

Innovation by User Communities: Learning from Open-Source Software

Creating complex products with limited manufacturer involvement is a growing phenomenon occurring in markets as diverse as windsurfing gear and open-source software.

Eric von Hippel

Imagine product development without manufacturers. Today's user innovation communities are making that idea increasingly real. Open-source software projects, among others, have led to innovation, development and consumption communities run completely by and for users. Such communities have a great advantage over the manufacturer-centered development systems that have been the mainstay of commerce for hundreds of years. Each using entity, whether an individual or a corporation, is able to create exactly what it wants without requiring a manufacturer to act as its agent. Individual users in a user innovation community do not have to develop everything they need on their own but can benefit from others' freely shared innovations.

Examples of User Innovation Communities

User innovation communities existed long before the advent of open-source software and extend far beyond it. They are not limited to information products such as software code. Some develop physical products. Consider and compare two examples of early-stage user innovation communities — one in software, the other in sports.

Apache Open-Source Software Apache open-source software is used on Web server computers that host Web pages and provide appropriate content as requested by Internet browsers. Such computers are the backbone of the World Wide Web.

The server software that evolved into Apache was developed by Rob McCool (at that time a University of Illinois

student) for, and while working at, the National Center for Supercomputing Applications (NCSA). The source code as developed and periodically modified by McCool was posted on the Web so that users at other sites could download, use, and further modify and develop it.

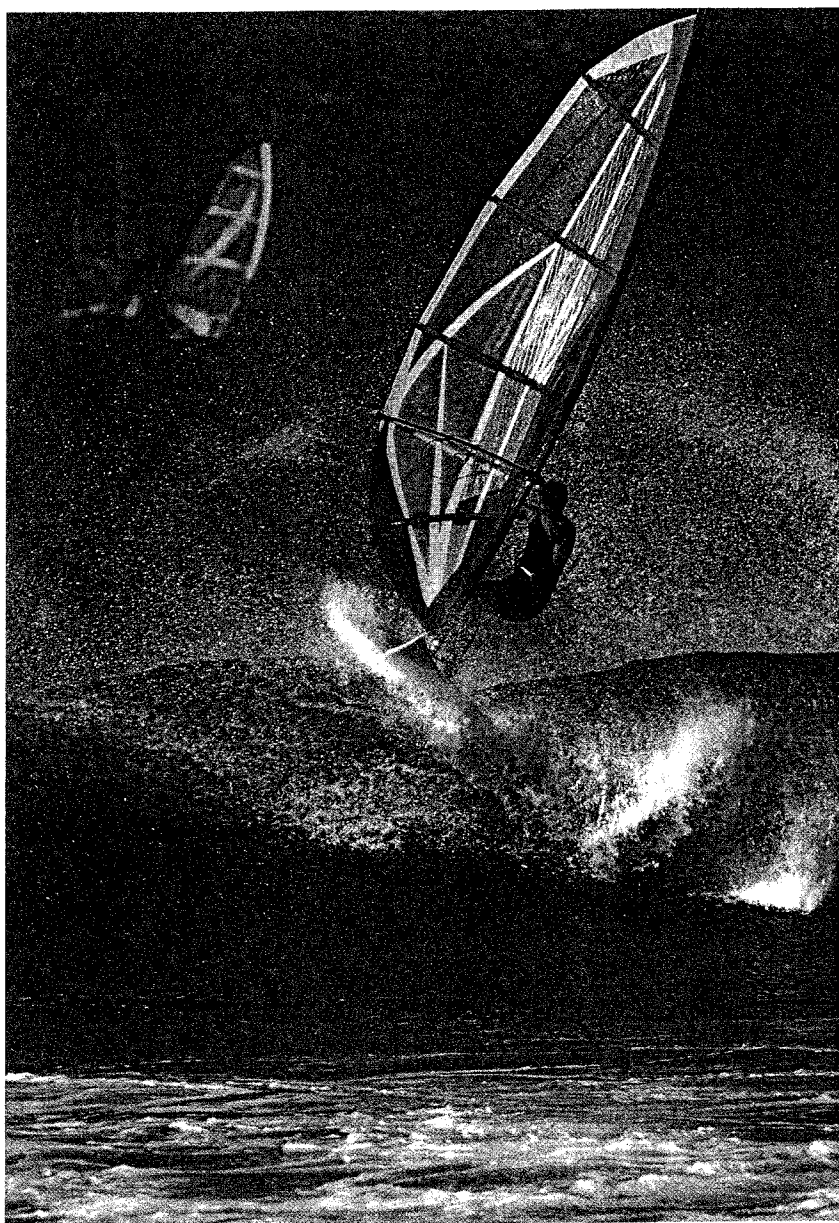
When McCool departed NCSA in mid-1994, a small group of webmasters who had adopted his server software for their own sites decided to continue developing it. Eight users gathered all the documentation and bug fixes and issued a consolidated patch. This "patchy" server software evolved over time into Apache. Extensive user feedback and modification yielded Apache 1.0, released on Dec. 1, 1995.

