

---

# Open-Source Software: Who Needs Intellectual Property?

BY MICHELE BOLDRIN AND DAVID K. LEVINE

---

The market for open-source software—uncopyrighted, freely reproducible computer programs—is not well understood by economists. A central source of surprise is that innovation can thrive in a market without traditional intellectual property (IP). But as we argued in a 2005 unpublished paper, “Perfectly Competitive Innovation,” as a matter of theory there is no reason to believe that monopoly power through IP is needed for innovation. The market for open-source software is the poster child for this perspective.

First, understand that the market for open-source software is a classic example of a competitive market. It is characterized by the voluntary renunciation of copyright and patent. Buyers are entitled to make their own copies, modified or not, and sell them. “Free software” in this context means “free as in freedom, not free as in beer.” There is also voluntary renunciation of trade secrecy: the original creator publishes the source code—the “blueprint” for producing the software—along with the software itself. Some open-source software has the further requirement that as a condition of use, buyers make their modification available under the same terms. The open-source movement has been called everything from a virus to socialism—so it may or may not be surprising to hear it called a model of a fully competitive market. Yet that is what it is, as much so as the market for wheat. All purchasers of software can compete with the seller and one another, and often they do.

Given that there are fixed costs of producing software and (it is commonly thought) competition drives profits to zero, how does this market function? How are the fixed costs covered? In the absence of profits from monopoly power, the source of income used to pay

fixed costs is competitive returns. We will investigate three issues here. First, what is the source of the competitive returns that pay the bills of software developers? Second, is the market a real market? That is, do software producers in fact get adequate compensation for the fixed costs of their efforts? Or is open-source software, as is sometimes alleged, simply an elaborate altruistic charity? Finally, we ask how significant the open-source software market is. Is it a thriving source of innovation, or just a free-rider off the innovations of more traditional closed-source IP-protected software, making cheap imitations that never would have been produced in the first place absent monopoly power?

The evidence (and the common sense of anyone involved with open-source software) shows that the source of competitive returns that pay the bills of software developers is the complementary sale of expertise. To earn a return through the sale of something, it must be something scarce. Copies of software may be scarce, but we shall see that the actual duplication of copies is sufficiently quick and inexpensive that only small returns can be obtained through the sale of copies. However, purchasers of copies of software programs also have a demand for services—ranging from support and consulting to customization. They naturally prefer to hire the creators of the programs, who in the process of writing the software have developed specialized expertise that is not so easily matched by imitators.

---

*Michele Boldrin (micheleboldrin@gmail.com) and David Levine (david@dklevine.com) are professors of economics at Washington University in St. Louis. Boldrin is co-editor of the Review of Economic Dynamics. Levine is co-editor of Econometrica.*

To understand the sources of competitive returns in this market, it is helpful to look at an example. A leading and profitable open-source software firm is Red Hat, a company that sells the computer operating system GNU/Linux, a modified and customized version of the underlying Linux system, with many features that can be optionally installed. Although the base system is in principle obtained by Red Hat for free, in fact it pays the developers. Alan Cox, one of the main kernel developers, works for Red Hat. The firm is also a contributor to the Open Source Development Labs, which employs Linus Torvalds, the developer of Linux. In addition Torvalds benefited from a substantial “gift” of stock options from Red Hat. Beyond this, the customization and testing conducted by Red Hat is costly. So Red Hat faces a substantial fixed cost of providing its software.

Let us first consider returns earned through the sale of physical copies. In this market physical copies of software sell for greater than marginal cost. Using prices quoted on the Internet on July 10, 2002, Red Hat charged \$59.95 for a package containing its system. Because it is based on Linux, competitors can legally duplicate and sell the exact same Red Hat system. In fact, at least two companies, Hcidesign and Linuxemporium, did exactly this. On July 10, 2002, Hcidesign offered for sale Red Hat Linux 7.2 for \$16, about a third the price charged by Red Hat. Linuxemporium.co.uk offered a similar deal. The striking fact is that Red Hat sold many more packages at \$59.95 than Hcidesign and Linuxemporium did at \$16. Neither company ever represented a threat to Red Hat.

Notice that the premium charged by Red Hat was not likely due to the physical scarcity of copies. Rather, it was for the sale of expertise that came with developing the system. Anyone who uses computer software knows that it rarely functions as expected. If you buy software and have a question or problem, whom would you prefer to call? The people who wrote and developed the program? Or the people who duplicated the CD?

## A Better Business Model

In fact, selling expertise by charging a premium on physical copies has not turned out to be the most successful business model. Red Hat eventually concluded that it was not selling enough \$59.95 copies and switched strategies. What had previously been sold is now given away for free as Fedora Core and is used as a platform to get feedback on features that are incorporated into the commercial system called Red Hat Enterprise Linux. The latter is available only by annual subscription at a price that—depending on features—ranges from \$349 to \$2,499.

