circumstances, just as proprietary software can offer the best solution in others. In either case, the software user is relying on available software, not a truly custom solution.³⁴ The open-source software that developing nations begin with is generally available worldwide—local efforts are typically aimed at adapting existing programs, not at creating new programs from scratch.

³³ The LinEx project in Extremadura, Spain, for example, is part of a region-wide effort to promote information technology, in which a large portion of it is the development of a localized version of Linux called LinEx. *See FLOSS Deployment in Extremadura, Spain*, EUROPEAN COMMISSION INTERCHANGE OF DATA BETWEEN ADMINISTRATIONS, available at <http://europa.eu.int/ISPO/ida/jsps/index.jsp?fuseAction=showDocument&documentID=1637&parent=chapter&preChapterID=0-452-470>

³⁴ In all of the examples we are aware of, local programmers have taken existing software and modified the language. Expanding features is a far more ambitious and time-consuming task, and one that requires significant programming skill.

Open-source adaptations of this type can provide an outlet for local IT talent and may support one or two local service companies, but is not the foundation for an innovative industry capable of competing globally. Even if local open-source companies were to grow into innovative (as opposed to adaptive) firms, if they continued to follow the open-source model then their innovations would be freely copied by numerous global competitors. For example, if programmers in Peru wrote a useful new utility for the Linux operating system and licensed it under the GPL, Red Hat in the United States, Mandrake in France, SuSE in Germany,³⁵ and Red Flag in China, among many others worldwide, would all be able to include the new program in their Linux distributions. Thus, as is standard in the open-source model, the Peruvian programmers would be able to charge a positive price only for the service and support they provided, not to recoup expenditures for the effort involved in creating the new program. In this adjunct service market, they would face competition from all of the companies named above. It is unclear what competitive advantage the Peruvians would have over these other well-established global players. Even if they were able to survive as a profit-earning, and thus tax-paying, economy-supporting entity, one company does not create an industry—Peru would still be faced with spurring the growth of a larger network of software or IT firms.

At this point in the debate, open-source advocates often point to IBM as the epitome of a profitable firm with open-source offerings. In 2001, IBM spent \$1 billion (USD) backing Linux. In 2002, it announced that it had recouped this investment in full.³⁶ Certainly, IBM's highly visible support of open-source software has been profitable for the company. IBM is not, however, primarily a software company. It is a savvy services and hardware company that has successfully deployed Linux as a means to sell its services and hardware as well as its proprietary (*not* open-source) software. Thus IBM's experience does not provide developing nations with a road map to large financial rewards via open-source software. To even attempt this route to economic growth, countries would first need to foster a high tech hardware industry, along with a services and proprietary software industry. At a minimum, the developing nation would need to promote one successful hardware, services and/or proprietary

³⁵ It appears that despite its recent acquisition by Novell, SuSE will retain its brand and European focus.

³⁶ "IBM Revives the Mainframe," *Forbes.com*, 1/30/02, <http://www.forbes.com/2002/01/30/0130ibm.html>, downloaded 2/3/04.

software company. Open-source software, while it might be an enabling technology for certain hardware applications, is far from sufficient for this task.

C. Developing Nations Are Inherently Different

One of the more qualitative arguments made regarding developing nations is based on the belief that these countries are fundamentally different than economically developed countries and, as such, that they have a “moral” right to adopt different policies. This argument frequently appears in debates over intellectual property rights—a policy area that goes hand-in-hand with knowledge-based products like software. During the 1800s, for instance, the United States took such a stance in an effort to boost local writers and publishers and refused to acknowledge foreign authors’ copyrights.³⁷ Present day arguments run along similar lines: local citizens cannot afford expensive Western medicines and thus should not have to abide by drug patents; local computer users cannot afford expensive licenses and thus local governments should not have to enforce piracy rules. Not surprisingly, these are contentious issues with developed nations.

Proponents note that open-source software can allow a developing nation to sidestep some of these intellectual property fights. Licensing fees for open-source software, when charged at all, are much smaller than for proprietary software. Moreover, open-source encourages copying and sharing, so piracy is not much of an issue. Thus, advocates point out, encouraging open-source software enables a country to develop its local software industry without having to tackle thorny intellectual property rights issues.

Certainly some aspects of this argument are valid, but it too misses a fundamental point. That open-source software can avoid some intellectual property issues says nothing about its ability to promote economic growth. If the case for governments using open-source as a means to create a local software industry were well established, then this argument might provide additional reasons for governments to pursue the policy. As the examinations above make

³⁷ Niranjan Rajani, *Free as in Education, Significance of the Free/Libre and Open Source Software for Developing Countries*, for the Ministry of Foreign Affairs, Finland, May 2003, available at <http://www.maailma.kaapeli.fi/FLOSSReport1.0.html> (downloaded 4/27/04). See also, David S. Evans, *Who Owns Ideas? The War Over Global Intellectual Property*, FOREIGN AFFAIRS, November/December 2002.

clear, though, the case for using open-source as a means of economic development is not established at all.

Note that while piracy rules generally do not apply to open-source software, open-source licenses do rely on copyright laws for their enforcement, so this intellectual property issue cannot be avoided. Other intellectual property issues, such as patent laws or trade secret rules, can provide important incentives for local companies to invest in R&D and innovate.³⁸ If countries focus exclusively on the money flowing out to foreign software companies, without considering local constituents, they may establish policies that hinder local producers in the long run.