After four years and many modifications and improvements contributed by many users, Apache became the most popular Web server software on the Internet, garnering many industry awards for excellence. Despite strong competition from commercial software developers such as Microsoft Corp. and Netscape, it is currently used by approximately 60% of the millions of Web sites worldwide.

High-Performance Windsurfing High-performance windsurfing, the evolution of which was documented by MIT doctoral candidate Sonali Shah in a March 2000 MIT Sloan working paper (<http://opensource.mit.edu>), requires gear suitable for midair jumps and turns. Previously, windsurfers used their boards essentially as small, agile sailboats, and the boards' specifications reflected that.

The fundamentals of high-performance windsurfing were developed in 1978 in Hawaii by users. Larry Stanley, a pioneer in the sport, explained to Shah how a major innovation in technique and equipment came about.

"In 1978," Stanley recalled, "Jurgen Honscheid came over from West Germany for the first Hawaiian World Cup and discovered jumping, which was new to him, although Mike Horgan and I were jumping in 1974 and 1975. There was a new enthusiasm for jumping, and we were all trying to outdo each other by jumping higher and higher. The problem was that ... the riders flew off in



midair because there was no way to keep the board with you — and as a result, you hurt your feet, your legs and the board.

“Then I remembered the ‘Chip,’ a small, experimental board we had built with foot straps, and thought, ‘It’s dumb not to use this for jumping.’ That’s when I first started jumping with foot straps and discovering controlled flight. I could go so much faster than I ever thought and when you hit a wave it was like a motorcycle rider hitting a ramp; you just flew into the air. All of a sudden not only could you fly into the air, but you could land the thing and not only that, but you could change direction in the air!

“The whole sport of high-performance windsurfing really started from that. As soon as I did it, there were about 10 of us who sailed all the time together, and within one or two days there

were various boards out there that had foot straps of various kinds on them and we were all going fast and jumping waves.”

By 1998 more than a million people were engaged in windsurfing, and a large fraction of the boards sold incorporated the user-developed innovations for the high-performance sport.

Both of the user innovation communities just described have evolved and become more complex. Although they look different on the surface, they are similar in fundamental ways. Both grew to include many thousands of volunteer participants. Participants in open-source software projects interact primarily via the Internet, using various specialized Web sites set up by other volunteer users. (See “Free and Open-Source Software.”) Participants in innovating sports communities tend to interact by traveling to favorite sites and contests for their sport. Most users of open-source software simply use the source code, relying on interested volunteers to write new code, debug existing code, answer requests for help posted on Internet help sites, and help coordinate the project. Similarly, as Shah and Nikolaus Franke show in an April 2001 MIT Sloan working paper, most participants in an evolving sport simply play the game, relying on those so inclined to develop new techniques and equipment, try out and improve on innovations, voluntarily provide coaching and help coordinate group activities such as meets.

Often, commercial enterprises attach to or complement user innovation communities. Red Hat and VA Linux Systems are well-known exam-

ples of commercial involvement in the open-source software context; professional sports leagues and commercial producers of sports equipment are examples in the case of sports communities.

User Innovation Communities Shouldn’t Exist, But They Do

Manufacturers, not users, traditionally have been considered the most logical developers of innovative products for two reasons. First, the financial incentives to innovate have seemed higher for manufacturers than for users. After all, a manufacturer has the opportunity to sell to all users what it develops. Individual user-innovators, on the other hand, typically can expect to benefit financially only from their own use of their innovations. In order to benefit from diffusion of an innovation to the other users in a marketplace, innovating users would have to obtain

Free and Open-Source Software

When people say that software is “free” or “open source,” they mean that a user may obtain a copy at no cost and then legally study its source code, modify it and distribute it to others — also for free. A software author uses his or her own copyright to guarantee those rights to all users by affixing to the code a standard licensing notice, such as the General Public License (GPL), commonly referred to as “copyleft” (a play on the word “copyright”). Well-known examples of open-source software are the GNU/Linux computer operating system, Perl programming language and Internet e-mail engine sendmail.

The practice of granting extensive rights to users through licensing dates back to the free-software movement that Richard Stallman launched in the early 1980s. Stallman founded the Free Software Foundation to counter the trend toward proprietary development of software packages and release of software without source code. Then, in 1998, several prominent computer hackers, including Bruce Perens and Eric Raymond, launched the Open Source Initiative. The founders had some political differences with the free-software movement, but agreed in general with its licensing practices. They also advanced new ideas about how to spread the practices more broadly.

