

Why Open Source Works

Author Steven Weber looks beyond the hype on Open Source. More than a self-governing utopia, it's a practical, sustainable way of organizing and innovating. Its methods may soon be applied successfully in other sectors. Plus, a "crazy" idea for Microsoft.

Steven Weber, author of the new book "The Success of Open Source" (Harvard University Press, April 30, 2004) is a political scientist at the University of California, Berkeley. His specialties include the political economy of global economic change and the open source software process. He is also a consultant with the Global Business Network and a former adviser to the president of the European Bank for Reconstruction and Development.

UBIQUITY: How is your book doing so far?

WEBER: Quite nicely. The first printing of 2,500 copies is already gone.

UBIQUITY: A good sign indeed. What made you decide to write about open source?

WEBER: Many people don't understand what the open source movement is really about, so I'd like to get folks past the hype so they can see more clearly the profound innovation at play. Another purpose is to tweak people into thinking about what are the bigger implications of this phenomenon for the IT market as well as other sectors of the economy.

UBIQUITY: Before we talk about the book, let's talk a bit about you. Tell us a little about your background and interests.

WEBER: Like many confused 20 year olds, when I graduated college I didn't know what to do with my life. I moved to California to go to medical school at Stanford. I liked it at first but towards the end I learned that it was really about being around sick people all the time. So I switched out of the hard sciences and went into political science. It felt more intellectually challenging for me. Ironically, very few of the people from my medical school class are still practicing medicine. Many of us dropped out and did other things.

UBIQUITY: Why is that?

WEBER: It was partly because of the particular intellectual climate around Stanford at that time. It was a research-oriented climate rather than patient care-oriented and so many people went into the research end. Other people went into the business end. It was also the beginning of the managed care revolution that made it difficult for practicing physicians to deal with their patients the way they thought they were going to when they started school.

UBIQUITY: What was your interest in political science?

WEBER: I've always been interested in the circumstances under which non-authoritative environments can reach and sustain collaborative equilibrium, which is a fancy way of saying, "How do people work together when there's no authority around?" I've looked at large-scale non-hierarchical collective action problems in a number of settings.

UBIQUITY: What led you to study the open source community?

WEBER: I discovered the open source community about four or five years ago. My original thinking was that this group of people was solving familiar governance problems in an unfamiliar way. I was interested in the question, how does the community govern itself? How do the members set up rules? How do they maintain norms? How do they sustain collaboration over time? And most important, why would individuals contribute to this joint product from which they don't get any directly controllable economic reward? That is the first question that many people ask. Why do programmers contribute their time and mind space and energy to this thing?

UBIQUITY: Let me read you a sentence from your book. It says, "People often see in the open-source software movement the politics that they would like to see -- a libertarian reverie, a perfect meritocracy, a utopian 'gift culture' that celebrates an economics of abundance instead of scarcity, a 'virtual' or electronic existence of proof of communitarian ideals, a political movement aimed at replacing obsolete 19th century capitalist structures with new 'relations of production' more suited to the information age."

WEBER: I'll say this carefully because it's important to put the right tone on it. When Eric Raymond wrote that very well known paper, "The Cathedral and the Bazaar" he did something extremely important and he also did something misleading. The extremely important thing he did was to create a self-conscience statement to the open source developer community. "Here's what we are. Here's what we're doing

collectively." That was important because it made explicit among this group a sense that they were doing something unusual.

UBIQUITY: What was misleading about Raymond's paper?

WEBER: What I think he did that was misleading was to use the image of a bazaar. If you read the paper carefully, he describes a magic bubbling cauldron that somehow comes together into a complex operating system called Linux. He didn't have a good powerful explanation for how the community holds itself together. People saw in that what they wanted to see. That's why it became a kind of Rorschach test.

UBIQUITY: How is the open source community like a Rorschach test?

WEBER: People tend to look at the surface and see what they want to see in the picture. If you believe that self-organizing communities are good things, then you'll look at formally organized communities and see that they have lots of problems and that their bureaucracies are very inefficient and expensive. If you don't look closely enough at the open source community, you think, "Wouldn't it be great if the world were organized like that?"

UBIQUITY: What's the sustainability of the open-source idea?