D. Open-Source Software Develops Local Programming Talent

This argument is related to much of the reasoning above: using local programmers for open-source projects promotes the diffusion of knowledge far more than proprietary software does because open-source grants free access to information. Once programming techniques are learned, they can be applied to any manner of software production and thus could lead to a sustainable software industry. The UN, in the quote cited in the introduction, makes such an argument, drawing a connection between open-source use, IT talent retention, and a nascent local software industry.

Unfortunately, this connection is weak, although some of the logic is sound. For instance, government sponsored open-source initiatives are likely a good labor training device because they can provide opportunities for basic programming in a (virtual) team environment. With all of the source code available and the global feedback available on open-source projects over the internet, novice programmers could gain valuable learning-by-doing experience.³⁹ That said, it is difficult to see the next link in the chain of logic. It is unclear whether this kind of training opportunity would result in a sizable pool of programmer talent, or whether it would

³⁸ For a discussion of the intersection between open-source software and software patents, see David S. Evans & Anne Layne-Farrar, *Software Patents and Open Source: The Battle over Intellectual Property Rights*, VIRGINIA J. L. TECH., forthcoming 2004.

lead instead to a small cadre of programming leaders surrounded by a larger pool of minimal contributors. This has been the norm with many open-source projects,⁴⁰ including the a-typical Linux development.⁴¹

Assume, for the sake of argument, that a significant number of local programmers do gain valuable programming expertise by working on government sponsored open-source projects. How can they employ their newly developed skills? It is unlikely that developing nation governments with limited resources can employ all of the new programmers at reasonable wages. Unless technology-using sectors, such as manufacturing, travel, finance, and the like, already exist locally, the domestic need for IT support will be minimal. Thus, the ability for the newly trained programmers to create local software and support companies is limited as well. Without a local proprietary software market to employ them, taking advantage of their skills will be difficult as local employment options will be limited. The new programmers can either leave the country for better-paid work in developed nations, thereby contributing to “brain drain” problems,⁴² or they can limit their programming activities to the smattering of paid local open-source opportunities.

In India, the reverse of this argument has occurred. The proprietary software industry has boomed, leading to a labor supply and demand mismatch. The government and private institutes alike have been increasing investments in engineering colleges with an emphasis on information technology in order to increase the local supply of well-trained (and well-

³⁹ This assumes that the nation already has in place basic infrastructure, such as a reliable power grid and telecommunications services. These fundamentals must be in place before computers will be of much use to either the government or its constituents.

⁴⁰ In fact, the typical open source project may be even smaller. In an empirical paper examining a sample of 100 open source projects posted on Sourceforge, Kieran Healy and Alan Schussman find that median number of software developers for open source programs is one person! The maximum number of developers in their sample is only 5. Kieran Healy & Alan Schussman, *The Ecology of Open-Source Software Development*, Working Paper, January 14, 2003.

⁴¹ George Dafermos, *Management and Virtual Decentralized Networks*, 6 FIRST MONDAY 11, 5 November 2001.

⁴² This is a significant problem in India, which has “an average annual attrition rate of 18-22 percent in the Indian market with a higher share for domestic enterprises and small export firms than for the multinationals.” Rothboeck & Gayathri, *supra* note 15.

credentialed⁴³) talent. Because the proprietary software industry has been one of the fastest growing sectors, “accounting for over 7 percent of the growth of its GDP,”⁴⁴ resources have been available to expand educational initiatives.

V. CONCLUSION

The case for open-source software as a growth and development tool is weak. Often, the arguments muddle reasons for using the software with reasons the software might promote economic growth. The low initial cost of open-source software, the freedom it affords from Western-based companies, and the opportunities it can provide for local programmers may well be valid points, but none of them speak to the ability of open-source to spur economic growth or even to its ability to establish a viable local software industry.

The most prominent example in the developing world of a newly emergent software industry is India, where open-source played no role. The rapid growth of India’s software exports, which comprise 70% of its software industry, can be attributed to its comparative advantage in labor. India has a large reserve of well-qualified, English speaking engineers and technicians that it has parlayed into outsourced proprietary software production for mostly Western clients. These particular circumstances raise the question of whether India’s experience can be replicated elsewhere among developing countries. Only a few developing countries, such as Russia and China, have a larger reserve of engineers than India.⁴⁵ These countries have other disadvantages, though, such as a lack of international language skills.

At a more fundamental level, it is unclear whether a national software industry (if it can be created) can help spur general economic development. Even in the world’s largest software producing nation, the United States, the link between the tech sector and general productivity growth are much disputed. The best to be hoped for from government support of open-source

⁴³ In selling their services to developed nations, Indian software firms have relied heavily on visible signs of employee talent, most notably touting the high number of workers with advanced engineering degrees. Arora & Athreye, *supra* note 14. Open-source “training” does not result in as visible a signal of achievement, although project contribution records sometimes can fill that purpose.

⁴⁴ Arora & Athreye, *supra* note 14.

⁴⁵ Arora & Athreye, *supra* note 14

software, is that it is an enabling technology. The pure open-source model is not capable of supporting for-profit firms. While the service-support model can provide sustainable profits, as the U.S. experience has demonstrated this model can only support a handful of firms at best. Moreover, the support model requires a broad open-source user market, more than one government alone can provide. Proprietary software applications designed to run on open-source software appear to be the most viable profit-making option, ala IBM. However, if creating profitable hardware, service, and support companies is the ultimate goal, then it implies a much different policy program. Governments supporting open-source software at the expense of proprietary software will not aid the creation of this kind of growth, and may even hinder its development.

Unfortunately, there is no panacea for economic development. While developing nations' interest in open-source is understandable, given its low investment costs and the overall appeal of software, the open-source model on its own does not appear to provide a solid foundation for profitable business operations that can meaningfully contribute to a nations' economic growth.