

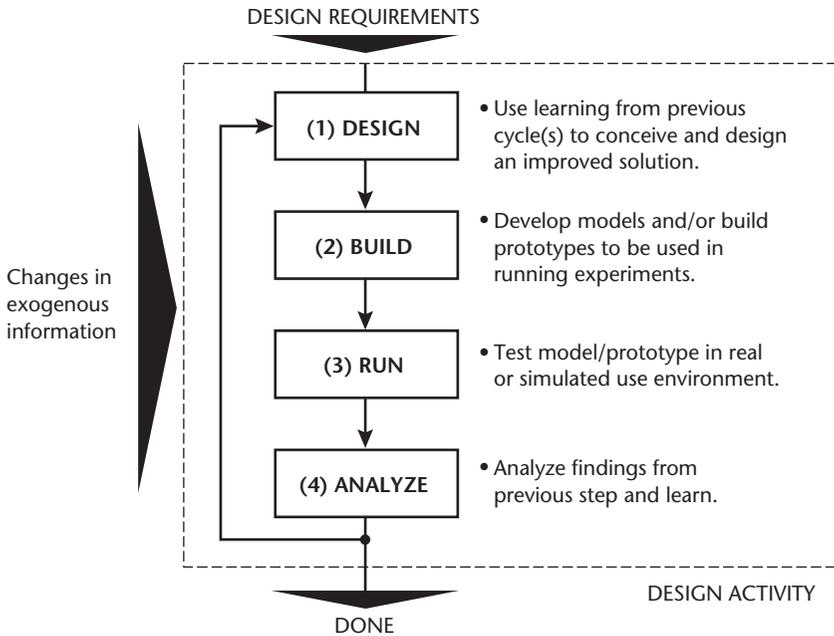
## 5 Users' Low-Cost Innovation Niches

### The Problem-Solving Process

Product and service development is at its core a problem-solving process. Research into the nature of problem solving shows it to consist of trial and error, directed by some amount of insight as to the direction in which a solution might lie (Baron 1988). Trial and error has also been found to be prominent in the problem-solving work of product and process development (Marples 1961; Allen 1966; von Hippel and Tyre 1995; Thomke 1998, 2003).

Trial-and-error problem solving can be envisioned as a four-phase cycle that is typically repeated many times during the development of a new product or service. Problem solvers first conceive of a problem and a related solution based on their best knowledge and insight. Next, they build a physical or virtual prototype of both the possible solution they have envisioned and the intended use environment. Third, they run the experiment—that is, they operate their prototyped solution and see what happens. Fourth and finally, they analyze the result to understand what happened in the trial and to assess the “error information” that they gained. (In the trial-and-error formulation of the learning process, error is the new information or learning derived from an experiment by an experimenter: it is the aspect(s) of the outcome that the experimenter did not predict.) Developers then use the new learning to modify and improve the solution under development before building and running a new trial (figure 5.1).

Trial-and-error experimentation can be informal or formal; the underlying principles are the same. As an example on the informal side, consider a user experiencing a need and then developing what eventually turns out to be a new product: the skateboard. In phase 1 of the cycle, the user combines



**Figure 5.1**

The trial-and-error cycle of product development.

need and solution information into a product idea: “I am bored with roller skating. How can I get down this hill in a more exciting way? Maybe it would be fun to put my skates’ wheels under a board and ride down on that.” In phase 2, the user builds a prototype by taking his skates apart and hammering the wheels onto the underside of a board. In phase 3, he runs the experiment by climbing onto the board and heading down the hill. In phase 4, he picks himself up from an inaugural crash and thinks about the error information he has gained: “It is harder to stay on this thing than I thought. What went wrong, and how can I improve things before my next run down the hill?”

