9 Democratizing Innovation

We have learned that lead users sometimes develop and modify products for themselves and often freely reveal what they have done. We have also seen that many users can be interested in adopting the solutions that lead users have developed. Taken together, these findings offer the basis for user-centered innovation systems that can entirely supplant manufacturer-based innovation systems under some conditions and complement them under most. User-centered innovation is steadily increasing in importance as computing and communication technologies improve.

I begin this chapter with a discussion of the ongoing democratization of innovation. I then describe some of the patterns in user-centered innovation that are emerging. Finally, I discuss how manufacturers can find ways to profitably participate in emerging, user-centered innovation processes.

The Trend toward Democratization

Users' abilities to develop high-quality new products and services for themselves are improving radically and rapidly. Steady improvements in computer software and hardware are making it possible to develop increasingly capable and steadily cheaper tools for innovation that require less and less skill and training to use. In addition, improving tools for communication are making it easier for user innovators to gain access to the rich libraries of modifiable innovations and innovation components that have been placed into the public domain. The net result is that rates of user innovation will increase even if users' heterogeneity of need and willingness to pay for "exactly right" products remain constant.

The radical nature of the change that is occurring in design capabilities available to even individual users is perhaps difficult for those without personal innovation experience to appreciate. An anecdote from my own experience may help as illustration. When I was a child and designed new products that I wanted to build and use, the ratio of not-too-pleasurable (for me) effort required to actually build a prototype relative to the very pleasurable effort of inventing it and use-testing it was huge. (That is, in terms of the design, build, test, evaluate cycle illustrated in figure 5.1, the effort devoted to the "build" element of the cycle was very large and the rate of iteration and learning via trial and error was very low.)

In my case it was especially frustrating to try to build anything sophisticated from mechanical parts. I did not have a machine shop in which I could make good parts from scratch, and it often was difficult to find or buy the components I needed. As a consequence, I had to try to assemble an approximation of my ideas out of vacuum cleaner parts and other bits of metal and plastic and rubber that I could buy or that were lying around. Sometimes I failed at this and had to drop an exciting project. For example, I found no way to make the combustion chamber I needed to build a large pulse-jet engine for my bicycle (in retrospect, perhaps a lucky thing!). Even when I succeeded, the result was typically "unaesthetic": the gap between the elegant design in my mind and the crude prototype that I could realize was discouragingly large.

Today, in sharp contrast, user firms and increasingly even individual hobbyists have access to sophisticated design tools for fields ranging from software to electronics to musical composition. All these information-based tools can be run on a personal computer and are rapidly coming down in price. With relatively little training and practice, they enable users to design new products and services—and music and art—at a satisfyingly sophisticated level. Then, if what has been created is an information product, such as software or music, the design is the actual product—software you can use or music you can play.

If one is designing a physical product, it is possible to create a design and even conduct some performance testing by computer simulation. After that, constructing a real physical prototype is still not easy. However, today users do have ready access to kits that offer basic electronic and mechanical building blocks at an affordable price, and physical product prototyping is becoming steadily easier as computer-driven 3-D parts printers continue to go up in sophistication while dropping in price. Very excitingly, even today home-built prototypes need not be poorly fashioned items that will fall apart with a touch in the wrong place—the solution components now available to users are often as good as those available to professional designers.

Functional equivalents of the resources for innovation just described have long been available within corporations to a lucky few. Senior designers at firms have long been supported by engineers and designers under their direct control, and also with other resources needed to quickly construct and test prototype designs. When I took a job as R&D manager at a start-up firm after college, I was astounded at the difference professional-quality resources made to both the speed and the joy of innovation. Product development under these conditions meant that the proportion of one's effort that could be focused on the design and test portions of the innovation cycle rather than on prototype building was much higher, and the rate of progress was much faster.

The same story can be told in fields from machine design to clothing design: just think of the staffs of seamstresses and models supplied by clothing manufacturers to their "top designers" so that these few can quickly realize and test many variations on their designs. In contrast, think of the time and effort that equally talented designers without such staff assistance must engage in to stitch together even a single high-quality garment prototype on their own.

But, as we learned in chapter 7, the capability and the information needed to innovate in important ways are in fact widely distributed. Given this finding, we can see that the traditional pattern of concentrating innovation-support resources on just a few pre-selected potential innovators is hugely inefficient. High-cost resources for innovation support cannot be allocated to "the right people," because one does not know who they are until they develop an important innovation. When the cost of high-quality resources for design and prototyping becomes very low—which is the trend we have described—these resources can be diffused widely, and the allocation problem then diminishes in significance. The net result is and will be to democratize the opportunity to create.