WEBER: One of the reasons I wanted to call my book, "The Success of Open Source" is because I think the challenge to the community has always been, can you sustain the level of collaboration that exists in a big project over time and scale? Many folks inside the community have

been surprised by the degree to which the Linux system has demonstrated sustainability over time and the ability to scale to involve large numbers of people -- although, of course, at different levels of effort.

UBIQUITY: What is needed to sustain an open source organization over time?

WEBER: It's not easy. It requires organizational innovation. In particular, Linux has innovated in interesting ways that help to sustain it over time. There have been many failures and that's fine, too. I see these folks as developing a new way of organizing. They're figuring out the governance structures over time through a process of trial and error.

UBIQUITY: What is the core of the open-source idea in your opinion?

WEBER: The core of the open-source idea is the GPL, the General Public License. It says, essentially, that we're going to put an intellectual product out for people to use, improve, customize, fuse with and do whatever they want. The single effective restraint is that anything they do with it, they need to give back to everybody so that as a whole, the community can select and incorporate the innovations, changes, and modifications that upgrade the common product in such a way that nobody can individually benefit without everybody benefiting. There are twists and turns, details that change the precise implementation of that core idea, but the core is there.

UBIQUITY: What do you think of the idea of open content?

WEBER: Wikipedia, an online encyclopedia that anyone can contribute to, is often thought of as an open-source style project. It's open-source in the sense that it taps into the same voluntary motivations that the open-source community tries to tap into but it's not open-source in the sense that the property rights are configured as they are within an open source software community. I can take something off of Wikipedia and essentially privatize it. I can improve on it and change it but then I can keep it to myself. The point: not all voluntary organizations are "open source". They may be very good things (or bad things, I'm agnostic about voluntarism as a human motivation) but they are different than open source.

UBIQUITY: Have you used Wikipedia yourself?

WEBER: I've looked at it, yes. Some of it's good and some of it's bad, exactly as you would expect. There's no editing function, per se.

UBIQUITY: Isn't that troublesome?

WEBER: Yes. It's very troublesome. I don't mean to diminish its importance because there are many things in the world that can happen simply by people making contributions to a joint product. I often use an analogy to try to explain this. Why is it that most poetry gets written by one person? The reason why there's not a lot of collective poetry is not because you couldn't get people to contribute words. The reason is because if everybody just contributed words, you wouldn't have a poem. You would have a mess.

UBIQUITY: What prevents open-source code from becoming a mess?

WEBER: Open-source has a governance structure and an editing function so that the code, over time, works together. That's what things such as Wikipedia lack. It is a naive version of open-source, which in some cases works. It's more like barn raising. If you get enough people together, you can all lift the barn but if you get enough people together, you can't blow away a complex operating system for a modern microprocessor. Unless, of course, you organize their work in an extremely sophisticated fashion.

UBIQUITY: Everyone's heard of Fred Brooks and his famous essay, "The Mythical Man-Month." Talk about the open-source movement in relation to his basic idea.

WEBER: Brooks' basic idea is that because of the complexities of human communication, as you add layers onto a bureaucratically organized project, the work that gets done scales linearly but the vulnerability to mistakes scales geometrically. As he put it, if you add additional labor to a software project that's already behind, it will only make it further behind. Bureaucracies are remarkably inefficient in organizing human labor -- we've used them so much only because they are the 'least bad' way of organizing that we've known. In the software world, it's particularly problematic because of the complexity of the way in which software works together on many different levels. It's inevitable that there will be mistakes and bugs. The hard question for the open-source software community is not how to transcend Brooks' law, because no one can really do that yet. It is how to

diminish the practical effects of Brooks' law. And make the software development process a little bit less inefficient.

UBIQUITY: How does the open source community mitigate Brook's law?

WEBER: By opening the code up to essentially unlimited peer review, they take advantage of a law of large numbers. Eric Raymond said, "With enough eyes all bugs are shallow." If enough people look at the code, it's likely that someone will see a mistake or someone will have a better idea of how to fix it. The open-source community's bet is to distribute the choices across the community as a whole and assume that the best choice will eventually survive in an evolutionary primordial soup competition to get to the top.