As an example of more formal experimentation, consider a product-development engineer working in a laboratory to improve the performance of an automobile engine. In phase 1, need and solution information are again combined into a design idea: “I need to improve engine fuel efficiency. I think that a more even expansion of the flame in the cylinders is a possible solution direction, and I think that changing the shape of the

spark plug electrodes will improve this.” In phase 2, the engineer builds a spark plug incorporating her new idea. In phase 3, she inserts the new spark plug into a lab test engine equipped with the elaborate instrumentation needed to measure the very rapid propagation of a flame in the cylinders of an auto engine and runs the test. In phase 4, she feeds the data into a computer and analyzes the results. She asks: “Did the change in spark plug design change the flame front as expected? Did it change fuel efficiency? How can I use what I have learned from this trial to improve things for the next one?”

In addition to the difference in formality, there is another important difference between these two examples. In the first example, the skateboard user was conducting trial and error with a full prototype of the intended product in a real use environment—his own. In the second example, the experimental spark plug might have been a full prototype of a real product, but it probably consisted only of that portion of a real spark plug that actually extends into a combustion chamber. Also, only *aspects* of the use environment were involved in the lab experiment. That is, the test engine was not a real auto engine, and it was not being operated in a real car traveling over real roads.

Experimentation is often carried out using simplified versions—models—of the product being designed and its intended use environment. These models can be physical (as in the example just given), or they can be virtual (as in the case of thought experiments or computer simulations). In a computer simulation, both the product and the environment are represented in digital form, and their interaction is tested entirely within a computer. For example, one might make a digital model of an automobile and a crash barrier. One could then use a computer to simulate the crash of the model car into the model barrier. One would analyze the results by calculating the effects of that crash on the structure of the car.

The value of using models rather than the real thing in experimentation is twofold. First, it can reduce the cost of an experiment—it can be much cheaper to crash a simulated BMW than a real one. Second, it can make experimental results clearer by making them simpler or otherwise different than real life. If one is trying to test the effect of a small change on car safety, for example, it can be helpful to remove everything not related to that change from the experiment. For example, if one is testing the way a particular wheel suspension structure deforms in a crash, one does not have

to know (or spend time computing) how a taillight lens will react in the crash. Also, in a real crash things happen only once and happen very fast. In a virtual crash executed by computer, on the other hand, one can repeat the crash sequence over and over, and can stretch time out or compress it exactly as one likes to better understand what is happening (Thomke 2003).

Users and others experimenting with real prototypes in real use environments can also modify things to make tests simpler and clearer. A restaurant chef, for example, can make slight variations in just a small part of a recipe each time a customer calls for it, in order to better understand what is happening and make improvements. Similarly, a process machine user can experiment with only a small portion of machine functioning over and over to test changes and detect errors.

Sometimes designers will test a real experimental object in a real experimental context only after experimenting with several generations of models that isolate different aspects of the real and/or encompass increasing amounts of the complexity of the real. Developers of pharmaceuticals, for example, might begin by testing a candidate drug molecule against just the purified enzyme or receptor it is intended to affect, then test it again and again against successively more complex models of the human organism (tissue cultures, animal models, etc.) before finally seeking to test its effect on real human patients during clinical trials (Thomke, von Hippel, and Franke 1998).

### **Sticky Information**

Any experiment is only as accurate as the information that is used as inputs. If inputs are not accurate, outcomes will not be accurate: “garbage in, garbage out.”

The goal of product development and service development is to create a solution that will satisfy needs of real users within real contexts of use. The more complete and accurate the information on these factors, the higher the fidelity of the models being tested. If information could be transferred costlessly from place to place, the quality of the information available to problem solvers would or could be independent of location. But if information is costly to transfer, things are different. User-innovators, for example, will then have better information about their needs and their use context than will manufacturers. After all, they create and live in that type

of information in full fidelity! Manufacturer-innovators, on the other hand, must transfer that information to themselves at some cost, and are unlikely to be able to obtain it in full fidelity at any cost. However, manufacturers might well have a higher-fidelity model of the solution types in which they specialize than users have.

