Open Source Shows the Way:
Innovation by and for Users – No Manufacturer Required!

Eric von Hippel

March, 2001

Forthcoming, Sloan Management Review, Summer, 2001

Open source software projects, much in the news lately, are exciting examples of complete innovation development and consumption communities run by and for users, no manufacturer required. User innovation communities have a great advantage over the manufacturer-centered innovation development systems that have been the mainstay of commerce for hundreds of years: they enable each using entity, whether an individual or a corporation, to develop exactly what it wants rather than relying on a manufacturer to act as its (often very imperfect) agent. Moreover, individual users do not have to develop everything they need on their own: they can benefit from innovations developed by others and freely shared within the user community.

User innovation communities existed long before and extend far beyond open source software. Such communities can be found developing physical products as well. Consider and compare the following examples of early stage user innovation communities, one in software, the other in sports.

Apache Server 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 Internet-based World Wide Web infrastructure.

The server software that evolved into Apache was developed by University of Illinois undergraduate Rob McCool 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 modify and further develop it.

When McCool departed NCSA in mid-1994, a small group of web masters who had adopted his server software for their own sites decided to take on the task of continued development. A core group of eight users gathered all documentation and bug fixes and issued a consolidated patch. This *patchy* web server software evolved over time into Apache. Extensive user feedback and modification yielded Apache 1.0, released on December 1, 1995.

In the space of four years and after many modifications and improvements contributed by many users, Apache has become 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 and Netscape, it is currently in use by some 60% of the millions of web sites worldwide.

High performance windsurfing

"High-performance" windsurfing, the evolution of which was documented by MIT PhD student Sonali Shah (MIT Sloan WP # 4105, 2000), involves acrobatics such as mid-air jumps and turns. Previously, the sport tended to focus on traditional sailing techniques, windsurfing boards being used essentially as small, agile sailboats.

The fundamentals of high-performance windsurfing were developed in 1978 in Hawaii by a group of like-minded users . The development of a major innovation in technique and equipment was described to Shah by high-performance windsurfing pioneer Larry Stanley.

> "In 1978 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 mid-air because there was no way to keep the board with you.
>
> "Then I remembered the "Chip," a small experimental board we had built with footstraps. . . . That's when I first started jumping with footstraps 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 footstraps of various kinds on them and we were all going fast and jumping waves and stuff. It just kind of snowballed from there."

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 these user innovation communities have evolved and became more complex. Today, although they look different on the surface, they are in fact very similar in fundamental ways. Both evolved to include many thousands of volunteer participants.

Participants in open source software projects interact primarily via the Internet using various specialized websites volunteer users have set up for their use. Participants in innovation sports communities tend to interact by physically travelling to favorite sports sites and to types of contests that innovative users have designed for their sport. Most users of open source software simply "use the code," relying on interested volunteers to write new code, debug others' code, answer requests for help posted on Internet help sites, and help coordinate the project. Similarly, as Franke and Shah show (MIT Sloan WP #), 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 innovations developed by others, voluntarily provide coaching, and help to coordinate group activities such as "leagues," and "meets."

Often, commercial enterprises attach to or assume complementary roles to user innovation communities. Red Hat and VALinux as well-known examples of commercial involvement in the open source software context; professional sports leagues and commercial producers of sports equipment are examples in the case of user sports communities.

User innovation communities "shouldn't exist," but they do

Manufacturers, not users, have traditionally been considered the most logical developers of the innovative products they sell. There are two major reasons for this. First, financial incentives to innovate seem to be higher for manufacturers than for individual or corporate users of a product or service. After all, a manufacturer has the opportunity to sell what it develops to an entire marketplace of users. Individual user-innovators, on the other hand, can typically expect to benefit financially only from their own internal use of their innovations. Benefiting from diffusion of an innovation to the other users in a marketplace would require some form of intellectual property protection followed by licensing. Both are costly to attempt, with very uncertain outcomes.

The second reason is that for an innovation to achieve widespread diffusion invention and development must be followed by production, distribution, and field support. Because these tasks involve large economies of scale for physical products,

manufacturers have generally been assumed to have major cost advantages over individual users and communities of users. How could users possibly accomplish these tasks as cost-effectively as manufacturers? One might imagine users effectively uniting in a temporary fit of passion, such as that felt by many computer hackers today to "beat Microsoft." But as a stable part of an ordinary economic landscape? Never!

Yet, impossible or not, user innovation development and consumption communities clearly do exist. Moreover, when products they develop compete head-to-head against products developed by manufacturer-centric entities--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 these 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 here?