Democratization of the opportunity to create is important beyond giving more users the ability to make exactly right products for themselves. As we saw in a previous chapter, the joy and the learning associated with creativity and membership in creative communities are also important, and these experiences too are made more widely available as innovation is democratized. The aforementioned Chris Hanson, a Principal Research Scientist at MIT and a maintainer in the Debian Linux community, speaks eloquently of this in his description of the joy and value he finds from his participation in an open source software community:

> Creation is unbelievably addictive. And programming, at least for skilled programmers, is highly creative. So good programmers are compelled to program to feed the addiction. (Just ask my wife!) Creative programming takes time, and careful attention to the details. Programming is all about expressing intent, and in any large program there are many areas in which the programmer's intent is unclear. Clarification requires insight, and acquiring insight is the primary creative act in programming. But insight takes time and often requires extensive conversation with one's peers.
>
> Free-software programmers are relatively unconstrained by time. Community standards encourage deep understanding, because programmers know that understanding is essential to proper function. They are also programming for themselves, and naturally they want the resulting programs to be as good as they can be. For many, a free software project is the only context in which they can write a program that expresses their own vision, rather than implementing someone else's design, or hacking together something that the marketing department insists on. No wonder programmers are willing to do this in their spare time. This is a place where creativity thrives.
>
> Creativity also plays a role in the programming community: programming, like architecture, has both an expressive and a functional component. Unlike architecture, though, the expressive component of a program is inaccessible to non-programmers. A close analogy is to appreciate the artistic expression of a novel when you don't know the language in which it is written, or even if you know the language but are not fluent. This means that creative programmers want to associate with one another: only their peers are able to truly appreciate their art. Part of this is that programmers want to earn respect by showing others their talents. But it's also important that people want to share the beauty of what they have found. This sharing is another act that helps build community and friendship.

Adapting to User-Centered Innovation—Like It or Not

User-centered innovation systems involving free revealing can sometimes supplant product development carried out by manufacturers. This outcome seems reasonable when manufacturers can obtain field-tested user designs at no cost. As an illustration, consider kitesurfing (previously discussed in chapter 7). The recent evolution of this field nicely shows how manufacturer-based product design may not be able to survive when challenged by a user innovation community that freely reveals leading-edge designs developed by users. In such a case, manufacturers may be obliged to retreat to manufacturing only, specializing in modifying user-developed designs for producibility and manufacturing these in volume.

Recall that equipment for kitesurfing was initially developed and built by user-enthusiasts who were inventing both kitesurfing techniques and kitesurfing equipment interdependently. Around 1999, the first of several small manufacturers began to design and sell kitesurfing equipment commercially. The market for kitesurfing equipment then began to grow very rapidly. In 2001 about 5,000 kite-and-board sets were sold worldwide. In 2002 the number was about 30,000, and in 2003 it was about 70,000. With a basic kite-and-board set selling for about $1,500, total sales in 2003 exceeded $100 million. (Many additional kites, home-made by users, are not included in this calculation.) As of 2003, about 40 percent of the commercial market was held by a US firm called Robbie Naish (Naishkites.com).

Recall also that in 2001 Saul Griffith, an MIT graduate student, established an Internet site called Zeroprestige.com as a home for a community of kitesurfing users and user-innovators. In 2003, the general consensus of both site participants and manufacturers was that the kite designs developed by users and freely revealed on Zeroprestige.com were at least as advanced as those developed by the leading manufacturers. There was also a consensus that the level of engineering design tools and aggregate rate of experimentation by kite users participating on the Zeroprestige.com site was superior to that within any kite manufacturer. Indeed, this collective user effort was probably superior in quality and quantity to the product-development work carried out by all manufacturers in the industry taken together.

In late 2003, a perhaps predictable event occurred: a kite manufacturer began downloading users' designs from Zeroprestige.com and producing them for commercial sale. This firm had no internal kitesurfing product-development effort and offered no royalties to user-innovators—who sought none. It also sold its products at prices much lower than those charged by companies that both developed and manufactured kites.

It is not clear that manufacturers of kitesurfing equipment adhering to the traditional developer-manufacturer model can—or should—survive this new and powerful combination of freely revealed collaborative design and prototyping effort by a user innovation community combined with volume production by a specialist manufacturer. In effect, free revealing of product designs by users offsets manufacturers' economies of scale in design with user communities' economies of scope. These economies arise from the heterogeneity in information and resources found in a user community.

Manufacturers' Roles in User-Centered Innovation

Users are not required to incorporate manufacturers in their product-development and product-diffusion activities. Indeed, as open source software projects clearly show, horizontal innovation communities consisting entirely of users can develop, diffuse, maintain, and consume software and other *information* products by and for themselves—no manufacturer is required. Freedom from manufacturer involvement is possible because information products can be "produced" and distributed by users essentially for free on the web (Kollock 1999). In contrast, production and diffusion of physical products involves activities with significant economies of scale. For this reason, while product development and early diffusion of copies of physical products developed by users can be carried out by users themselves and within user innovation communities, mass production and general diffusion of physical products incorporating user innovations are usually carried out by manufacturing firms.