UBIQUITY: How is the process different for the proprietary software community?

WEBER: The proprietary software community has placed a bet that the best way to get around Brooks' law is to have things work their way up in a command-and-control hierarchy toward one individual who'll eventually make an authoritative decision. It's hard to say which approach will prevail over time in the abstract, but clearly the open-source community way of doing things is reasonably successful.

UBIQUITY: Linux and Windows are two examples of open-source versus proprietary software development. Windows, theoretically, could become open-source tomorrow, right?

WEBER: Yes. Microsoft could do that. Microsoft could announce that it is releasing the source code for Windows under the GPL (don't hold your breath waiting for this, though) Netscape did this when they were losing market share to Internet Explorer. At a certain point, for a variety of reasons, Netscape made a huge bet and said, "Let's make the code open-source and see if we can accelerate the rate of innovation and generate a community of interest and developers around this code." It didn't work very well.

UBIQUITY: Why not?

WEBER: The bulk of the Netscape code was developed in-house in a proprietary setting under time pressure. By the time it was released to the community, it was such a mess that people couldn't understand what the hell was going on with it. Nobody felt a sense of ownership. I suspect that if Microsoft were to release the Windows code, it would be such an enormous patchwork of layers built up over time that it would be a mess. Developers would probably prefer to start over and do it anew. The more interesting and only slightly crazy idea for Microsoft would be to release their own version of Linux, Microsoft Linux, and customize Linux the way Red Hat and others have done. They could essentially become an open-source company and drop the Windows operating system entirely.

UBIQUITY: That sounds like a crazy idea. Would it work?

WEBER: In some ways it would be crazy but most 'revolutionary' business ideas are. This is essentially what IBM did with its Websphere software. They recognized that their Web server package was inferior

to the open-source package Apache. They dropped their server package, picked up Apache, became a member of the Apache software developer community and capitalized on what they are good at, which is integrating business processes, hardware, services and customization. It's been a very successful strategy for IBM. Microsoft could do something like that in a couple of years if Linux continues to improve at an accelerated rate and Windows continues to fumble.

UBIQUITY: As a betting man, what would you say are the chances of that happening?

WEBER: I think in five years most of us will look at the idea of a proprietary operating system as a quaint old-fashioned notion. There are fewer and fewer things that you can do on an expensive Sun box running Solaris that you can't do on a very cheap Intel box running Linux. It's hard for me to see what in the long-run saves the notion of proprietary operating systems except in some few, small mid-segments of the market other than the installed base. And the installed base deteriorates over time.

UBIQUITY: Let me ask you about "the Great Man Theory of history." Would there be an open-source movement without Linus Torvalds?

WEBER: Yes. There are hundreds of other projects, though none as large or as much in the public eye as Linux. There were other Unix clones or Unix-style operating systems developed in open-source fashion at the time that Linus started doing his.

UBIQUITY: Such as?

WEBER: There was BSD Unix, Berkeley Software Distribution, which was principally a product of the Computer Science Research Group at UC Berkeley. My Macintosh runs on OS X, whose Darwin core uses much of the BSD Unix code with a proprietary operating interface on top of it. Torvalds himself has said that if he'd known in 1991 when he was first looking for a Unix clone how much work had already been done on BSD Unix, he doubts he would have started Linux. Why remake the wheel?

UBIQUITY: Why didn't he know about it?

WEBER: It was before the World Wide Web. He was a college student in Finland and he didn't know what was happening here at Berkeley. It was the good old days before everybody knew what everybody else was doing.

UBIQUITY: What about the next good old days? What do you think will happen next?

WEBER: Open-source systems will continue to erode the market share of proprietary systems in increasing segments of the business market, for example, large-scale relational databases. Those in the very high-end of supercomputing, IBM, in particular, have become specialists at tying together Linux boxes. They pull old Pentium III machines out of the basement, put Linux on them, and tie them together with open-source architecture to create supercomputers for a fraction of the price of what it would cost to buy one from Cray.

UBIQUITY: What's happening at the low end?