Conditions that favor user innovation communities

Complete user-centric innovation development and consumption communities can flourish when (1) at least some users have sufficient incentive to innovate, (2) at least some users have an incentive to voluntarily reveal their innovations and the means to do so, and (3) diffusion of innovations by users 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 innovations that prove to be of general interest.

User incentives to innovate

Users have sufficient incentive to innovate when they expect their benefits to exceed their costs of doing so. Clearly, many users engaged in the development of open source software and novel sports equipment consider this condition to be met. Indeed, 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. Proof of the pudding is in empirical research that documents the presence of user innovation in many fields,

concentrated in the most advanced and motivated "lead user" segment of the user community. This was originally shown to be the case for industrial products and processes (von Hippel, *The Sources of Innovation,* Oxford U Press 1988), but has recently been shown to be true for consumer products as well. For example, Christian Luthje (University of Munich WP, 2000) has recently shown that 10% of German users who purchase outdoor consumer sports equipment and clothing through specialty catalogs have actually developed "home made" improvements to their equipment or created entirely new equipment. Since there are hundreds of thousands of such users in Germany, this is a *lot* of user innovation in just this single category and country.

(see exhibit attached)

Figure 1: "Lead Users" are the innovators within user innovation communities

The great advantage of direct innovation by users over innovation by manufacturers, from the users' point of view, is captured by the well-known adage: "If you want something done right, do it yourself!" In the case of new product and service development this adage holds because (1) a manufacturer cannot know what a user wants to the depth and detail that the user does, and (2) even if a manufacturer *does* know exactly what a user wants, it will not have an incentive to provide *exactly* that.

New product developers clearly must have accurate information on users' needs and context of use if they are to build a product that accurately responds to user needs. This information is *generated* at user sites and is naturally accessible there but it is, as I have shown elsewhere (*Management Science* 40, no.4, 1994), typically very "sticky," - costly to move 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 very difficult to reproduce elsewhere.) Also, this information is not a static matter that can be transferred to manufacturer-based developers all at once. Rather, it evolves at the user site through "learning by doing" as the user

experiments with prototype innovations. (Recall from the windsurfing example that users only *discovered* that they could and wanted to control the direction of a board when it was in the air *after* they began experimenting with the prototype footstraps they had developed.)

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. The trouble is that because manufacturers' incentives don't match those of users, users end up paying an "agency cost" when they delegate design to manufacturers.

A major part of this agency cost takes the form of being offered products that are not be the best possible fit with users' needs, even assuming that manufacturers know precisely what those needs are. Manufacturers want to spread their development costs over as many users as possible. This leads them to want to design products that are a close-enough fit to induce purchase from many users rather than to design precisely what any particular user really wants. One can see this incentive at work in the operation of product "users' groups" set up by manufacturers to give advice on desired product improvements. Commonly, manufacturer representatives in these groups urge user-members to make "really difficult compromises" regarding what they really want and to create a common specification for a desired new product. After all, as they point out, a manufacturer cannot afford to design and build a product unless many users will want to buy.

This view is reasonable from the manufacturers' perspective, but can retard the innovation process in the absence of innovation by users. As was mentioned earlier, research shows that innovations wanted by only a few "lead users" today will often turn out to represent general demand tomorrow, *if* lead users have a chance to innovate, to learn by doing and to sometimes reveal the general utility of their innovations.

User incentives to freely reveal innovations

Progress and success in user innovation communities is contingent on at least some users freely sharing their innovations with others. Absent free revealing, each user

would have to redevelop the same innovation in order to use it, resulting in a huge system-level cost, or resort to protecting and licensing their innovations and collecting revenues from other users, which would burden the communities with tremendous 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, for example, post improvements and code on project websites where anyone, from rival users to manufacturers, can view and download them for free. Free revealing is also clearly 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 such behavior? Free revealing does not make sense from the point of view of conventional economic wisdom, which reasons that innovating users should attempt to keep their innovation-related information secret. After all, as this reasoning goes, innovating users spend money and time to create their innovations, and revealing their developments without compensation to non-innovating users, either directly or via a manufacturer, should represent a loss that users should strive to avoid .

Users will reveal innovations when the costs of revealing are outweighed by the benefits. In the case of user innovation communities, the costs of revealing are generally low. Harhoff et al. (MIT WP # 4125, 2000) have identified two kinds of costs associated with revealing an innovation: those associated with the loss of proprietary intellectual property; and the cost of diffusion. With respect to intellectual property losses, users who have innovated will expect low losses if they have low rivalry with potential adopters. (For example, town libraries have low rivalry; they serve different populations and do not seek to gain market share from each other.) Even those who would prefer not to reveal due to rivalry considerations will do so if they expect that others will reveal if they do not. Lakhani and von Hippel (MIT WP#4117-2000) report that this belief appears to be held by many open source software project participants . Also, of course, users who cannot hide their innovations will freely reveal. High-performance windsurfers

experimenting on the open beach, for example, have no plausible way to hide their technique and hardware innovations from fellow users.