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

Innovating lead users ➔ All users.

For physical products, general distribution typically requires manufacturers:

Innovating lead users ➔ Manufacturer ➔ All users.

In light of this situation, how can, should, or will manufacturers of products, services, and processes play profitable roles in user-centered innovation systems? Behlendorf (1999), Hecker (1999) and Raymond (1999) explore what might be possible in the specific context of open source software. More generally, many are experimenting with three possibilities: (1) Manufacturers may produce user-developed innovations for general commercial sale and/or offer a custom manufacturing service to specific users. (2) Manufacturers may sell kits of product-design tools and/or "product platforms" to ease users' innovation-related tasks. (3) Manufacturers may sell products or services that are complementary to user-developed innovations.

Producing User-Developed Products

Firms can make a profitable business from identifying and mass producing user-developed innovations or developing and building new products based on ideas drawn from such innovations. They can gain advantages over competitors by learning to do this better than other manufacturers. They may, for example, learn to identify commercially promising user innovations more effectively that other firms. Firms using lead user search techniques such as those we will describe in chapter 10 are beginning to do this systematically rather than accidentally—surely an improvement. Effectively transferring user-developed innovations to mass manufacture is seldom as simple as producing a product based on a design by a single lead user. Often, a manufacturer combines features developed by several independent lead users to create an attractive commercial offering. This is a skill that a company can learn better than others in order to gain a competitive advantage.

The decision as to whether or when to take the plunge and commercialize a lead user innovation(s) is also not typically straightforward, and companies can improve their skills at inviting in the relevant information and making such assessments. As was discussed previously, manufacturers often do not understand emerging user needs and markets nearly as well as lead users do. Lead users therefore may engage in entrepreneurial activities, such as "selling" the potential of an idea to potential manufacturers and even lining up financing for a manufacturer when they think it very important to rapidly get widespread diffusion of a user-developed product. Lettl, Herstatt, and Gemünden (2004), who studied the commercialization of major advances in surgical equipment, found innovating users commonly engaging in these activities. It is also possible, of course, for innovating lead users to become manufacturers and produce the products they developed for general commercial sale. This has been shown to occur fairly frequently in the field of sporting goods (Shah 2000; Shah and Tripsas 2004; Hienerth 2004).

Manufacturers can also elect to provide custom production or "foundry" services to users, differentiating themselves by producing users' designs faster, better, and/or cheaper than competitors. This type of business model is already advanced in many fields. Custom machine shops specialize in manufacturing mechanical parts to order; electronic assembly shops produce custom electronic products, chemical manufacturers offer "toll" manufacturing of custom products designed by others, and so on. Suppliers of custom integrated circuits offer an especially good example of custom manufacture of products designed by users. More than $15 billion worth of custom integrated circuits were produced in 2002, and the cumulative average growth rate of that market segment was 29 percent. Users benefit from designing their own circuits by getting exactly what they want more quickly than manufacturer-based engineers could supply what they need, and manufacturers benefit from producing the custom designs for users (Thomke and von Hippel 2002).

Supplying Toolkits and/or Platform Products to Users

Users interested in designing their own products want to do it efficiently. Manufacturers can therefore attract them to kits of design tools that ease their product-development tasks and to products that can serve as "platforms" upon which to develop and operate user-developed modifications. Some are supplying users with proprietary sets of design tools only. Cadence, a supplier of design tools for corporate and even individual users interested in designing their own custom semiconductor chips, is an example of this. Other manufacturers, including Harley-Davidson in the case of motorcycles and Microsoft in the case of its Excel spreadsheet software, sell platform products intentionally designed for post-sale modification by users.

Some firms that sell platform products or design tools to users have learned to systematically incorporate valuable innovations that users may develop back into their commercial products. In effect, this second strategy can often be pursued jointly with the manufacturing strategy described above. Consider, for example, StataCorp of College Station, Texas. StataCorp produces and sells Stata, a proprietary software program designed for statistics. It sells the basic system bundled with a number of families of statistical tests and with design tools that enable users to develop new tests for operation on the Stata platform. Advanced customers, many of them statisticians and social science researchers, find this capability very important to their work and do develop their own tests. Many then freely reveal tests they have developed on Internet websites set up by the users themselves. Other users then visit these sites to download and use, and perhaps to test, comment on, and improve these tests, much as users do in open source software communities.