It turns out that much information needed by product and service designers is “sticky.” In any particular instance, the stickiness of a unit of information is defined as the incremental expenditure required to transfer that unit of information to a specified location in a form usable by a specified information seeker. When this expenditure is low, information stickiness is low; when it is high, stickiness is high (von Hippel 1994). That information is often sticky has been shown by studying the costs of transferring information regarding fully developed process technology from one location to another with full cooperation on both sides. Even under these favorable conditions, costs have been found to be high—leading one to conclude that the costs of transferring information during product and service development are likely to be at least as high. Teece (1977), for example, studied 26 international technology-transfer projects and found that the costs of information transfer ranged from 2 percent to 59 percent of total project costs and averaged 19 percent—a considerable fraction. Mansfield et al. (1982) also studied a number of projects involving technology transfer to overseas plants, and also found technology-transfer costs averaging about 20 percent of total project costs. Winter and Suzlanski (2001) explored replication of well-known organizational routines at new sites and found the process difficult and costly.

Why is information transfer so costly? The term “stickiness” refers only to a consequence, not to a cause. Information stickiness can result from causes ranging from attributes of the information itself to access fees charged by an information owner. Consider tacitness—a lack of explicit encoding. Polanyi (1958, pp. 49–53) noted that many human skills are tacit because “the aim of a skillful performance is achieved by the observance of a set of rules which are not known as such to the person following them.” For example, swimmers are probably not aware of the rules they employ to keep afloat (e.g., in exhaling, they do not completely empty their lungs), nor are medical experts generally aware of the rules they follow in order to reach a diagnosis of a disease. “Indeed,” Polanyi says, “even in modern industries the indefinable knowledge is still an essential part of technology.” Information that

is tacit is also sticky because it cannot be transferred at low cost. As Polanyi points out, “an art which cannot be specified in detail cannot be transmitted by prescription, since no prescription for it exists. It can be passed on only by example from master to apprentice. . . .” Apprenticeship is a relatively costly mode of transfer.

Another cause of information stickiness is related to absorptive capacity. A firm’s or an individual’s capacity to absorb new, outside technical information is largely a function of prior related knowledge (Cohen and Levinthal 1990). Thus, a firm knowing nothing about circuit design but seeking to apply an advanced technique for circuit engineering may be unable to apply it without first learning more basic information. The stickiness of the information about the advanced technique for the firm in question is therefore higher than it would be for a firm that already knows that basic information. (Recall that the stickiness of a unit of information is defined as the incremental expenditure required to transfer a unit of information to a specified site in a form usable by a *specific* information seeker.)

Total information stickiness associated with solving a specific problem is also determined by the amount of information required by a problem solver. Sometimes a great deal is required, for two reasons. First, as Rosenberg (1976, 1982) and Nelson (1982, 1990) point out, much technological knowledge deals with the specific and the particular. Second, one does not know in advance of problem solving which particular items will be important.

An example from a study by von Hippel and Tyre (1995) illustrates both points nicely. Tyre and I studied how and why novel production machines failed when they were first introduced into factory use. One of the machines studied was an automated machine used by a computer manufacturing firm to place large integrated circuits onto computer circuit boards. The user firm had asked an outside group to develop what was needed, and that group had developed and delivered a robot arm coupled to a machine-vision system. The arm, guided by the vision system, was designed to pick up integrated circuits and place them on a circuit board at precise locations.

Upon being installed in the factory, the new component-placing machine failed many times as a result of its developers’ lack of some bit of information about the need or use environment. For example, one day machine operators reported that the machine was malfunctioning—again—and they

did not know why. Investigation traced the problem to the machine-vision system. This system used a small TV camera to locate specific metalized patterns on the surface of each circuit board being processed. To function, the system needed to “see” these metalized patterns clearly against the background color of the board’s surface. The vision system developed by the machine-development group had functioned properly in their lab when tested with sample boards from the user factory. However, the field investigation showed that in the factory it failed when boards that were light yellow in color were being processed.

The fact that some of the boards being processed *were* sometimes light yellow was a surprise to the machine developers. The factory personnel who had set the specifications for the machine knew that the boards they processed varied in color; however, they had not volunteered the information, because they did not know that the developers would be interested. Early in the machine-development process, they had simply provided samples of boards used in the factory to the machine-development group. And, as it happened, these samples were green. On the basis of the samples, developers had then (implicitly) assumed that all boards processed in the field were green. It had not occurred to them to ask users “How much variation in board color do you generally experience?” Thus, they had designed the vision system to work successfully with boards that were green.

