stances continue to evolve. Mueller has helped people to think about such challenges, but the theory of Internet governance seems too immature for it to yield reliable predictions.

Marjory S. Blumenthal (blumentm@ georgetown.edu) is associate provost (academic) of Georgetown University.

Is open-source software the answer?

The Comingled Code: Open Source and Economic Development

by Josh Lerner and Mark Schankerman. Cambridge, MA: MIT Press, 2010, 264 pp.

Pamela Samuelson

Open-source software has been an enigma to many commentators. Economists, for example, have pondered why any rational person would invest time and energy developing computer programs in order to make them freely copyable and modifiable, when developing proprietary software seems so much more likely to be lucrative. Are there reputational advantages or improved prospects for subsequent consulting gigs that might explain these otherwise irrational investments?

Even more puzzling perhaps is why major firms such as IBM, Hewlett-Packard, and Apple allow (or even encourage) their employees to participate in open-source projects and sometimes even dedicate previously proprietary software to open-source projects.

Leading software industry figures, among them Richard Stallman and Bill Gates, sometimes view open-source software in almost Manichean terms: Stallman, the founder of the Free Soft-

ware Foundation, views open-source software as inherently good and proprietary software as evil, whereas Gates has sometimes viewed open-source software as a form of intellectual property communism (that's not a compliment) in contrast with proprietary software, which is an admirable capitalist enterprise.

Leaving aside moral considerations, some regard open-source software as inherently better than proprietary software in technical terms because users can alter the underlying code to correct glitches or improve performance. The conventional wisdom also has it that open-source software is always cheaper than proprietary software.

The belief that open-source software is technically better, as well as cheaper, than proprietary software has led some developing countries to consider subsidizing the development of open-source projects or giving preference to open-source over proprietary software in procurements. Some have suggested that developing countries should adopt open-source software in order to promote economic development. The argument is that local engineers may attain more expertise by working with open-source software and then become better developers of software and related technologies thereafter.

But is open-source software really better and cheaper? Should developing countries subsidize or favor open-source over proprietary software? Is the software industry really bifurcated between the open-source and proprietary models, as the Manichean debate suggests? How do firms make money when developing open-source software?

We have Josh Lerner and Mark Schankerman to thank for undertaking a study of the software industry to produce the data necessary to address these questions with something other than philosophical assumptions. Lerner
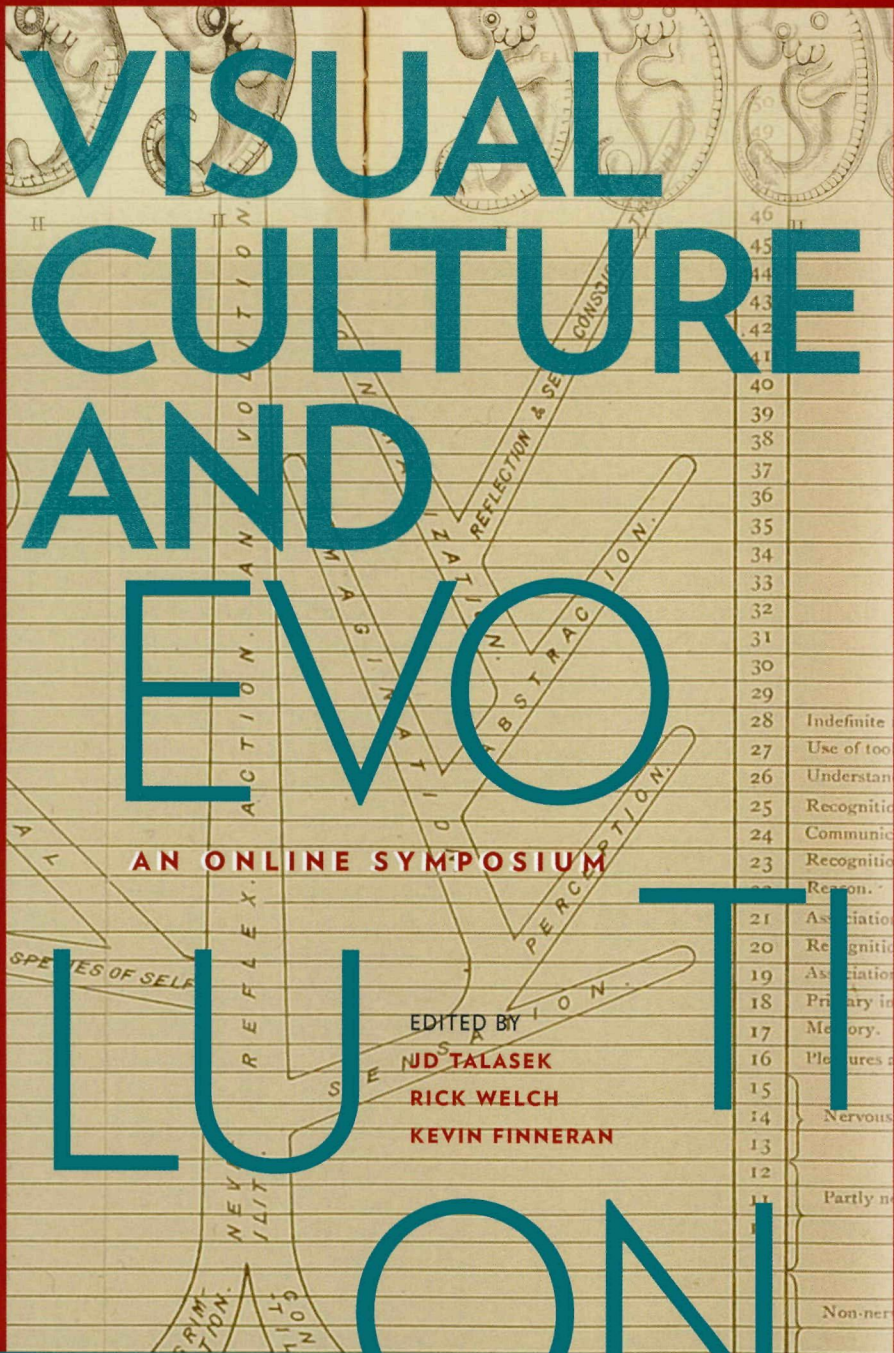
and Schankerman conducted surveys of software developers from 15 countries to gain insights about the software industry that might shed some light on the roles that open-source and proprietary software play in this industry today. Some of these countries had been the subject of previous case studies by these authors; they represent a mixture of developed- and developing-country economies. *The Comingled Code* reports on the results of these surveys and analyzes the implications of their findings for policymakers.