Many thousands of free and open-source software projects exist today, and the number is growing. SourceForge, a repository of open-source projects (<http://sourceforge.net>), lists more than 10,000 projects and more than 100,000 registered users. It is becoming easier to undertake new projects as effective project design becomes better understood and as prepackaged infrastructural support for projects, such as SourceForge provides, becomes available on the Web.

intellectual-property protection and set up licensing arrangements — costly endeavors with uncertain outcomes.

Second, manufacturers, through their production, distribution and field-support capability, appear to have an edge in getting widespread diffusion of an innovation. Such tasks involve large economies of scale for physical products. How could users accomplish them as cost-effectively as manufacturers? One might imagine users uniting in a fit of passion — such as the indignation many computer hackers feel toward Microsoft. But as a stable part of an ordinary economic landscape? Never!

Yet, impossible or not, user communities clearly do innovate. Moreover, when products they develop compete head-to-

head against products developed by manufacturers — Apache against Microsoft's and Netscape's server software, for example — the former seem capable of beating the latter handily in the marketplace. Not only do the communities exist, they even triumph. As Galileo is said to have murmured after officially recanting his statement that the earth moves around the sun: “And yet it moves!” What is going on?

Conditions That Favor User Innovation Communities

User communities engaged in innovation, development and consumption are most likely to flourish when three conditions are met: At least some users have sufficient incentive to innovate; at least some users have an incentive to voluntarily reveal their innovations and the means to do so; and user-led diffusion of innovations can compete with commercial production and distribution. When only the first two conditions hold, a pattern of user innovation and trial will occur, followed by commercial manufacture and distribution of any innovations proven to be of general interest.

Incentives for Users To Innovate Users have sufficient incentive to innovate when they expect the benefits of innovating to exceed their costs. Clearly, users engaged in the development of open-source software and novel sports equipment consider that condition to be met. The costs incurred by innovating users, many of whom report enjoying as well as benefiting from their efforts, can be extremely low or even negative.

Empirical research documents that user innovation exists in many fields and is concentrated in the most advanced and motivated lead-user segment of the user community. My 1988 book, “The Sources of Innovation,” showed that such was the case for industrial products and processes, and recently it has been shown to be true for consumer products as well. As Christian Luthje reported in a 2000 University of Munich working paper, 10% of German lead users of outdoor consumer-sports equipment actually improved their equipment or created entirely new equipment. With hundreds of thousands of such users in Germany, there is a tremendous amount of user innovation in that category alone.

Consider the saying “If you want something done right, do it yourself.” In the case of product and service development, the adage holds true for users in part because a manufacturer cannot know what users want as well as they themselves do — and in part because, even knowing exactly what users want, the manufacturer lacks the incentive to match their wishes in every detail.

New-product developers must have accurate information on users' needs and the context of use if they are to create prod-

ucts that will meet those needs precisely. Such information is generated at user sites, but it is typically "sticky": costly to transfer from users' sites to outside developers. (For example, the conditions that cause software — or jumping windsurfers — to crash are available for free at the site of a user with the problem but can be difficult to reproduce elsewhere.) Also, because users' needs and habits constantly change, the necessary information cannot be transferred to manufacturer-based developers all at once. Rather, it evolves at the user site as the user experiments with prototype innovations. Recall that the windsurfers discovered that they could control the direction of a board when it was in the air only *after* they began experimenting with their prototype foot straps.

that, each user would have to develop the innovation anew, which would create a huge system-level cost, or else innovators would have to protect and license their innovations and collect revenues, burdening user communities with high overhead.

Research has shown that users in a number of fields do freely reveal details of their innovations to other users and even to manufacturers. Users in open-source software communities post improvements and code on project Web sites, where anyone can view and download them for free. Free revealing is also present in the sports innovation example: Innovating users gather on the beach, inspect one another's creations, and imitate or develop additional modifications that they, in turn, freely reveal.

How are we to understand free revealing? It does not make

Consider the saying "If you want something done right, do it yourself." In the case of product and service development, the adage holds true for users in part because a manufacturer cannot know what users want as well as they themselves do.

Manufacturers are the agents of users with respect to new products and services. It is their job to develop and build what users want and need; they do not want the products for themselves. If manufacturers' incentives don't match those of users, users end up paying an "agency cost" when they delegate design to manufacturers. That agency cost often takes the form of products not being the best possible fit with users' needs. Manufacturers want to spread their development costs over as many users as possible and hence design products that are a close enough fit to induce purchase from many users. One can see that incentive at work in the users' groups manufacturers set up to get advice on desired product improvements. At group meetings, manufacturer representatives seek specifications for new products that many user-members would find acceptable and commonly urge them to make "really difficult compromises."