In the case of this field failure, the item of information needed to understand or predict this problem was known to the users and could easily have been provided to the machine developers—had the developers thought to ask and/or had users thought to volunteer it. But in the actual evolution of events this was not done. The important point is that this omission was not due to poor practice; it was due to the huge amount of information about the need and the use environment that was *potentially* relevant to problem solvers. Note that the use environment and the novel machine contain many highly specific attributes that could potentially interact to cause field problems. Note also that the property of the board causing this particular type of failure was very narrow and specific. That is, the problem was not that the board had physical properties, nor that it had a color. The problem was precisely that some boards were yellow, and a particular shade of yellow at that. Since a circuit board, like most other components, has many attributes in addition to color (shape, size, weight, chemical composition, resonant frequency, dielectric constant, flexibility, and so on), it is likely that

problem solvers seeking to learn everything they might need to know about the use and the use environment would have to collect a very large (perhaps unfeasibly large) number of very specific items of information.

Next, consider that the information items the problem solver will actually need (of the many that exist) are contingent on the solution path taken by the engineer designing the product. In the example, the problem caused by the yellow color of the circuit board was contingent on the design solution to the component-placing problem selected by the engineer during the development process. That is, the color of the circuit boards in the user factory became an item the problem solvers needed to know only when engineers, in the course of their development of the component placer, decided to use a vision system in the component-placing machine they were designing, and the fact that the boards were yellow became relevant only when the engineers chose a video camera and lighting that could not distinguish the metalized patterns on the board against a yellow background. Clearly, it can be costly to transfer the many items of information that a product or service developer might require—even if each individual item has low stickiness—from one site to another.

### **How Information Asymmetries Affect User Innovation vs. Manufacturer Innovation**

An important consequence of information stickiness is that it results in information asymmetries that cannot be erased easily or cheaply. Different users and manufacturers will have different stocks of information, and may find it costly to acquire information they need but do not have. As a result, each innovator will tend to develop innovations that draw on the sticky information it already has, because that is the cheapest course of action (Arora and Gambardella 1994; von Hippel 1994). In the specific case of product development, this means that users as a class will tend to develop innovations that draw heavily on their own information about need and context of use. Similarly, manufacturers as a class will tend to develop innovations that draw heavily on the types of solution information in which they specialize.

This effect is visible in studies of innovation. Riggs and von Hippel (1994) studied the types of innovations made by users and manufacturers that improved the functioning of two major types of scientific instruments.

**Table 5.1**

Users tend to develop innovations that deliver novel functions.

Type of improvement provided by innovation	Innovation developed by		<i>n</i>
	User	Manufacturer	
New functional capability	82%	18%	17
Sensitivity, resolution, or accuracy improvement	48%	52%	23
Convenience or reliability improvement	13%	87%	24
Total sample size			64

Source: Riggs and von Hippel 1994, table 3.

They found that users tended to develop innovations that enabled the instruments to do qualitatively new types of things for the first time. In contrast, manufacturers tended to develop innovations that enabled users to do the same things they had been doing, but to do them more conveniently or reliably (table 5.1). For example, users were the first to modify the instruments to enable them to image and analyze magnetic domains at sub-microscopic dimensions. In contrast, manufacturers were the first to computerize instrument adjustments to improve ease of operation. Sensitivity, resolution, and accuracy improvements fall somewhere in the middle, as the data show. These types of improvements can be driven by users seeking to do specific new things, or by manufacturers applying their technical expertise to improve the products along known dimensions of merit, such as accuracy.

The variation in locus of innovation for different types of innovations, seen in table 5.1 does fit our expectations from the point of view of sticky information considerations. But these findings are not controlled for profitability, and so it might be that profits for new functional capabilities are systematically smaller than profits obtainable from improvements made to existing functionality. If so, this could also explain the patterns seen.