WEBER: Tivo, for example, runs a version of the Linux operating system. If you want to, you can go to the Tivo website and download the code and fuss around with it. There's a PDA in the US market called the Sharp Zaurus that runs a Linux system. You will continue to see the migration of the open-source code into more enterprise operating systems across both the high- and low-ends of computing. Over the next couple of years you'll increasingly see Linux on the desktop, which has not been a popular choice so far largely because of the installed base and in part because the graphical user interface from Linux is not on the level of either the Windows or Mac graphical user interface.

UBIQUITY: Why is that?

WEBER: The people who work on Linux haven't had as much interest in developing user-friendly interfaces as they have in developing sophisticated back-end code. Second, it's harder to develop a user interface in an open-source manner because much of what you learn about user interface happens face-to-face watching people use it or with focus groups.

UBIQUITY: What do you think will happen when the people in organizations outside the IT world become more aware of the success of open source software development?

WEBER: I'm starting to see this already. You hear people say, "It's just software. It's just a bunch of hackers writing for hackers. It's

something unique to the culture of people who do this." That's not necessarily true. People are going to start asking, "Are there general principles of organization that are laid out in something like the general public license that could be applied to other industries?"

UBIQUITY: How do you explain it to them?

WEBER: I use an analogy. In the 1980s, Toyota started taking away market share from General Motors and Ford and others. The Toyota car seemed to be made in a way that was less expensive but with better quality. How was that possible? Everybody was fixating on the car. A bunch of researchers from MIT spent time in the factories where the car was being made and wrote a book called, "The Machine that Changed the World." They argued that by focusing on the car, people had missed the big picture. The important thing was the way in which Toyota made cars. The machine that changed the world was not a machine at all; it was a way of making things.

UBIQUITY: Can you think of a market outside of IT that would be a likely candidate to experiment with open source development?

WEBER: I talk about this some in the conclusion to the book. Being a betting man, I'm willing to bet that at some point in the next couple of years someone will try an open-source style experiment in the pharmaceutical industry.

UBIQUITY: How would that work?

WEBER: Here's the way I would propose it might work. Let's say a company like Eli Lilly has a molecule that they sell as Prozac. It has made a lot of money for a number of years. At some point it goes off patent, as it did a couple of years ago, which makes it subject to generic competition, and at that moment Lilly's profits from that molecule fall by 80 or so percent. Now imagine an alternative trajectory. Instead of trying to squeeze a sponge for the proprietary patent rents for that last year before the patent expires, what if a pharmaceutical company were to license the molecule under the GPL so that others could add to it, customize it, modify it, change the packaging, et cetera? Under the terms of the GPL, anything that they do has to be given back to the community. They can't privatize it, enclose it or sell it on their own.

Essentially, what that company would then be doing is the same as Torvalds and others do with the code they write, which is to give it away to everybody who thinks they can play with it and modify it. If someone customizes it and does something with it, that's great. But nobody can learn more about it than the original developers because everything has to be given back to the community that writes the code.

UBIQUITY: The obvious benefit to individuals would be better drugs. How would this benefit the pharmaceutical company?

WEBER: The pharmaceutical company would incorporate valuable changes or innovations. It would use its marketing channels, distribution channels, clinical trial expertise, and other resources to build a new business model around a different part of the value chain.

The real question is, is the rate of return that has existed in the software industry under the proprietary model likely to diminish in an open-source environment? It would be a more competitive market, which means that software tools will likely become somewhat less expensive. For the economy as a whole that's a really good thing.

UBIQUITY: What are you going to do next?

WEBER: I'm thinking about the pharmaceutical industry and maybe going back to my roots and training in medicine. The rate of innovation in drug discovery over the last ten years has declined substantially despite the fact that the big drug companies are pouring increasing amounts of research and development funds into trying to discover new molecules. The rate of new drug discovery has gone down substantially. I spend a lot of time thinking about how open-source innovation models could accelerate the rate of innovation in drug discovery and perhaps reduce the price of drugs. I think it's an interesting test of whether the open-source style of innovation model can extend to other sectors in the economy. So, in this space, there's money, there's the potential to do good things for human beings, and there's an interesting intellectual problem. What else could you want?

END

Source: Ubiquity, Volume 5, Issue 11, May 12 - 18, 2004,
<http://www.acm.org/ubiquity/>