When the costs of freely revealing an innovation are low, even a low level of benefit can be adequate reward. As Lerner and Tirole (NBER WP#7600, 2000, and von Krogh, California Management Review (40)3, 1998) observe, adequate rewards can be provided to participants in user innovation communities in a variety of forms, including elevated reputations, expected reciprocity, and incentives to help build a community .

Innovation diffusion by users

"Full-function" user innovation and *production* communities – no manufacturer required – are possible only when self-manufacture and/or distribution of innovative products directly by users can compete with commercial production and distribution. In the case of open source software this is possible because innovations can be "produced" and distributed essentially for free on the web, software being an information rather than a physical product. In the case of the sports innovation example, however, equipment (but not technique) innovations are embodied in physical product that, to achieve general diffusion, must be produced and physically distributed. These activities do, as was mentioned earlier, involve significant economies of scale. The result, in the case of the windsurfing example and for physical products generally, is that, while innovation can be be carried out by users and within user innovation communities, production and diffusion of products incorporating those innovations is usually carried out by manufacturing firms.

For information products - no manufacturer required

Lead users → user community

For physical products – general distribution by manufacturers

Lead users → manufacturer → routine users

Figure 2: How lead user innovations are distributed

Ongoing exploration of user innovation communities

The advent of the web and consequent public proliferation of open source software development projects has focused intense academic attention on the phenomenon of user innovation communities in general and open source software, in particular. The thousands of extant open source software projects represent natural experiments that academics and others can study to better understand this phenomenon. Among the issues being explored now are conditions under which open source software projects can be expected to succeed, how they can be most successfully managed, and what attracts the interest of volunteers. We can expect rapid progress on these fronts.

Of course, 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 rapidly evolving in advanced fields. I explain elsewhere (*Journal of Product and Innovation Management*, summer 2001) how the field of custom integrated circuits has pioneered an especially exciting user-manufacturer innovation model that involves providing customers with "user toolkits for innovation." Each year, thousands of customers use these toolkits to innovate by and for themselves, successfully designing billions of dollars worth of custom circuits that are both precisely suited to their needs and also easily producible in manufacturers' production facilities.

But what is most exciting is that innovation communities exclusively by and for users, communities that by any yardstick of traditional economics shouldn't exist, work well enough to create and sustain complex innovation products without *any* manufacturer involvement. This means that in at least some, and probably in many, important fields users can build, consume, and support innovations on their own independent of manufacturer incentives to participate. As we learn to understand such communities better, we will be in a position to improve them where they now exist and systematically extend their reach and attendant advantages throughout the economy.

NOTE: Recent readings on open source software and user innovation communities, including those mentioned in this article, can be downloaded from the website opensource.mit.edu. This resource is intended for those interested in keeping updated on, and perhaps contributing to, our understanding of these phenomena.

Sidebar: What is Open Source Software?

Open source designates software that is universally accessible and can be downloaded, used, and modified by anyone "for free." The legal mechanism that makes this possible is Copyleft and similar legal agreements; the technical mechanism is free access to the "source code" used to create the software. Well-known examples of open source software are the GNU/Linux computer operating system, Perl programming language, and Internet e-mail engine SendMail.

Open source software has its roots in the "free software" movement started by Richard Stallman in the early 1980s. Stallman founded the Free Software Foundation (FSF) to counter the trend towards proprietary development of software packages and release of software without accompanying source code. A key feature of FSF- based development is a licensing scheme called General Public License (GPL), commonly referred to as Copyleft. GPL accords program authors the traditional and legal entitlements of copyright protection together with unique distribution terms that grant all users the right to use, modify, and redistribute a program's code or any program derived from it provided that the distribution terms are preserved. Code and freedoms are thus rendered legally inseparable. The Copyleft concept prevents private hoarding that would be possible if free software were simply released into the public domain.

The open and free sharing of information and innovations in an open source project supports the emergence of a community of users with a range of interests related to extending and supporting the initial innovation. For example, some members might discover errors that are unique to their situation, others write code that enables them to

use the product in ways and environments unanticipated by the initial innovator. Still other users might be interested in providing support in the areas of trouble shooting and novice user help that might not be of interest to the original innovator.

Many thousands of open source projects exist today and the number is growing rapidly. A repository of open source projects, Sourceforge.net, lists in excess of 10,000 projects and more than 100,00 registered users. Implementing new open source projects is becoming progressively easier as effective project design becomes better understood and prepackaged infrastructural support for such projects, such as is provided by SourceForge, becomes available on the Web.