Almost 2,000 software developers responded to their survey. A substantial majority (62%) were small companies, just under a third were medium-sized, and just under 5% were large firms. The business models of these firms varied: Just under one-quarter were software service providers, and another one-quarter developed customized software; 14% developed software for bundling with hardware; and just under 40% developed software products for a more general market.

One important finding was that 40% of these firms do some open-source development. An interesting sub-finding is that a higher percentage of large firms than small firms (53% as compared with 38%) did open-source development. Only 15% of the firms, however, devoted more than 75% of development hours to open-source projects.

Open-source development was most common among firms that provided software support services, followed closely by customizers and bundlers, but nearly a third of the firms that were pure software developers did some open-source projects as well. It is thus much more common than has conventionally been assumed that firms do both open-source and proprietary development.

One reason to do both is that firms benefit from positive publicity that

VISUAL CULTURE AND EVO LU TI ON

AN ONLINE SYMPOSIUM

EDITED BY

JD TALASEK

RICK WELCH

KEVIN FINNERAN

Visual Culture and Evolution: An Online Symposium

Published by
Center for Art Design and Visual Culture at the University of Maryland, Baltimore County

Distributed by
Distributed Art Publishers, New York - artbook.com

This volume documents an online symposium held in April 2010 that was hosted by Cultural Programs of the National Academy of Sciences, Center for Art, Design, and Visual Culture at the University of Maryland, Baltimore County, and Johns Hopkins University's Master of Arts in Museum Studies Program.

A distinguished panel of artists, curators, scientists, historians, educators, media theorists, and critics participated in a lively, informative and synergistic discourse at the interface of art and science. Opening interviews by famed bio-artist Eduardo Kac and Pulitzer Prize-winning biologist E. O. Wilson set the stage for the symposium.

If ever there was a subject of controversy today that is in need of communication and unification across the disciplines, it is "evolution."

COMPETITION BETWEEN OPEN-SOURCE AND PROPRIETARY SOFTWARE IS HEALTHY, THE AUTHORS SAY, AND DECISIONS ABOUT WHICH TYPE OF SOFTWARE SHOULD BE PURCHASED SHOULD BE MADE ON A CASE-BY-CASE BASIS.

flows from their contributions to open-source projects. Some mixed-model firms offer one version of their software on an open-source basis and another with more-advanced features on a proprietary basis. In some cases, the same software is made available under both open-source and proprietary licenses. Lerner and Schankerman surmise that there are sometimes development cost and marketing synergies that can be achieved through a mixed-license model for software. Proprietary firms, however, export software more frequently than open-source developers, seemingly because they are better situated to recoup large sunk costs associated with exports.