Ogawa (1998) took the next necessary step and conducted an empirical study that did control for profitability of innovation opportunities. He too found the sticky-information effect—this time visible in the division of labor *within* product-development projects. He studied patterns in the development of a sample of 24 inventory-management innovations. All were jointly developed by a Japanese equipment manufacturer, NEC, and by a user firm, Seven-Eleven Japan (SEJ). SEJ, the leading convenience-store

company in Japan, is known for its inventory management. Using innovative methods and equipment, it is able to turn over its inventory as many as 30 times a year, versus 12 times a year for competitors (Kotabe 1995). An example of such an innovation jointly developed by SEJ and NEC is just-in-time reordering, for which SEJ created the procedures and NEC the hand-held equipment to aid store clerks in carrying out their newly designed tasks. Equipment sales to SEJ are important to NEC: SEJ has thousands of stores in Japan.

The 24 innovations studied by Ogawa varied in the amount of sticky need information each required from users (having to do with store inventory-management practices) and the amount of sticky solution information required from manufacturers (having to do with new equipment technologies). Each also varied in terms of the profit expectations of both user and manufacturer. Ogawa determined how much of the design for each was done by the user firm and how much by the manufacturer firm. Controlling for profit expectations, he found that increases in the stickiness of user information were associated with a significant increase in the amount of need-related design undertaken by the user (Kendall correlation coefficient = 0.5784,  $P < 0.01$ ). Conversely he found that increased stickiness of technology-related information was associated in a significant reduction in the amount of technology design done by the user (Kendall correlation coefficients = 0.4789,  $P < 0.05$ ). In other words, need-intensive tasks within product-development projects will tend to be done by users, while solution-intensive ones will tend to be done by manufacturers.

### Low-Cost Innovation Niches

Just as there are information asymmetries between users and manufacturers as classes, there are also information asymmetries among *individual* user firms and individuals, and among individual manufacturers as well. A study of mountain biking by Lüthje, Herstatt, and von Hippel (2002) shows that information held locally by individual user-innovators strongly affects the type of innovations they develop.

Mountain biking involves bicycling on rough terrain such as mountain trails. It may also involve various other extreme conditions, such as bicycling on snow and ice and in the dark (van der Plas and Kelly 1998). Mountain biking began in the early 1970s when some young cyclists started

to use their bicycles off-road. Existing commercial bikes were not suited to this type of rough use, so early users put together their own bikes. They used strong bike frames, balloon tires, and powerful drum brakes designed for motorcycles. They called their creations "clunkers" (Penning 1998; Buenstorf 2002).

Commercial manufacture of mountain bikes began about 1975, when some of the early users of mountain bikes began to also build bikes for others. A tiny cottage industry developed, and by 1976 a half-dozen small assemblers existed in Marin County, California. In 1982, a small firm named Specialized, an importer of bikes and bike parts that supplied parts to the Marin County mountain bike assemblers, took the next step and brought the first mass-produced mountain bike to market. Major bike manufacturers then followed and started to produce mountain bikes and sell them at regular bike shops across the United States. By the mid 1980s the mountain bike was fully integrated in the mainstream bike market, and it has since grown to significant size. In 2000, about \$58 billion (65 percent) of total retail sales in the US bicycle market were generated in the mountain bike category (National Sporting Goods Association 2002).

Mountain biking enthusiasts did not stop their innovation activities after the introduction of commercially manufactured mountain bikes. They kept pushing mountain biking into more extreme environmental conditions, and they continued to develop new sports techniques involving mountain bikes (*Mountain Bike* 1996). Thus, some began jumping their bikes from house roofs and water towers and developing other forms of acrobatics. As they did so, they steadily discovered needs for improvements to their equipment. Many responded by developing and building the improvements they needed for themselves.

Our sample of mountain bikers came from the area that bikers call the North Shore of the Americas, ranging from British Columbia to Washington State. Expert mountain bikers told us that this was a current "hot spot" where new riding styles were being developed and where the sport was being pushed toward new limits. We used a questionnaire to collect data from members of North Shore mountain biking clubs and from contributors to the mailing lists of two North Shore online mountain biking forums. Information was obtained from 291 mountain bikers. Nineteen percent of bikers responding to the questionnaire reported developing and building a new or modified item of mountain biking equipment for their own use. The

**Table 5.2**

Innovators tended to use solution information they already had “in stock” to develop their ideas. Tabulated here are innovators’ answers to the question “How did you obtain the information needed to develop your solution?”