Manufacturers explain that they cannot afford to design and build a product unless many users will buy it: a view that is reasonable but likely to retard innovation. As research shows, innovations that only a few leaders use today may be in general demand tomorrow — particularly if lead users have a chance to innovate, to learn by doing and to develop the general utility of their innovations.

Incentives for Users To Reveal Innovations Freely Progress and success in user innovation communities is contingent on at least some users freely sharing their innovations with others. Without

conventional economic sense. In theory, innovating users should attempt to keep their innovations secret. After all, innovating users spend money and time to create their innovations, and revealing them without compensation to noninnovating users, either directly or via a manufacturer, should represent a loss.

Users will reveal innovations when their benefits outweigh their costs. In the case of user innovation communities, the costs of revealing are generally low. In a July 2000 MIT Sloan working paper, Dietmar Harhoff and colleagues identified two kinds of costs associated with revealing an innovation: the cost associated with the loss of proprietary intellectual property and the cost of diffusion. Innovating users generally will expect intellectual property losses to be low if their rivalry with potential adopters is low. (Thus town libraries share information freely because they serve different populations, do not seek to gain market share from one another and have little rivalry.) Even rivals who would prefer not to reveal an innovation will do so if they expect that others will reveal it if they do not, as K. Lakhani and I demonstrated in a May 2000 MIT Sloan working paper. That belief is held by many open-source software project participants. Also, users that cannot hide their innovations, such as high-performance windsurfers experimenting on the open beach, must reveal them.

When the costs of freely revealing an innovation are low, even a low level of benefit can be adequate reward. Various researchers, including Josh Lerner and Jean Tirole (in a March

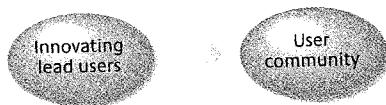
As research shows, innovations that only a few leaders use today may be in general demand tomorrow — particularly if lead users have a chance to innovate, to learn by doing and to develop the general utility of their innovations.

2000 working paper for the National Bureau of Economic Research) and Georg von Krogh (in a spring 1998 California Management Review article) have observed that adequate rewards may include improved reputations, expected reciprocity and helping to build a community.

Innovation Diffusion by Users Full-function communities in which users actually produce their innovations, bypassing manufacturers, can exist only when user manufacture and distribution can compete with commercial production and distribution. In the case of open-source software, innovations can be produced and distributed essentially for free on the Web because

Methods of Distributing User Innovations

For information products, general distribution within and beyond the user community is carried out by the community itself — no manufacturer required.



For physical products, general distribution typically requires manufacturers.



software is information rather than a physical product. In the case of the sports innovation example, however, innovations in equipment are embodied in a physical product that requires physical production and distribution and involves economies of scale. The result for physical products generally is that innovation can be carried out by users and within user innovation communities, but production and diffusion of products incorporating those innovations is usually handled by manufacturing companies. (See “Methods of Distributing User Innovations.”)

Ongoing Exploration of User Innovation Communities

The advent of the World Wide Web and the consequent proliferation of open-source software development projects has led to a growing academic focus on the phenomenon of user innovation communities in general and open-source software in particular. Thousands of open-source software projects provide natural laboratories for studying the phenomenon. Areas of interest include the conditions under which open-source software projects are likely to succeed, how they can be managed most successfully and what attracts volunteers. Our understanding of such issues is expanding daily.

However, the phenomenon is changing even as we study it. The rationale for user innovation, followed by manufacturer production in the case of physical products, is compelling, and joint user-manufacturer innovation models are evolving rapidly. An example I explore in the summer 2001 Journal of Product and Innovation Management comes from the field of custom integrated circuits. Customers are provided with tool kits that allow them to make innovations. Each year, thousands of customers use such tool kits to design billions of dollars’ worth of custom circuits that are both precisely suited to their needs and easily producible in manufacturers’ facilities.

But what is most exciting is that innovation communities composed of users and for users, communities that according to traditional economic views shouldn’t exist, work well enough to create and sustain complex innovations without *any* manufacturer involvement. This means that in at least some, and probably in many, fields users can build, consume and support innovations on their own. As we learn to understand such communities better, we will be in a position to improve them and systematically extend their reach and attendant advantages throughout the economy.

Eric von Hippel is a professor at MIT Sloan School of Management. Contact him at evhippel@mit.edu.

Reprint 4248

Copyright ©2001 by the Massachusetts Institute of Technology. All rights reserved.