Lerner and Schankerman also conducted a survey of software users to determine what factors affect their decisions about purchasing software on an open-source or proprietary basis. They gathered data from more than 2,300 firms from the same 15 countries. Just under 40% were small firms, a comparable percent were medium-sized, and just over 20% were large firms. Seven of 10 were domestic companies, 11% were foreign subsidiaries, and the rest were government agencies. The firms were a mixture of high-tech and low-tech manufacturers, wholesale and retail traders, and service providers.

They found that only about 60% of the surveyed users undertook a total cost of ownership (TCO) assessment, which accounts for procurement, man-

agement, support, associated hardware, and changeover costs, before purchasing software. Unsurprisingly perhaps, large-firm users were more likely to do a TCO analysis than were smaller firms, and high-tech firms are more likely than low-tech firms to do this. Government agencies do TCO analyses less often than companies do.
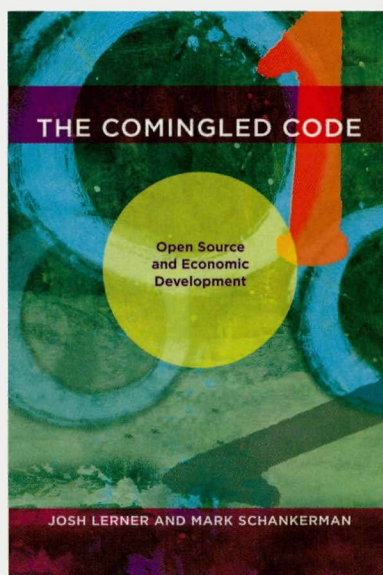
Of the users who reported doing TCO assessments, the initial cost of the software was, as you might expect, the most important factor. However, the availability of support services, the costs of switching (especially learning costs), interoperability, and the availability of upgrades were also considerations. This was true for users choosing to acquire open-source as well as proprietary software. Open-source

software may be cheaper initially, but a TCO analysis will sometimes show that it will be more expensive than proprietary software in the long run. Unsurprising perhaps is the finding that those who do not perform TCO assessments "appear to overestimate the importance of the initial cost of software but underestimate the importance of costs associated with interoperability and support services."

Although more than one-quarter of the respondents reported acquiring both proprietary and open-source software, there was considerable variation among users by country in this respect. Users from , , , , and reported much higher mixing of open-source and proprietary software (one-third or more of respondents did so). In contrast, at least 80% of firms in , , and reported using only proprietary software.

Lerner and Schankerman recommend that developing countries not favor open source in procurement policies nor offer special incentives to open-source developers. They should instead "adopt a neutral attitude that favors neither proprietary nor open-source software." Competition between open-source and proprietary software is healthy, they say, and decisions about which type of software should be purchased should be made on a case-by-case basis. Their data show that there is already a considerable amount of comingling of open-source and proprietary software, both by developers and by users, and this is likely to continue.

The one affirmative policy recommendation Lerner and Schankerman make is for governments to encourage the development of open standards that any software developer can implement on reasonable and nondiscriminatory terms. Both open-source and proprietary developers will benefit from the adoption of such standards.

The Comingled Code reports exceptionally interesting findings and offers many insights about the software industry and the prevalence of open-source and proprietary software in various sectors, from both the supply and demand sides. I would recommend it to anyone interested in the software industry, particularly for the data it provides about variations in practices among the 15 countries.

It is, however, an odd assortment of countries on which they report: Why Greece rather than Finland? Brazil and Chile, but not Argentina or Colombia? Kenya and South Africa, but not Morocco or Nigeria? The data they collected are highly informative, but given the variations among countries on many issues, it is hard to know how much one can generalize from the findings, apart from the policy prescription of neutrality, which one expects Lerner and Schankerman would favor in any event.

By the end of this book, one hungers for a similar survey to be conducted of U.S. software developers and users and a comparable report to be written. Code is comingled in the United States, but how is the market similar to and different from the markets on which Lerner and Schankerman report? One can only hope that this will be their next project, because the differences could be interesting. I would guess, for instance, that the comingling of open-source and proprietary software is even more common among U.S. than Turkish users but that TCO analyses are probably about as common in the United States as in the countries studied.

Some myths about open source may have been dispelled by this book, but perhaps not all. In particular, the book does not offer as much insight about the authors' views about the value of open source in promoting economic development in the countries surveyed as the title of the book suggests. But perhaps that will be a future topic of research for these researchers or others who follow in their wake.

Pamela Samuelson (pam@law.berkeley.edu) is the Richard M. Sherman Distinguished Professor of Law and Information at the University of California, Berkeley, and a director of the Berkeley Center for Law & Technology.