This blurb from the Red Hat website’s promotional material makes clear what customers are paying for: “Unlimited access to service and support: Subscriptions

include ongoing service and support to guarantee your systems remain secure, reliable, and up-to-date. When you have a technical question, you’ll speak to Red Hat Certified Software Engineers. Or you can access a self-serve knowledge base of technical information and updates.”

Notice how this market works: First expertise is passed from the developers to Red Hat Certified Software Engineers. Then others acquire the expertise, the stock of expertise expands, and the price at which it can be sold drops.

Of course, in the meantime innovations are created, and new expertise is generated.

The presence of profitable firms such as Red Hat—not to speak of IBM—in the open-source industry suggests that it is a viable concern and not a charitable or altruistic activity. In their 2004 paper “The Economics of Technology Sharing: Open Source and Beyond,” Josh Lerner and Jean Tirole documented some of the financial benefit to individual developers of contributing to open-source projects. For example, the team of programmers that developed the Apache web server are ranked according to the significance of their contributions and hold other jobs. Work by Il-Horn Hann et al. shows that the salaries the programmers receive in these other jobs are heavily influenced by their rank within the Apache Foundation. In other words, the “expertise”

---

The evidence shows that the source of competitive returns that pay the bills of software developers is the complementary sale of expertise.

---

model at Apache is much like that in academia—the programmer writes software in order to receive recognition and financial payment for the expertise he demonstrates through his published product.

Examination of particular individual developers reinforces this point. Torvalds is a multimillionaire, and Bram Cohen, the developer of BitTorrent, recently received \$8.75 million in venture capital for his open-source project. These figures and the success of open-source software also teach us something important about the (expected) payments needed to get smart people like Torvalds or Cohen to develop innovative software. It is unlikely that Torvalds originally wrote Linux with the aim of becoming a multimillionaire. Still, he must have hoped for some revenue stream when starting his work. His current wealth is probably higher than he expected. Still it is four orders of magnitude less than that of Bill Gates. Hence, at least in the case of Torvalds, the opportunity cost for writing innovative software is not in the tens of billion of dollars, but just in the millions. This is worth keeping in mind when someone claims that without the huge monopoly rents through IP, innovators would not be innovating.

Finally, it is possible to imagine that the open-source industry is not a real industry at all. Perhaps it exists only because it is able to free-ride off the innovations created in the proprietary part of the industry, in which the monopoly power of copyright plays a key role. It is certainly true that Linux is a knock-off of Unix and that OpenOffice Writer is a knock-off of Microsoft Word. But this means little, because practically all software, proprietary or not, is an imitation of some other software. Microsoft Windows is an imitation of the Macintosh, which is an imitation of Smalltalk. Microsoft Word is an imitation of WordPerfect, which is an imitation of WordStar. Microsoft Excel is an imitation of Lotus 1-2-3, which was an imitation of VisiCalc. And so on.

But is the free-rider argument correct?

Consider word-processing. Many open-source alternatives to Microsoft Word exist, including Kword, AbiWord, and OpenOffice Writer, the latter being the most widely used. How did the cost of developing this soft-

ware—financed as it was by an open-source model—compare to the cost of developing Microsoft Word? The fact is that most of the cost of writing software is not in the observation that it might be nice to have a button to justify text or in the algorithms for spacing lines—which after all were developed by Gutenberg back in 1450—but rather in the detailed implementation and debugging of computer code. As far as we know, none of these open-source projects benefited at all from the work done by Microsoft.

### More Expensive to Develop

Indeed, the development of these open-source projects was probably more expensive than the development of Microsoft Word. The most difficult and expensive programming tasks faced by the developers of these projects appear to be reverse engineering of Microsoft Word documents and providing compatible formatting capability so that documents can be exchanged with Microsoft Word. Had these projects gone first, this substantial cost would have been avoided.

It is also worth noting that the competitive returns generated by these projects are significantly smaller than they would have been had they hit the market before Microsoft Word did. So it seems reasonable to conclude that the competitive market without Microsoft Word would have delivered both these programs.

Probably the most innovative program in the last few years is BitTorrent, a program that decentralizes and vastly increases the speed at which very large files can be downloaded off the Internet. It is commercially successful in the sense that 50,000 copies a day are downloaded. It is also sufficiently innovative that it is now being imitated—by Microsoft. BitTorrent, however, is open-source, and according to its website, author Bram Cohen maintains the program for a living.

The final point to emphasize here is that the market for software is not unique. Innovation and competition unprotected by patent and copyright have gone hand in hand in other industries, from financial securities to fashion. The message of open-source software is a message for all industries: IP not needed for innovation here. 