StataCorp personnel monitor the activity at user sites, and note the new tests that are of interest to many users. They then bring the most popular tests into their product portfolio as Stata modules. To do this, they rewrite the user's software code while adhering to the principles pioneered by the user-innovator. They then subject the module to extensive validation testing—a very important matter for statisticians. The net result is a symbiotic relationship. User-innovators are publicly credited by Stata for their ideas, and benefit by having their modules professionally tested. StataCorp gains a new commercial test module, rewritten and sold under its own copyright. Add-ons developed by users that are freely revealed will increase StataCorp's profits more than will equivalent add-ons developed and sold by manufacturers (Jokisch 2001). Similar strategies are pursued by manufacturers of simulator software (Henkel and Thies 2003).

Note, however, that StataCorp, in order to protect its proprietary position, does not reveal the core of its software program to users, and does not allow any user to modify it. This creates problems for those users who need to make modifications to the core in order to solve particular problems they encounter. Users with problems of this nature and users especially concerned about price have the option of turning to non-proprietary free statistical software packages available on the web, such as the "R" project (www.r-project.org). These alternatives are developed and supported by user communities and are available as open source software. The eventual effect of open source software alternatives on the viability of the business models of commercial vendors such as StataCorp and its competitors remains to be seen.

A very similar pattern exists in the online gaming industry. Vendors of early online computer games were surprised to discover that sophisticated users were deciphering their closed source code in order to modify the games to be more to their liking. Some of these "mods" attracted large followings, and some game vendors were both impressed and supportive. Manufacturers also discovered that the net effect of user-developed mods was positive for them: mods actually increased the sales of their basic software, because users had to buy the vendors' proprietary software engine code in order to play the mods. Accordingly, a number of vendors began to actively support user-developers by supplying them with design tools to make it easier for them to build mods on their proprietary engine platforms (Jeppesen and Molin 2003).

Both manufacturers and users involved with online gaming are experimenting with the possibilities of user-manufacturer symbiosis in a number of additional ways. For example, some vendors are experimenting with creating company-supported distribution channels through which users—who then become vendors—can sell their mods rather than simply offering them as free downloads (Jeppesen 2004). At the same time, some user communities are working in the opposite direction by joining together to develop open source software engines for video games. If the latter effort is successful, it will offer mod developers a platform and design tools that are entirely non-proprietary for the first time. As in the case of statistical software, the eventual outcomes of all these experiments are not yet clear.

As a final example of a strategy in which manufacturers offer a platform to support user innovation of value to them, consider General Electric's innovation pattern with respect to the magnetic-resonance imaging machines it sells for medical use. Michael Harsh (GE's Director of R&D in the division that produces MRI machines) and his colleagues realized that nearly all the major, commercially important improvements to these machines are developed by leading-edge users rather than by GE or by competing machine producers. They also knew that commercialization of user-developed improvements would be easier and faster for GE if the users had developed their innovations using a GE MRI machine as a platform rather than a competitor's machine. Since MRI machines are expensive, GE developed a policy of selectively supplying machines at a very low price to scientists GE managers judged most likely to develop important improvements. These machines are supplied with restrictive interlocks removed so that the users can easily modify them. In exchange for this research support, the medical researchers give GE preferred access to innovations they develop. Over the years, supported researchers have provided a steady flow of significant improvements that have been first commercialized by GE. Managers consider the policy a major source of GE's commercial success in the MRI field.

Providing Complementary Products or Services

Many user innovations require or benefit from complementary products or services, and manufacturers can often supply these at a profit. For example, IBM profits from user innovation in open source software by selling the complement of computer hardware. Specifically, it sells computer servers with open source software pre-installed, and as the popularity of that software goes up, so do server sales and profits. A firm named Red Hat distributes a version of the open source software computer operating system Linux, and also sells the complementary service of Linux technical support to users. Opportunities to provide profitable complements are not necessarily obvious at first glance, and providers often reap benefits without being aware of the user innovation for which they are providing a complement. Hospital emergency rooms, for example, certainly gain considerable business from providing medical care to the users and user-developers of physically demanding sports, but may not be aware of this.

Discussion

All the examples above explore how manufacturers can integrate themselves into a user-centered innovation system. However, manufacturers will not always find user innovations based on or related to their products to be in their interest. For example, manufacturers may be concerned about legal liabilities and costs sometimes associated with "unauthorized user tinkering." For example, an automaker might legitimately worry about the user-programmed engine controller chips that racing aficionados and others often install to change their cars' performance. The result can be findings of eventual commercial value as users explore new performance regimes that manufacturers' engineers might not have considered. However, if users choose to override manufacturers' programming to increase engine performance, there is also a clear risk of increased warrantee costs for manufacturers if engines fail as a consequence (Mollick 2004).

We have seen that manufacturers can often find ways to profit from user innovation. It is also the case, however, that user innovators and user innovation communities can provide many of these same functions for themselves. For example, StataCorp is successfully selling a proprietary statistical software package. User-developed alternatives exist on the web that are developed and maintained by user-innovators and can be downloaded at no charge. Which ownership model will prove more robust under what circumstances remains to be seen. Ultimately, since users are the customers, they get to choose.