	Mean	Median	Very high or high agreement
“I had it due to my professional background.”	4.22	4	47.5%
“I had it from mountain biking or another hobby.”	4.56	5	52.4%
“I learned it to develop this idea.”	2.11	2	16%

Source: Lüthje et al. 2003.  $N = 61$ . Responses were rated on a seven-point scale, with 1 = not at all true and 7 = very true.

innovations users developed were appropriate to the needs associated with their own riding specialties and were heterogeneous in function.

We asked mountain bikers who had innovated about the sources of the need and solution information they had used in their problem solving. In 84.5 percent of the cases respondents strongly agreed with the statement that their need information came from *personal needs they had frequently experienced* rather than from information about the needs of others. With respect to solution information, most strongly agreed with the statement that *they used solution information they already had*, rather than learning new solution information in order to develop their biking equipment innovation (table 5.2).

## Discussion

To the extent that users have heterogeneous and sticky need and solution information, they will have heterogeneous low-cost innovation niches. Users can be sophisticated developers within those niches, despite their reliance on their own need information and solution information that they already have in stock. On the need side, recall that user-innovators generally are lead users and generally are expert in the field or activity giving rise to their needs. With respect to solution information, user firms have specialties that may be at a world-class level. Individual users can also have high levels of solution expertise. After all, they are students or employees during the day, with training and jobs ranging from aerospace engineering

to orthopedic surgery. Thus, mountain bikers might not want to *learn* orthopedic surgery to improve their biking equipment, but if they already *are* expert in that field they could easily draw on what they know for relevant solution information. Consider the following example drawn from the study of mountain biking discussed earlier:

I'm a human movement scientist working in ergonomics and biomechanics. I used my medical experience for my design. I calculated a frame design suitable for different riding conditions (downhill, climb). I did a CAD frame design on Catia and conceived a spring or air coil that can be set to two different heights. I plan to build the bike next year.

Users' low-cost innovation niches can be narrow because their development "labs" for such experimentation often consist largely of their individual use environment and customary activities. Consider, for example, the low-cost innovation niches of individual mountain bikers. Serious mountain bikers generally specialize in a particular type of mountain biking activity. Repeated specialized play and practice leads to improvement in related specialized skills. This, in turn, may lead to a discovery of a problem in existing mountain biking equipment and a responsive innovation. Thus, an innovating user in our mountain biking study reported the following: "When doing tricks that require me to take my feet off the bike pedals in mid-air, the pedals often spin, making it hard to put my feet back onto them accurately before landing." Such a problem is encountered only when a user has gained a high level of skill in the very specific specialty of jumping and performing tricks in mid-air. Once the problem has been encountered and recognized, however, the skilled specialist user can re-evolve the same problematic conditions at will during ordinary practice. The result is the creation of a low-cost laboratory for testing and comparing different solutions to that problem. The user is benefiting from enjoyment of his chosen activity and is developing something new via learning by doing at the same time.

In sharp contrast, if that same user decides to stray outside his chosen activity in order to develop innovations of interest to others with needs that are different from his own, the cost properly assignable to innovation will rise. To gain an equivalent-quality context for innovation, such a user must invest in developing personal skill related to the new innovation topic. Only in this way will he gain an equivalently deep understanding of the problems relevant to practitioners of that skill, and acquire a "field

laboratory” appropriate to developing and testing possible solutions to those new problems.

Of course, these same considerations apply to user firms as well as to individual users. A firm that is in the business of polishing marble floors is a user of marble polishing equipment and techniques. It will have a low-cost learning laboratory with respect to improvements in these because it can conduct trial-and-error learning in that “lab” during the course of its customary business activities. Innovation costs can be very low because innovation activities are paid for in part by rewards unrelated to the novel equipment or technique being developed. The firm is polishing while innovating—and is getting paid for that work (Foray 2004). The low cost innovation niche of the marble polishing firm may be narrow. For example, it is unlikely to have any special advantage with respect to innovations in the polishing of wood floors, which requires different equipment and techniques.